Journal of
Sensor and Actuator Networks

Review

DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges

Kazeem B. Adedeji [1,*], Adnan M. Abu-Mahfouz [1,2] and Anish M. Kurien [1]

[1] Department of Electrical Engineering, Tshwane University of Technology, Pretoria 0001, South Africa; a.abumahfouz@ieee.org (A.M.A.-M.); kurienam@tut.ac.za (A.M.K.)
[2] Council for Scientific and Industrial Research, Pretoria 0184, South Africa
* Correspondence: adedejikb@tut.ac.za or kezman0474@yahoo.com

Abstract: In recent times, distributed denial of service (DDoS) has been one of the most prevalent security threats in internet-enabled networks, with many internet of things (IoT) devices having been exploited to carry out attacks. Due to their inherent security flaws, the attacks seek to deplete the resources of the target network by flooding it with numerous spoofed requests from a distributed system. Research studies have demonstrated that a DDoS attack has a considerable impact on the target network resources and can result in an extended operational outage if not detected. The detection of DDoS attacks has been approached using a variety of methods. In this paper, a comprehensive survey of the methods used for DDoS attack detection on selected internet-enabled networks is presented. This survey aimed to provide a concise introductory reference for early researchers in the development and application of attack detection methodologies in IoT-based applications. Unlike other studies, a wide variety of methods, ranging from the traditional methods to machine and deep learning methods, were covered. These methods were classified based on their nature of operation, investigated as to their strengths and weaknesses, and then examined via several research studies which made use of each approach. In addition, attack scenarios and detection studies in emerging networks such as the internet of drones, routing protocol based IoT, and named data networking were also covered. Furthermore, technical challenges in each research study were identified. Finally, some remarks for enhancing the research studies were provided, and potential directions for future research were highlighted.

Keywords: attack detection; cyber security; DDoS attack; deep learning; entropy; IoT; machine learning

check for updates

Citation: Adedeji, K.B.; Abu-Mahfouz, A.M.; Kurien, A.M. DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 51. https://doi.org/10.3390/jsan12040051

Academic Editor: Guangjie Han

Received: 27 May 2023
Revised: 27 June 2023
Accepted: 3 July 2023
Published: 6 July 2023

1. Introduction

The internet of things (IoT) has recently emerged as one of the enabling technologies that have been implemented in a variety of applications [1,2]. In the same vein, innovative internet-enabled technologies such as the Internet of Flying Things (IoFT) [3], the Internet of Drones (IoD) [4], the Flying Ad-hoc Network (FANET), or drone networks [5] are being implemented to provide decentralized and scalable solutions in these applications. Recently, the IoD has emerged with a new paradigm, where a set of flying vehicles (unmanned aerial vehicles (UAVs)) communicate among themselves and with a ground control station via the internet to execute a range of tasks in various ways [6,7]. Connections within the network entities in these technologies are made through the insecure internet and inherently broadcast wireless media. Coupled with the fact that most IoT devices lack access control, have insecure default passwords, and use unprotected credentials, these technologies are gradually becoming desirable targets for cyberattacks. For instance, the UAVs and some other entities involved in the IoD, FANET, or IoFT are vulnerable to jamming, command injection, and Global Positioning System (GPS) spoofing attacks [5–14]. With the creation of low-cost software-defined radios (SDRs) [15,16], the potential for GPS spoofing has substantially increased. A typical GPS spoofing attack against a UAV system is shown in Figure 1, where the attacker creates a fake GPS signal by tuning an SDR to the GPS frequency. The legitimate GPS signal is therefore overpowered by the fake signal. This could divert the target drone from its intended path.

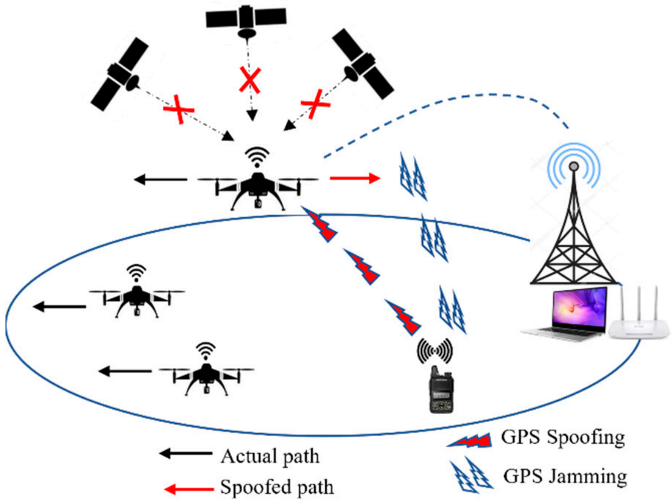

Figure 1. A typical GPS spoofing and jamming attack on a UAV system [8].

The emerging networks are being paired with the IoT in order to create autonomous and scalable solutions. The growing interest in IoT applications has contributed to the huge deployment of low-power and lossy networks (LLN) [17], which facilitate communication between physical objects in the real world and their connection to the Internet. However, due to the specific properties and constraints of these networks, such as a lack of infrastructure and limited physical security, among others, the Internet Engineering Task Force (IETF) specified a standard routing protocol—called the routing protocol for low-power and lossy networks (RPL) [18]—to address these constraints. This protocol has been discovered [18] to be susceptible to a wide range of attacks that cause denial of service. One of the most prevalent attacks on the internet and internet-enabled networks such as the IoT, RPL-IoT, software-defined networks (SDN), and named data networking (NDN) [19] is the DDoS attack, along with its variants. These attacks are frequently launched to bring down a target network with high volumes of traffic, which could exceed 2.5 TB/s, as reported in [20]. Therefore, timely detection of such an attack is crucial.

Various detection methods have been developed in the past, with varying degrees of success. These methods range from the traditional approach [21] to the recently applied machine learning [22]. This paper presents a survey of studies on attack detection methodologies. Therefore, a survey from 2000 to 2023 of relevant literature on DDoS attack detection studies in IoT networks was conducted. Figure 2 shows the analysis of the selected papers [21–211] on attack detection in the literature. As shown in Figure 2a, which depicts the temporal distribution of the articles per year, a growing trend of relevant articles in the field has been noticeable since 2013. Indeed, more than half of the articles analysed in this survey have been published in the last six years. A total of 211 articles were reviewed; more specifically, 64.6% of these articles were published in journals, 27.4% in conference proceedings, 1.9% in books, 0.5% in theses, and only 5.6% appeared in web sources, as illustrated in Figure 2b. Consequently, IEEE is the most relevant publisher within this scope, with 70 articles (35.4%), followed by Elsevier with 30 articles (15.2%). Springer, MDPI, and others (a combination of less famous publishers) have 22, 12, and 29 articles, respectively. Finally, Hindawi (5 articles) and IET, Inderscience, and Trans Tech (1 article each) have less relevance (see Figure 2b).

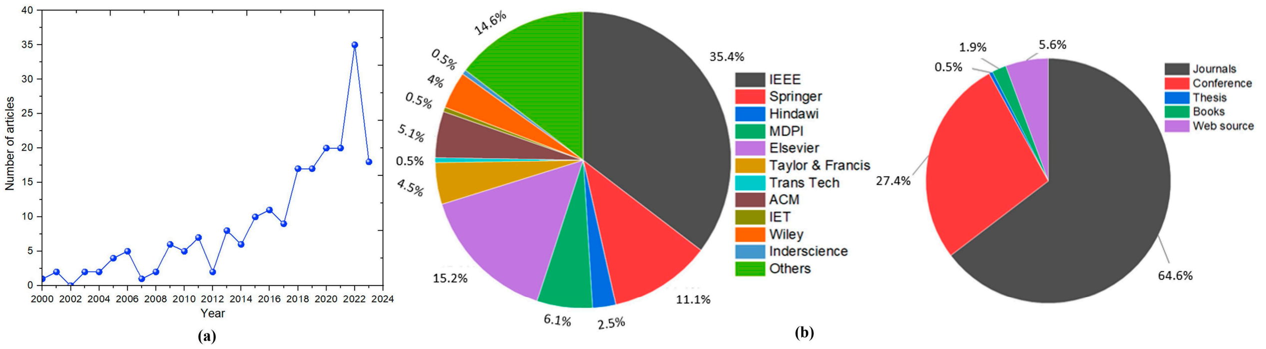

Figure 2. Analysis of the selected articles in the literature review: (**a**) temporal distribution of articles per year, (**b**) distribution of articles per source type and publisher.

Several related survey papers on DDoS attacks and detection methods are available in the literature [23–33]. Most of these surveys concentrate on a particular portion of this subject. For example, the survey by Nooribakhsh and Mollamotalebi [32] focused on a statistical approach for DDoS attack detection. In Khalaf et al. [29], research studies based on artificial intelligence and statistical methods were surveyed. In [28,31,33], only machine learning approaches for attack detection were emphasized. Table 1 presents a summary and comparison of our survey paper with other related surveys. In this table, a list of topics that were not covered, were partially covered, or were covered by other surveys is summarized. From Table 1, it is observed that related surveys either confine their analysis to certain DDoS attack detection methods [28,29,31–33] or only compare a small subset of them [24,26–28,33]. Recent articles on DDoS attack detection methods are not addressed in several other studies [23–27]. These, among other reasons, prompt the need to conduct an in-depth and updated survey on DDoS attack detection. Consequently, this paper presents a structured and broad survey of the existing research studies on DDoS attacks and detection methods in the IoT and other internet-enabled networks. A summary of the paper's main contributions is provided below:

- A thorough description of DDoS attack categories and architecture was provided in this paper. Attack detection methods were classified, and research studies under each category are extensively discussed. The research studies in each category are then compared and analysed;
- Attack scenarios and detection studies in emerging networks such as IoDs, IoFT, FANET, RPL-based IoT, and NDN are also investigated;
- This paper covers Chi-square, Chao-based, and queueing model-based attack detection methods that were not covered in existing surveys;
- Apart from the DDoS attacks and detection methods, our survey also provides an overview of the benchmark dataset used for attack detection validation;
- Finally, several research issues and challenges associated with these methods are identified. A focus for future studies is also provided.

The purpose of this study is to broaden the focus and provide an updated research direction on DDoS studies. Although this survey focused more on attack detection and studies in IoT, studies focusing on attack detection in SDN, RPL, NDN, and vehicular or flying things are also investigated and discussed. The remainder of the paper is structured as follows. Section 2 covers the taxonomy of DDoS attacks. It shows the attack architecture and the categories of attack. The categories of DDoS attack detection methods are illustrated in Section 3, along with a thorough review of the literature in each category. Section 4 discusses and compares some of the benchmark datasets used in DDoS attack detection research studies. In Section 5, a discussion of the key findings from the survey is presented. Section 6 provides a summary of some significant research challenges that could be the focus of future studies, and Section 7 concludes the paper.

Table 1. Comparison of the current study with other related survey papers.

Topics Covered	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	Our Work
Security issues in IoT	×	✔	×	✔	✔	✔	×	✔	✔	×	✔	✔
Detailed taxonomy of DDoS attacks	✗	✗	×	✗	✗	×	×	✗	✗	×	×	✔
Entropy-based detection	×	✗	✔	✗	×	×	×	✗	×	✗	×	✔
Chaos-based detection	×	×	×	×	×	×	×	×	×	×	×	✔
Detection-based on Chi-square	×	×	×	×	×	×	×	×	×	×	×	✔
Detection-based on queuing model	×	×	×	×	×	×	×	×	×	×	×	✔
Statistical forecasts methods	×	×	✔	×	×	×	✔	×	×	✔	×	✔
Traffic pattern analysis	×	×	×	×	×	×	×	×	×	×	×	✔
Correlation of IP address	✔	×	×	✗	×	×	×	×	×	×	×	✔
Heuristic-based detection	×	×	✔	×	×	×	✗	×	×	×	×	✔
Detection via machine learning	×	×	✔	×	×	✗	✔	✔	✔	×	✗	✔
Detection via deep learning	×	×	✔	×	×	×	×	✗	×	×	×	✔
Attack and detection studies in IoD, FANET, RPL-based IoT, and NDN	×	×	×	×	×	×	×	×	×	×	×	✗
Comparison of the research studies under each detection methods	×	×	✔	×	×	×	✔	✔	✔	✔	✗	✔
Benchmarked datasets	×	×	×	×	×	×	✗	×	✗	×	×	✔
Evaluation metrics	×	×	✔	×	×	×	✗	×	✔	×	×	✔
Open challenges	✔	×	✔	×	✗	×	✔	✔	✔	✗	×	✔

✗ Partially covered; × Not covered; ✔ Covered.

2. Taxonomy of DDoS Attacks

2.1. DDoS Attack Architecture

In a DDoS attack, numerous devices attack a single server or network. This attack aims to overload a targeted server or network with numerous spoof requests to interfere with its regular traffic. This overwhelms the network resources, and, as a result, legitimate traffic encounters service disruptions. These attacks are executed with networks of internet-connected devices—including PCs and other devices (such as IoT devices) that have become infected with malicious software and are, thus, susceptible to remote manipulation. These devices are known as bots. DDoS attacks are effective because they use botnets, or groups of compromised computers, as their primary attack source. Once a botnet has been established, the attacker can direct the attack by sending remote commands to each bot. Each of the bots in the botnet sends queries to the IPs of the victim's server while it is being targeted by the botnet, which may overwhelm the network and disrupt legitimate traffic. Each bot is a real internet device which makes it challenging to differentiate between attacks and legitimate traffic. As was already established, DDoS attackers initiate their attacks via a botnet; therefore, the architecture of a DDoS attack will consist of an attacker, a botnet, and the target network or server. Different architectures emerge from how botnets are managed. As reported in [34], DDoS attack architecture can be categorized as centralized or decentralized. These architectures are illustrated in Figure 3. A centralized one is depicted in Figure 3a. It has an attacker, the target server, a botnet, and a command-and-control (C&C) system. In certain literature studies [35–37], the C&C systems are sometimes referred to as handlers. The bot computers in the botnet cannot communicate with one another under this architecture. Instead, every bot is linked to a C&C system. Therefore, the botnet is controlled by sending commands to these bots directly. In a decentralized architecture (Figure 3b), the bots establish a peer-to-peer (P2P) network. An attack query is sent to a certain bot to start the DDoS attack. The commands are then forwarded by this bot through P2P to other bots in the network. Table 2 displays a comparison of the two DDoS attack architectures. The security in the centralized architecture is very strong because the botnet cannot be detected just by identifying the communications between the bots. Unlike the decentralized architecture, the P2P communication pattern between the bots may be recognized, which makes the botnet easier to identify. Once the botnet is discovered, the source of the attack may be determined, and the strength of such attacks becomes almost negligible. Additionally, the attacker can simply modify their attack strategy through real-time control of the botnet in the centralized architecture. The authors proposed a new low-cost architecture that consists of a DDoS attacker, a target server, and a botnet. In this architecture, an attack strategy is achieved by only writing a malware bot with an attack module. This eradicates the botnet management issues that are present in other architectures. Thus, the management cost is zero. Since there is no command-and-control system, the proposed architecture is robust and suitable for resource-constrained devices.

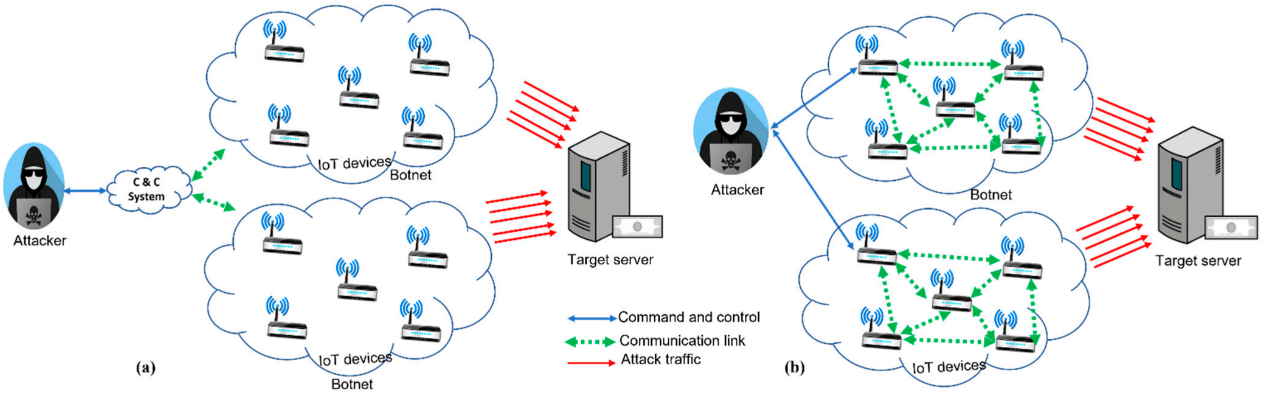

Figure 3. Architecture of a DDoS attack: (**a**) centralized, (**b**) decentralized.

Table 2. Comparison of the DDoS attack architectures.

Architecture	Advantages	Limitations
Centralized architecture	• Security is strong. • Flexible	• Cost of managing a botnet is high. • Not robust.
Decentralized architecture	• Robust • Flexible	• Security is weak. • Cost of managing a botnet is high.

Irrespective of the architecture, a DDoS attack shares a common goal. As shown in Figure 3, the bots under the control of the handlers send out the attack packets. These packets converge at the target server to exhaust its resources. Server resource exhaustion can occur due to a server's bandwidth, memory size, or CPU cycle [38,39]. Singh and De [38] describe the probability of bandwidth exhaustion using (1) [38]:

$$P_B = \frac{\left(\frac{a^c}{c!}\right)}{\sum\limits_{i=0}^{c} \frac{a^i}{i!}} \tag{1}$$

where the following definition is used:

$$a = \frac{1}{B_T}\left[\frac{\delta_{BA}}{\tau_{BA}} + \frac{\delta_{BN}}{\tau_{BN}}\right] \tag{2}$$

In (1), P_B denotes the probability of bandwidth exhaustion, c is the number of unused bandwidths, and B_T is the total bandwidth consumed. The packet sizes of the attacking and legitimate clients are represented by δ_{BA} and δ_{BN}, respectively. τ_{BA} is the inter-arrival rate of the attacking packet, while τ_{BN} is that due to the legitimate packet. In a situation where the packet due to the attack and the legitimate client are of the same size, $\delta_{BA} = \delta_{BN} = \delta_B$. Then, a is expressed as [38] follows:

$$a = \frac{\delta_B}{B_T}\left[\frac{1}{\tau_{BA}} + \frac{1}{\tau_{BN}}\right] \tag{3}$$

$$a = k \times \frac{1}{\tau_{BA}} \tag{4}$$

These expressions show that the inter-arrival rate of the attacking traffic has a significant impact on the probability of bandwidth depletion. In this model, the distribution due to the legitimate client is Gaussian, whereas the arrival rate of the packet from the attacking client is modelled via Poisson distribution. The overall probability of depletion of the victim's resources P_{TA} is expressed using (5) [38]:

$$P_{TA} = 1 - (1 - P_B)(1 - P_M) \tag{5}$$

where P_M is the probability due to memory consumption.

Luo et al. [39] used a simple congestion window model to describe the probability of a successful attack on the bandwidth. For a given time $t > 0$, which indicates the time at which legitimate traffic X_t is successfully transmitted, the steady-state throughput P of the legitimate traffic flow during an attack period T and the probability of a successful attack P_{wa} are expressed using (6) [39]:

$$\begin{cases} P = \lim\limits_{t \to \underset{\leftharpoondown}{\infty}} \frac{X_t}{t \times C} \\ P_{wa} = P \end{cases} \tag{6}$$

where C indicates the ideal burst magnitude for a successful attack to occur. The values of P_{wa} vary between 0 and 1. A lower P_{wa} indicates that the attack is very significant. The model described by Singh and De [38] was demonstrated on the CAIDA dataset to evaluate how the DDoS attack affected bandwidth usage. In this dataset, 104 distinct IP addresses were considered and split into 52 regular and 52 attacking packets. Using an attack period of 40 s for each IP, evaluation results showed that, during the DDoS attack, attacking packets utilize over 6.5 GB/s of bandwidth while regular traffic uses just 5.6 MB/s. This study supports the assertion that DDoS attacks have a considerable impact on the target server's bandwidth depletion. To avoid a server shutdown, the attack needs to be detected and mitigated.

As can be observed in Figure 3, in DDoS, the use of botnets to launch attacks is evident. Today, botnets are not limited to personal computers. The attacker can further increase the traffic they produce by using handheld and IoT devices [39]. According to a report released by Akamai [40] (see Figure 4), the volume of DDoS attack traffic generated through IoT devices between 14 July and 16 December 2020 was more than 300 GB/s. The malware Mirai, XOR, and Spike were observed to have more than 300 GB/s. Attacks are increasingly significant, since hackers may use IoT devices such as Wi-Fi routers, security cameras, and smart TVs to launch attacks by taking advantage of their inherent weaknesses. These vulnerable devices can be exploited to flood target networks with traffic to take down their servers. IoT devices are susceptible to remote manipulation by attackers because of the open nature of the internet and poorly maintained firmware. Once infected, these devices are integrated into botnets and start to take over the targeted server or service [41]. The rapid expansion of unsecured IoT devices has provided an expanding pool of DDoS attack resources.

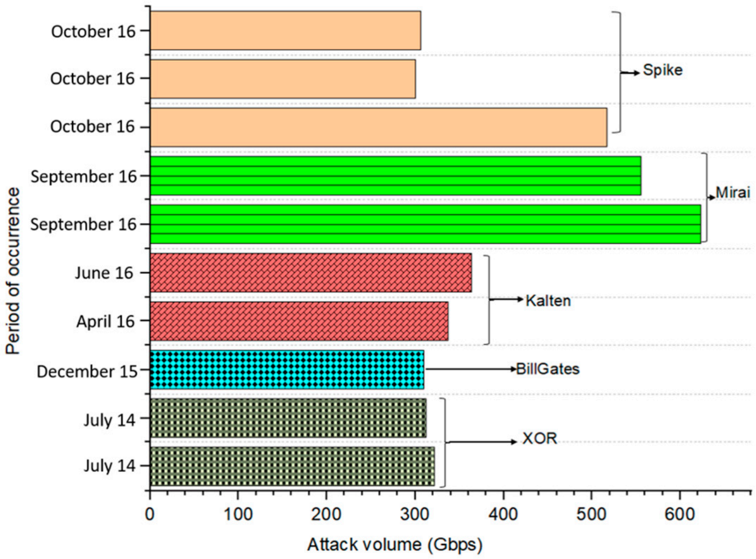

Figure 4. Volume of DDoS attacks generated through IoT devices in 2020 [40].

2.2. *DDoS Attack Classification and Types in IoT Networks*

DDoS attacks differ greatly in terms of how they are initiated and the impact they have on the target server. Even so, they all share the same goal of interfering with legitimate traffic. IoT networks and devices are heterogeneous in nature. As a result, there will be a variety of threats focused specifically on them. As shown in Figure 5, DDoS attacks can therefore be divided into three categories: volumetric base, protocol base, and application base. Numerous DDoS attack types under each division are also illustrated.

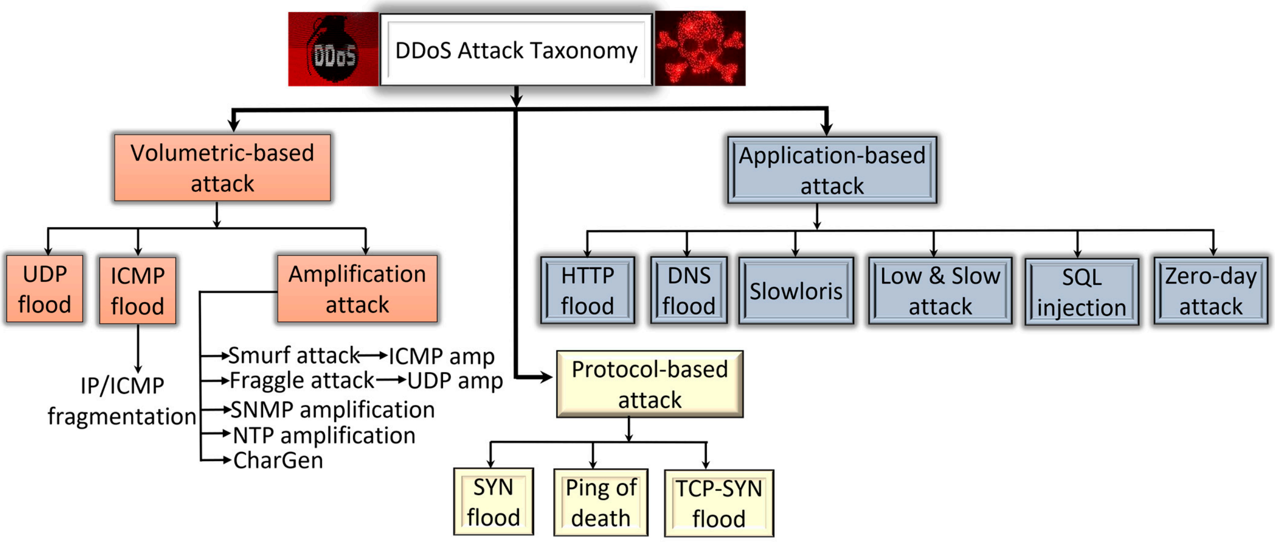

Figure 5. The major categories of DDoS attacks.

2.2.1. Volumetric-Based DDoS Attacks

This category of DDoS attack floods the target network's available bandwidth with huge data packets, thus overwhelming it. The attack saturates the targeted network with abnormally large amounts of malicious traffic to deny service to authorized users. Any server that cannot handle the increased traffic volume can be brought down instantly by such attacks. The generic structure of a conventional volumetric-based DDoS attack is illustrated in Figure 6.

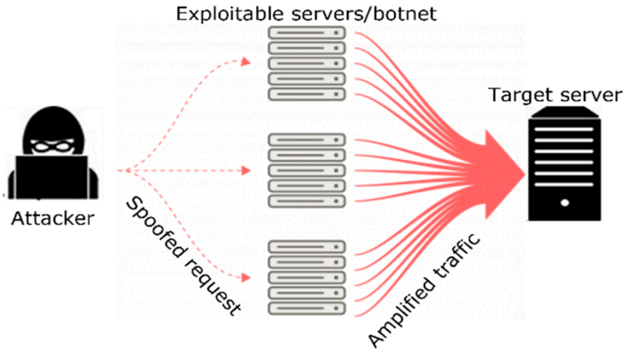

Figure 6. Conventional structure of a volumetric-based DDoS attack.

The attackers' goal, as depicted in Figure 6, is to overwhelm the bandwidth of a victim site by sending as much traffic as they can. Attackers mostly exploited amplification strategies, which involve sending brief legitimate requests to a domain name server (DNS) with a spoof source IP address of the victim to overwhelm the target server [42]. Volumetric attacks are recognized by a massive amount of traffic (100 GB/s or more). A reflection medium can be used to produce gigabits of traffic from a small amount of traffic. Examples of volumetric-based DDoS attacks include user datagram protocol (UDP) flooding, internet control message protocol (ICMP) flooding, network time protocol (NTP), simple network message protocol (SNMP) amplifications, character generator (CharGen), Smurf attack, and Fraggle attack, among others. In some of the literature [43,44], they are referred to as "bandwidth depletion attacks".

2.2.2. Protocol-Based DDoS Attacks

The protocol-based attack is often referred to as the "network-layer attack". Rather than solely relying on overwhelming traffic volume, as the volumetric base attack does, it takes advantage of the protocol stack's layer 3 and layer 4 weaknesses to render the target server inaccessible. Existing server resources as well as other resources, such as firewalls, are consumed by this kind of attack. This attack is therefore frequently referred to as a "resource depletion attack" [30]. The attack magnitude is expressed in packets per second (pps). Examples include synchronization (SYN) floods, ping of death, transmission control protocol (TCP) floods, TCP–SYN floods, and their variations.

2.2.3. Application-Based DDoS Attacks

This is also known as a "Layer 7" attack because it takes place at the seventh layer of the open system interconnection (OSI) model. The application layer is overloaded with too many login or search requests. These attacks are the hardest to localize because the attacker generates attack traffic at a reduced rate, and the request sent is very similar to regular traffic [45]. The strength of this attack is expressed in requests per second (rps). Examples include HTTP flood, Slowloris, zero-day attack, domain name server (DNS) flood, low and slow attack, and SQL injection [45], among others.

Table 3 provides a comparison of the three major attack classifications. The volumetric attack's objective is to jam up the network with many illegitimate network packets. Due to the increasing volume of traffic packets and traffic congestion, the impacted resources are unable to complete any operations or respond to any requests. The protocol-based attack tries to take advantage of flaws in the network protocol to consume the connection status table produced by some network devices. The application attack is more sophisticated and typically starts with minimal bandwidth utilization. The resources of the network are gradually depleted as it targets specific services or applications. To keep the connection active, the attacker sends the requested packets within a very small packet window. Volumetric attacks are the most prevalent types of attacks since they are straightforward and simple to produce. A report presented in [46] (see Figure 7a) revealed the frequency of occurrence of each DDoS attack category encountered between January 2020 and March 2021. It was observed that, over those 15 months, over 65.2% of all attacks were volumetric in nature, while protocol-based and application-based DDoS attacks accounted for about 20.5% and 14.3%, respectively. UDP flood attack was reported to have a 62.5% application percentage among these attack categories. Despite the complexity of attacks based on both protocols and applications, application-based attacks are the hardest to localize, since they closely resemble legitimate traffic. In Table 4, an overview of the most common DDoS attack types is presented. The attack features and their effect on the target server are also discussed. A combination of these attacks has been used recently to launch a multi-vector attack. With a fast increase in the development of DDoS detection solutions, attackers have discovered the usage of a variety of attack categories, including volumetric, protocol, and application-specific attacks. These attack vectors can be combined to perform a multi-vector DDoS attack, which has a greater effect on the target server than a single volumetric attack. Multi-vector DDoS attacks have been more common in the IoT network recently. According to a report presented by Nexus Guard [46], attackers have successfully deployed a mix of a UDP flood and an NTP amplification attack, as seen in Figure 7b [47], with a record utilization of 17.06%. Overall, 9.41% of multi-vector attacks utilize ICMP and UDP flooding, while 6.47% of multi-vector DDoS attacks use ICMP, UDP flood, and NTP amplification.

Table 3. Overview of the three major DDoS attack classification.

Attack Type	Features	Attack Magnitude	Effect on Target Server	Attack Complexity	Affected Layer	Frequency of Occurrence
Volumetric-based	The use of a huge amount of traffic to saturate the bandwidth of the target server	Bits per second (bps), Gbps, flood	Access to the target resources may be totally blocked by the attack's sheer volume of traffic.	Easy to generate using simple amplification techniques	Network layer	Most common
Protocol-based	It exploits the weakness in layers 3 and 4 of the protocol stack to make the target server not accessible.	Packets per second (pps)	It disrupts service by consuming all the target server's processing power or resources, including the firewall.	Less complex	Network and transport	More common
Application-based	It harnesses the flaws in layer 7 of the protocol stack to make the target server not accessible.	Requests per second (rps)	It creates a session with the target and then uses up its resources by completely dominating processes.	Complex and difficult to detect	Application	Less common

Table 4. Overview of the most common DDoS attacks.

Attack Type	Classification	Features	IP Spoofing	Attacked Layer	Effect
TCP–SYN flood	Protocol-based	Exploits TCP's three-way handshaking.	Spoofed	Transport	Obsess the server's resources
HTTP flood	Application-based	Exploits HTTP GET and HTTP POST request.	Non-spoofed	Application	Consumes server's entire resources
Slowloris	Application-based	Maintains the HTTP sessions for the longest feasible time.	Non-spoofed	Application	Consumes all sockets

Table 4. *Cont.*

Attack Type	Classification	Features	IP Spoofing	Attacked Layer	Effect
HTTP fragmentation	Application-based	Splits an HTTP packet into smaller pieces and broadcast them at the slowest rates possible.	Non-spoofed	Application	Consumes all sockets
IP packet option field/IP null	Protocol-based	Sets 1 to all quality-of-service bits.	Spoofed	Network layer	The victim's processing capacity is overloaded
Ping of death	Protocol-based	Forms a data packet that exceeds maximum packet size.	Spoofed	Network layer	Overloads the buffer and causes system crash
UDP flood	Volumetric-based	Sends a significant volume of UDP packets to a target's specified or random port.	Spoofed	Transport layer	Consumes network bandwidth
ICMP flood	Volumetric-based	Utilizes the ECHO request packet of ICMP.	Spoofed	IP layer	Saturates victim's network bandwidth
Fraggle	Volumetric-based	Sends UDP_ECHO packets to the network amplifier.	Spoofed	IP layer	Saturates victim's network bandwidth
NTP amplification	Volumetric-based	Exploits NTP using MON_GETLIST command.	Spoofed	Application layer	Saturates victim's network bandwidth
DNS flooding	Application-based	Utilizes an amplified DNS response query.	Spoofed	Application layer	Saturates victim's network bandwidth

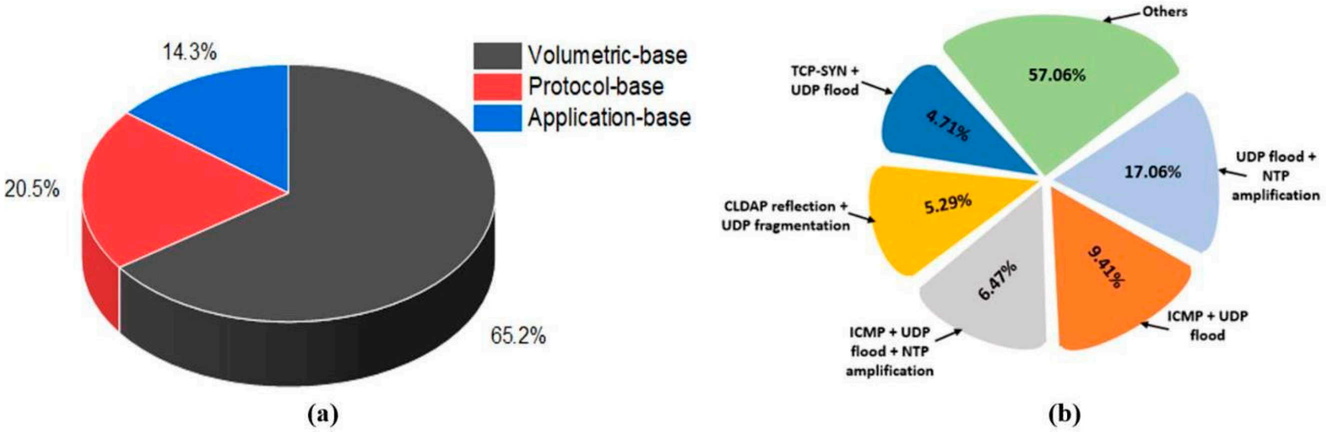

Figure 7. Distribution of DDoS attacks: (**a**) frequency of attack categories between January 2020 and March 2021, (**b**) multi-vector attack scenarios.

3. DDoS Attack Detection Methods

Based on the increased frequency of attacks and their effect on the targeted network resources, it is vital to mitigate the attacks through an effective attack detection methodology. Numerous research studies dealing with this problem have been published, and different attack detection methodologies have been proposed, with varying degrees of success. In this section, DDoS attack detection methodologies are classified into three categories based on their technical nature of operation. As shown in Figure 8, DDoS attack detection methodologies are classified as traditional methods, signature-based detection, and anomaly-based detection.

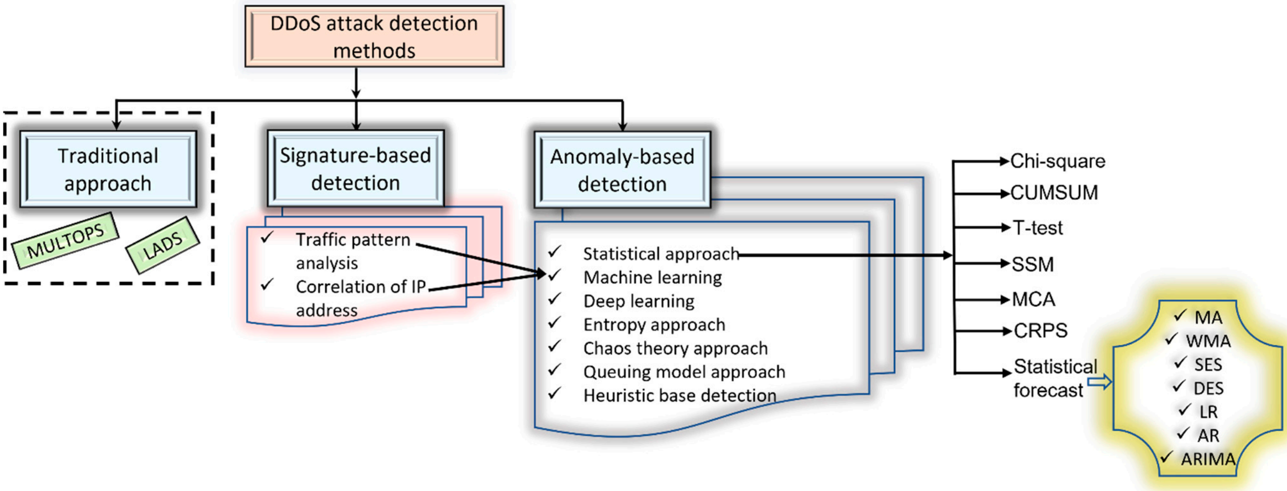

Figure 8. DDoS attack detection methodologies.

Traditional methods: These concentrate on measuring the traffic volume. When the measured traffic volume exceeds a predetermined level, a DDoS attack is identified.

Signature-based detection: This method uses attack signatures that are kept in a database to find attacks. It involves tracking traffic patterns and comparing them to pre-existing signatures. Any differences from the previously recorded patterns indicate malicious traffic. It implies that only attacks whose signatures have been previously stored in the database can be detected. The method has high accuracy in detecting known attacks, provided the database is updated. However, any divergence from the attack signatures or novel attack pattern cannot be detected.

Anomaly-based detection: This method involves gathering typical traffic behaviour over a predetermined period and creating a baseline profile. Any incoming pattern that is outside the scope of the baseline is viewed as an anomaly, which suggests that attacks have taken place. This method works incredibly well at identifying unknown and zero-day attackers.

In Table 5, a comparison of the three major classifications of attack detection methods is presented. Although the traditional approach is quick, its applicability to the present-day security threats is constrained by detection accuracy and false alarm rates. The signature-based detection method cannot detect an unknown attack or even a variation in a known attack. Any variations in the already-existing attack signature patterns go undetected by these methods. In this instance, a lot of false alarms are triggered. This necessitates routine database updates of the attack signatures. However, it can be expensive and occasionally difficult to keep an attack signature up to date. The major advantage of anomaly-based detection over signature-based detection is that it can localize fresh attacks whose signatures fall outside of the normal traffic patterns. However, its detection speed is relatively low, since it requires extensive monitoring due to the number of resources used. Further, greater computational overhead is observed, since it requires significant feature training of network traffic behaviours.

Table 5. Overview of the DDoS attack detection classifications.

Methods	Features	Advantages	Limitations
Traditional methods	Measures the volume of traffic.	• Fast detection speed	• Detection is based on thresholds. • Stealth attacks cannot be detected. • High false alarm rates. • Low detection accuracy.
Signature-based detection	Attacks are recognized using signatures of well-known attacks that have been stored in the database.	• High detection accuracy for known attacks. • Low false positive rates • Fast detection speed	• Suffers from detection accuracy for variation of known attacks. • Unknown and zero-day attacks cannot be detected. • Misrepresentation of signature patterns leads to increased false negative rates. • Requires frequent update of the attack signatures in the database.
Anomaly-based detection	Establishes a baseline profile for normal traffic behavioural pattern collected over a predetermined period.	• Efficient for detecting unknown and zero-day attacks. • Detection accuracy is high	• Detection speed is relatively low. • More overheads. • Encrypted attack patterns have not been detected. • Sometimes generate significant false alarm rates

3.1. DDoS Attack Detection Studies Based on the Traditional Approach

The multi-level tree for online packets (MULTOPS) and the large-scale automated DDoS detection system (LADS) are the two most prevalent traditional DDoS attack detection methods. In [21], the authors demonstrated the use of MULTOPS on a software router with simulated attacks. MULTOPS permits routers to identify bandwidth attacks when there is a significant difference in the rates of incoming and outgoing packets from the victim or the attacker. As a result, MULTOPS triggers an attack alarm, since such packets are identified as malicious. When running on a 700 MHz Pentium III PC, routing software with MULTOPS may handle up to 340,000 packets per second [21]. This method has a number of significant flaws, one of which is its inability to detect attacks that leverage several proportional flows to disrupt a victim server. LAD uses a pre-defined bandwidth attack threshold to determine whether a specific incident needs to be reported as a potential DDoS attack or if no other DDoS-related features are present, such as high volumes of SYN, ICMP, or RST packets. This threshold was established in [48] at 26 Mbps. SYN, ICMP, and RST traces were reported in this study when demonstrated on a Tier-1 ISP network.

3.2. DDoS Attack Detection Studies Using Signature-Based Methods

As shown in Figure 8, DDoS attack detection using signatures of known attacks is divided into traffic pattern analysis and correlation of IP address. In most cases, both methods use a machine learning approach to improve detection accuracy. In this section, research studies under each classification are discussed.

3.2.1. Traffic Pattern Analysis

This method is predicated on the notion that infectious packets have the same behavioural patterns that are distinct from those of legitimate ones. For instance, in a botnet attack, a single bot master typically controls all the bots. The same patterns are seen due to requests being delivered to numerous botnet members, which is what is driving the behaviour. This method compares incoming traffic patterns to pre-established legitimate traffic profiles. Any deviation from these profiles indicates malicious traffic. A profile of legitimate traffic is obtained via traffic features recorded when the terminal generating the traffic is secure [49]. One major limitation with this approach is the imbalanced traffic flows due to the dynamic nature of internet traffic patterns. This might lead to the selection of flow features being incorrect [50]. Since traffic pattern analysis requires that the pattern of the network traffic be accurately characterized for better detection accuracy, machine learning algorithms are mostly employed. Thus, there are studies that assess the potential of machine learning classifiers for traffic pattern classification to improve DDoS attack detection. Moore and Zuev [50] attempted to correctly classify internet traffic patterns for DDoS attack detection using Bayesian techniques and obtained 60% accuracy. In [51], network traffic samples collected using sflow protocol from network devices were classified and analysed using a random forest (RF) classifier. The network traffic was compared to signatures collected earlier from network traffic samples to make the detection. The method was tested using a synthesized dataset, comprised of the CIC-DoS, CICIDS2017, and CICIDS2018 datasets. According to the results, the method had a 96% detection rate, a relatively high level of precision, and a low false alarm rate. The detection method has certain inherent shortcomings. The comparison's traffic signatures were made using previously retrieved network traffic samples. However, since internet traffic loads change over time, it becomes harder to choose the right flow features, which leads to imbalanced traffic flows. Shafiq et al. [52] categorized traffic features using a machine learning-based hybrid feature selection approach. Using various network environment datasets, the method was able to tackle the issue of dispersed traffic classification in high-dimensional unbalanced data and obtain 80% flow classification accuracy. For TCP traffic, however, the system's results are not very accurate. Some research investigations not only identify the attack but also pinpoint its source or traceback. In [53], attack detection and traceback were presented using decision tree and grey relation assessment. This study used a traffic-flow pattern-matching approach to pinpoint the attacker's location. The approach achieved detection and localization of attacks with a false positive ratio of 2.4% and a 2–10% false negative ratio during attack detection, while 8–12% false negative rate and 12–14% false positive rate was achieved during source attack tracing. Similar research was performed by Waizumi and Nemoto [22], who used independent component analysis to create a new pattern matching algorithm for DDoS attack source detection. The method bases its requirements for the attack traceback on variations in the number of packets over time. Thus, by comparing the geometries of input traffic patterns and the geometries of output traffic patterns seen at a network branch point, similar to a router, the source of the attack can be determined. In [54], an algorithm for detecting attacks via network traffic pattern assessment was introduced. A simple Hilbert operator was used to describe the traffic pattern over various time intervals. DDoS attack detection was then achieved using Bayesian decision theory. With a detection probability of 0.95, the algorithm detects IP fragments, TCP SYN floods, UDP floods, NTP amplification, and HTTP floods. The method identified a few missed cases. About 5% of network attacks in the dataset were not detected. The approach is also based on the signature of known attacks, which limits its applicability to detecting other attack vectors with dynamic patterns different from the classified data.

3.2.2. Correlation of IP Address

As attackers are notorious for forging packets from originating IPs, it is straightforward to localize the DDoS attack by filtering the attack traffic if the spoofed IP address can be successfully recognized. This method examines and compares the difference between the attacker's spoof IP and the host server's IP. When these IP addresses are not uniform, a DDoS attack alarm is triggered. A method to efficiently identify and block spoofed source IPs was presented by Guo et al. [55]. The detection accuracy of the proposed approach is relatively high. However, because it must constantly communicate with the source side, there is a significant increase in traffic. In Wang and Wang [56], network traffic distribution was analysed, and IP address correlation-based non-uniformity was found. The amount of IP data packets throughout a period was determined in this study, and the amount of data packets in a sliding window was approximated. The correlation coefficient between the IPs for the two subsequent periods was then calculated. Under typical network traffic conditions, the correlation between the source IP address access and the target network is steady. Due to the dispersion of the IPs during an attack scenario, the correlation coefficient is dramatically reduced. With this analysis, the approach could identify the presence of an attack because of the distribution of IPs when attacks occur. Several other notable studies that report the use of IP address correlation and analysis for DDoS attack detection are reported in [57,58]. In [57], an attack detection method using the analysis of source and destination IP address databases was presented, with a reduced false alarm rate. In Xiao et al. [58], this approach was demonstrated for attack detection in a wired network. Results demonstrate that attack traffic was distinguished from normal traffic. In addition, detection accuracy ranging from 91% to 96% was reported when examined on internet data, a data centre traffic trace, and the KDD'99 dataset.

Machine learning has been utilized to enhance the accuracy of this approach, as reported in the literature. A three-layer backpropagation neural network (BPNN) was suggested in [59] to detect and categorize attacks against DNS servers. The findings demonstrate that a three-layered BPNN with a 3-7-3 structure can classify direct DoS and amplification assaults with 99% accuracy. In [60], an artificial neural network (ANN) was used to detect TCP, UDP, and ICMP attacks by blocking the spoofed packet before reaching the target. The proposed approach recorded a detection accuracy of 98% when demonstrated on old datasets and 92% on new datasets.

In Table 6, a summary and comparison of the signature-based DDoS attack detection methods based on their features, advantages, and limitations is provided.

Table 6. Comparison of the signature-based DDoS attack detection methods.

Methods	Features	Advantages	Limitations
Traffic pattern analysis [49–54]	Compares the traffic patterns of infected hosts to the benign hosts	• High precision. • Low false alarm rate	• Inaccurate flow feature selection.
Correlation of IP address [53–55]	Correlates attacker's spoofed IP to the host server's IP.	• High detection accuracy	• Impossible to detect attacks when the protocol headers are encrypted.

3.3. DDoS Attack Detection Studies Using Anomaly-Based Methods

DDoS attack detection methods based on anomaly approaches are classified under the following: detection based on entropy, chaos theory approach, queuing modelling approach, statistical approach, heuristic-based detection, machine learning, and deep learning approaches, as illustrated in Figure 8. In this section, these methods are discussed. As well, research studies in each case are presented.

3.3.1. Entropy-Based Detection Method

Conventionally, entropy assesses the degree of information uncertainty and has been successfully used to calculate the randomness of datasets [61]. Low entropy levels represent the concentration of a distribution, whereas high entropy levels represent a more dispersed probability distribution. Entropy has been suggested as a useful tool for analysing traffic distributions in a number of recent studies [61–64]. These studies have described its application to detect attacks in IoT and SDN networks. This method calculates the entropy by analysing the distribution of features in traffic packets, like source IP, destination IP, flow count, and port numbers. The presence of anomalies in these features is then localized by comparing the entropy values against a predetermined threshold. A sudden shift in entropy levels is typically a potential sign that a DDoS attack may have taken place. Entropy dramatically decreases in the presence of an attack because one flow count dominates. It was discovered in the research studies conducted by Ozcelik and Brooks [65] that the degree to which the entropy changes during these attacks depends on the observed packet header field. While the entropy of the source IP increases due to an attack, the entropy of the destination IP decreases. Entropy will be constant in the absence of an attack. A typical framework for an entropy-based approach to detecting DDoS attacks is shown in Figure 9.

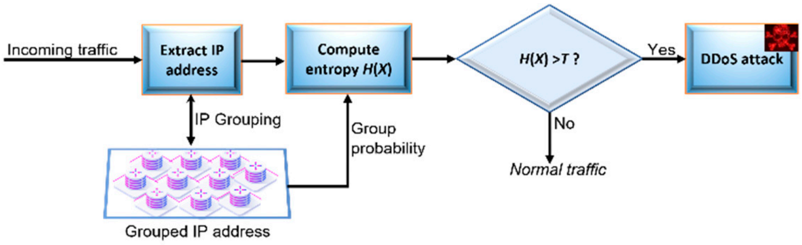

Figure 9. Generic framework of a conventional entropy-based DDoS attack detection approach adapted from [63].

In Figure 9, for each packet of traffic at time slot t, the IP addresses of each packet are extracted and batched in accordance with the source IP addresses. If X represents a random variable that denotes the extracted IP, then the probability of occurrence of each batch $P(x_i)$ and the overall entropy $H(X)$ are estimated as follows:

$$P(x_i) = \frac{x}{\sum\limits_{i=0}^{n} x_i} \tag{7}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log(P(x_i)), \quad x_i \in X \tag{8}$$

Entropy is computed using Shannon's entropy theory and compared to a pre-defined threshold T. An attack alarm is then raised if $H(X)$ is greater than T, as illustrated in (9):

$$H(X)\begin{cases} > T \Rightarrow & Attack \quad traffic \\ \leq T \Rightarrow & Normal \quad traffic \end{cases} \tag{9}$$

Gaurav et al. [63] included a packet discarding process in the traditional entropy-based framework in Figure 9. When a DDoS warning occurs during this process, all the packets with the highest $P(x_i)$ are blacklisted and are subsequently regarded as malicious packets in the following time frame. These packets are discarded if the newly arriving IP address is on the blacklist. Otherwise, a new group of IP addresses is created, and the entropy value is calculated and compared to the threshold. The batch with the highest likelihood of occurrence is identified, and all the IP addresses in this batch are blacklisted once the entropy is above the threshold.

Advancing the studies in [63], the authors in [64] utilized clustering and packet scoring methods to detect and discard malicious requests. In this approach, the change during a DDoS attack is represented by a monotonically increasing convex function following Jensen's inequality using (10) [64]:

$$H[f(x)] \geq f(H[x]) \tag{10}$$

In (10), $H[x]$ is the anticipated value of the convex function. Therefore, it is expected that the cluster entropy of legitimate traffic will be lower than that of a DDoS attack. Thus, the following inequality arises:

$$H(X_{NT}) < H(X_{AT}) \tag{11}$$

In (11), $H(X_{NT})$ is the cluster entropy during normal traffic, whereas $H(X_{AT})$ is that due to attack traffic. In this method, the cluster entropy is contrasted against a threshold, and, if it is above the threshold, then packet scoring methods are used to discard the blacklisted packets, where the score of each packet P_{sc}, is estimated using (12) [64]:

$$P_{sc} = \frac{P_{t_i}}{P_T} \tag{12}$$

where the following holds:

$$P_T = \sum_{i=1}^{n} P_{t_i} \tag{13}$$

In (13), P_{ti} is the incoming traffic packets at time slot t, P_T is the aggregated number of packets, and n is the number of packets arriving during the time slot. The approach was implemented using OMNET++ and achieved high precision, which means that attack traffic is accurately detected.

Giotis et al. [61] implemented an entropy-based approach proposed by [65] for the detection of attacks using flow-based features. Anomalies are detected using pre-established thresholds based on variations in the entropy levels. For portscan, DDoS, and worm attacks, evaluation results demonstrate that good detection accuracy is attained despite having 23%, 27%, and 34% false positive rates, respectively. A three-module detection system using joint entropy metrics was also suggested by the authors in [66]. The detection system has a module for handling incoming traffic, after which the entropy calculator module evaluates the entropy of the packet features. The detection module, which evaluates the estimated entropy against a threshold, makes up the third component. If the estimated entropy exceeds the threshold, an attack alarm is raised, as considered in [61–65]. The method was successfully tested on the DARPA'99, 2009, and current CICDDoS2019 datasets. The use of a static threshold limits its applicability to real-world packets with fluctuating traffic. The threshold chosen has a significant impact on how well the entropy-based approaches function. It has been noted that the effectiveness of any entropy-based solution for DDoS attack detection depends greatly on the threshold value chosen. A static threshold might not always produce the correct results. The threshold value must be updated according to the incoming packet traffic conditions. Thus, David and Thomas [67] proposed an adaptive threshold algorithm and fast entropy computation method for flooding attack detection using the flow count feature. In this method, fast entropy is computed during each time slot using (14) [67]:

$$\overset{\frown}{H}(i,t) = -\log\left(\frac{x(i,t)}{\sum_{i=1}^{n} x(i,t)}\right) + \tau(i,t) \tag{14}$$

where the following definition is necessary:

$$\tau(i,t) = \left\{ \left| \begin{array}{l} \log\left(\frac{x(i,t+1)}{x(i,t)}\right) \\ \log\left(\frac{x(i,t)}{x(i,t+1)}\right) \end{array} \right| \begin{array}{ll} , & x(i,t) \geq x(i,t+1) \\ , & x(i,t) < x(i,t+1) \end{array} \right. \tag{15}$$

Additionally, mean μ_t and standard deviation δ of the flow count during time slot t are computed, and a difference between the mean and the fast entropy is estimated using (16) [67]:

$$D(i,t) = \left| \mu_t - \overset{\frown}{H}(i,t) \right| \tag{16}$$

The adaptive threshold algorithm then raises an alarm by checking the difference $D(i,t)$, as shown in (16). If $D(i,t) > \beta\delta$, it is assumed that attack traffic will occur, and a DDoS attack alarm will be raised; otherwise, normal traffic will occur, and the value of is updated. The adaptive threshold β value takes the following form [67]:

$$\beta = \begin{cases} \beta - 1 & if \quad \overset{\frown}{H}(i,t) < 0.5\mu_t \\ \beta & if \quad 0.5\mu_t \leq \overset{\frown}{H}(i,t) < 1.5\mu_t \\ \beta + 1 & if \quad \overset{\frown}{H}(i,t) > 1.5\mu_t \end{cases} \tag{17}$$

The results obtained demonstrate that using a threshold that is adaptively adjusted based on the conditions of the traffic pattern increases detection accuracy. However, the processing time increases. Additionally, the efficacy of this strategy is called into doubt when numerous slow-rate DDoS attacks with various source IP addresses surface. An updated study by the same author is presented in [68]. In this study, DDoS attacks were detected using dynamic thresholding on flow-based features. Different traffic features were extracted in relation to packet amount, source IP, destination IP, and protocol, and then four attributes were calculated based on DDoS characteristics. Experimental observation showed that, during a DDoS attack, the estimated attribute values are extremely high. The entropy of the four attributes is compared to a threshold value, and a DDoS attack is considered to have occurred when it surpasses the threshold. The threshold is estimated in a similar manner to the research study in [67]. The threshold values are updated on a regular basis and change depending on the state of the network. Though the false positive rate was not examined, the method has a relatively high detection rate.

Most entropy-based attack detection studies that have been conducted so far [61–67] rely on a few entropy-based features, which may limit the type of attack that may be detected as well as the accuracy. To overcome this issue, some other studies [69–71] have thought about using multi-entropy features. Winter et al. [69] estimated the entropy across five flow parameters, including source and destination IPs, ports, and packets-per-flow. The outcomes demonstrate that the suggested approach can identify large alterations in network entropy time series. Although multiple features were considered, the mix of features employed is still straightforward and not comprehensive enough for practical application. Qin *et al*. [70] also utilized entropy vectors of different features from traffic flow for attack detection. The use of more thorough features to build clustering models makes this approach different from the studies in [69]. Additionally, based on the traffic models, a detection threshold was automatically created. Experimental results proved that the suggested approach is adaptable to real-world environments and has higher detection accuracy. Although the detection speed is poor, the accuracy of the detection improves when the data scale exceeds 4000. Furthermore, it is impossible to pinpoint the rationale behind the choice of feature thresholds. Koay et al. [71] introduced a set of new entropy-based features, including source and destination IPs, ports, and protocols. Following that, a multi-classifier system (see Figure 10) was built using a set of various entropy-based features. The entropy of each traffic feature was computed for a 60 s interval. Regular and entropy variation features, as seen in Figure 10, were two different forms of entropy-based features that were computed. While the latter was obtained using the fluctuation of two different regular entropy features following a Lyapunov exponent separation, the former was computed using the entropy of raw traffic features.

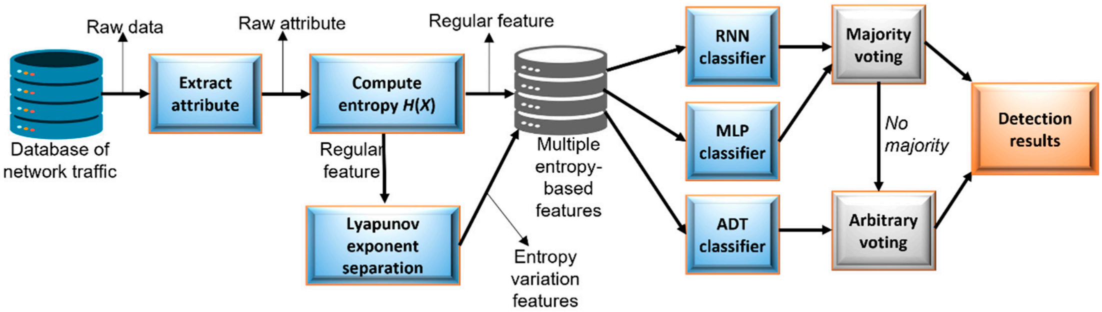

Figure 10. Framework for multi-classifier entropy-based features for DDoS attack detection, adapted from [71].

The performance of the approach was assessed using ISCX2012 and DARPA'98 datasets with a sensitivity of 94.7%. Although the sensitivity results are thought to be superior, the dataset utilized comprises obsolete DDoS attack vectors, making it uncertain whether the method can be applied to identifying modern attacks. An assessment of the traffic-based features used in an entropy-based approach is presented in [72]. The results presented revealed that a better approach must be adopted for choosing traffic features. The ability to detect anomalies is distinct and frequently enhanced by behavioural distributions that are qualitatively different from port and address distributions.

A method for identifying traffic-based attacks using UAV and Wavelet Packet Energy Entropy (UWPEE) is suggested in Xie et al. [11]. The wavelet packet energy entropy is used in the UWPEE system to identify attacks, while UAVs are sent to collect the real traffic from IoT devices. In this method, the traffic sequence is partitioned into multiple layers of wavelet packets, and the wavelet packet coefficients of each layer are then reconstructed to reveal the sequence's influencing factors. The energy entropy is then calculated to determine if the traffic data exhibit distinct properties at various scales. A traffic signal with a higher degree of order has a lower entropy value than one with a higher degree of disorder [11]. Entropy can, therefore, reflect the distinct traits of malicious nodes when they periodically emit fake packets. The experimental results show that the UWPEE scheme can effectively identify traffic-based attacks with an accuracy rate of 84.47% and an average recognition efficiency of 4.89 for malicious nodes. Meanwhile, compared with the greedy algorithm, the flight path of the UAVs is reduced by 15.44%. In [73], a threshold-based detection scheme was proposed to detect RREQ flooding attacks in mobile ad hoc networks (MANET). In this study, the throughput, packet delivery fraction, and end-to-end delay of network traffic were compared with legitimate network traffic (without flooding attacks) and a network with one or more flooder nodes. A sender node is regarded as normal if its rate of RREQ falls below a certain threshold; otherwise, it is considered malicious. Simulation results indicate that a flooding attack could be detected, although the effectiveness depends on the threshold value chosen. Additionally, the method experiences more false positives and misdetections due to seasonal fluctuations in network traffic.

3.3.2. Queue Modelling-Based Detection Methods

In this approach, a multidimensional algorithm is used to analyse how networking components process traffic based on traffic theory. Since DDoS attackers aim to engulf servers' resources and prevent legitimate clients from accessing them, a good queue management algorithm enables the system to manage access to a fixed amount of bandwidth by identifying which packets should be transferred and which ones should be dropped when the queue limit is fully occupied. In the queuing model, the memory of a server is assumed to be fixed [74]. It is then easy for an attacker to launch the attack and somehow disable the server, preventing it from providing the service to its legitimate user. A simple queue system is illustrated in Figure 11.

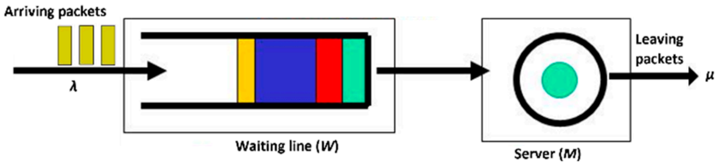

Figure 11. A simple queue system.

In Figure 11, λ represents the arrival rate of packets at the queue, with a waiting time of W, until they receive no response from server M, while μ indicates the system's service rate. This system is based on Little's law [75], which is expressed as shown:

$$L = \lambda \times W \tag{18}$$

The expression in (18) describes the average number of packets in a queue. DDoS attacks try to clog up the system's queue so that legitimate users cannot obtain service. By imposing sophisticated computational processes on the victim device, DDoS attacks can extend the time it takes to process packets or increase their service rate [76]. This attack scenario can be evaluated using queueing theory, which estimates the likelihoods of bandwidth, memory, and CPU exhaustion. In [77], the probability of bandwidth exhaustion P_b is represented by a M/G/K/K queue model. In this model, P_b is estimated using (19) [77]:

$$P_b = \frac{\frac{\rho^k}{k!}}{\sum_{j=0}^{k} \frac{\rho^j}{j!}}, \quad \rho = \frac{\lambda_b}{\mu_b} \tag{19}$$

In (19), k indicates the number of communication channels between the attacker and the target server, λ_b is the arrival rate of packets, which determines the attack intensity, μ_b indicates the service rate, and ρ denotes the utilization factor for the queue system. When the DDoS attack exhausts the CPU of the target server, the probability of CPU consumption is represented by a simple M/M/1 queue model, described by (20) [77]:

$$P_C = \begin{cases} 1, & \frac{\lambda_C}{\mu_C} \geq 1 \\ \frac{L}{t_w}, & \frac{\lambda_C}{\mu_C} < 1 \end{cases} \tag{20}$$

In (20), L indicates the time the attack spent on the network, t_w indicates the amount of time a legitimate client is prepared to wait to be served, and λ_c and μ_c are the arrival and service rates due to the CPU exhaustion. The total depletion probability is obtained by evaluating the likelihood attributable to buffer exhaustion. The result of the simulation shows that the attack probabilities increase as the arrival rates increase.

In most of the queue strategies for attack detection, such as drop-tail, random early detection (RED), and nonlinear random early detection (NLRED) [78], a pre-defined value is set for the maximum length of the queue. Newly arriving packets are discarded when the length of the queued packets exceeds the set threshold. In this queue approach, all the traffic packets are considered equal, regardless of the traffic type. The attacker will then send fewer TCP packets before waiting for the target server to respond because of packet loss. Consequently, the TCP session's throughput will decline [79]. In most queue modelling studies [77–82], Poison distribution is used to describe packet arrival according to a random process. According to Singh et al. [74], for traffic analysis, the queue must support exponential data, and requests must be processed using a first-come-first-served queuing analogy with a single server and obviously finite buffer state. Using this concept, a collection of data patterns was generated, and UDP floods were detected. In [80], a framework to identify DDoS attacks using the packet flows of particular protocols was presented. In this study, the normal behaviour is estimated using a Gaussian parametrical mixture model, while the attacks are detected using a queue model. The results show the approach is effective with reasonable detection accuracy. Khan and Traore [81] analysed the effects of attacks on variables such as queue growth rate using a standard M/M/1/K queue model with round robin discipline. The given results demonstrate that the queue growth rate linearly increases as the frequency of flooding attacks increases. The authors in [82] presented the use of the queueing model for network router attack detection. In this study, the traffic congestion due to attack packets can be readily noticed at locations near the target rather than the attack sources; consequently, it is anticipated that the technique will have a comparatively higher false negative rate. In [83], a queue scheme was developed for detecting malicious attacks. In this study, the arrival requests are provided with a queue service at a base station that oversees assessing the forwarded packets. Once the traffic is backed up for an extended period, malicious attacks are discovered.

In [84,85], the effectiveness of queuing management mechanisms under DDoS attack detection were evaluated. Five distinct queuing algorithms—drop-tail, RED, deficit round robin (DRR), fair queue (FQ), and stochastic fair queue (SFQ)—were tested for how UDP flooding affected their performance in [84]. The study demonstrates that SFQ outperforms the other queuing mechanisms for UDP traffic. Recently, using NS2 software, Wei et al. [85] evaluated the effectiveness of drop-tail, RED, and REM queue management mechanisms on ad hoc networks under attack. This study evaluated the performance of the three mechanisms under small-, medium-, and large-scale DDoS attacks based on the packet rate and average end-to-end latency. Simulation results revealed that drop-tail was less effective at detecting medium- and small-scale DDoS attacks than REM and RED. However, all three mechanisms showed inadequate detection abilities when subjected to large-scale DDoS attacks.

3.3.3. Statistical-Based Detection Methods

This approach analyses the statistical features of normal traffic to create a baseline traffic pattern. Any incoming traffic that falls outside the baseline is judged to be malicious traffic. This approach processes network traffic using sophisticated statistical algorithms and differentiates anomalous traffic from legitimate patterns of established network traffic. With the statistical technique, expected behaviour can be inferred from observations without any prior knowledge of the target system's typical operations. This can potentially lead to more accurate detection of malicious activity. Statistical algorithms used for DDoS attack detection may include, among others, statistical forecasting and time series methods, as shown in Figure 8.

***Chi-squareapproach*:** The Chi-square (χ^2) is a test of independence used to determine if two categories of variables are connected to one another. Given the overall frequency of each category, it looks for patterns in these observations to determine whether any combinations of the categories occur more frequently than would be predicted by chance. A very small value of χ^2 indicates a good correlation between the actual and expected values, whereas a large value implies that the actual values do not closely match the anticipated values. This approach has been used in several research studies [86–89] for anomaly detection in internet-based networks. Ref. [86] tested it to assess the prevalence of TCP–SYN flag values and protocol numbers. In method in this study, service ports are examined using the Chi-square method while considering HTTP, FTP, and DNS. Similarly, packet lengths are binned into ranges. If there are N numbers of incoming traffic packets while B represents the available bins, the amount packets with values within the i^{th} bin is represented by N_i, and n_i denotes the anticipated number of packets in this bin based on the usual distribution. Thus, χ^2 is estimated using (21):

$$\chi^2 = \sum_{i=1}^{B}\left(\frac{(N_i - n_i)^2}{n_i}\right) \tag{21}$$

Abouzakhar and Bakar [87] used the same expression for attack detection by analysing RST, SYN, ACK, and ICMP packets. The proposed method consists of a database storage block, a Chi-square test block, a feature extraction and distribution block, a distribution and categorization block, and a decision-making block, as shown in Figure 12.

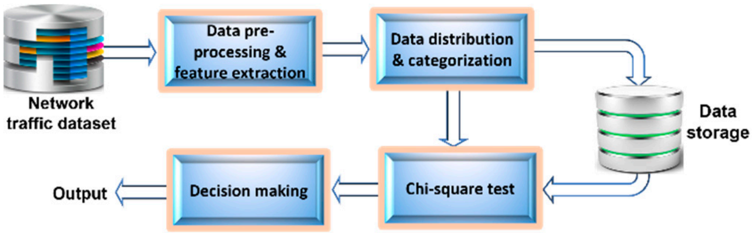

Figure 12. Framework for DDoS attack detection using Chi-square approach.

Firstly, TCP flags are extracted for each input packet from the network traffic dataset. Under the data distribution and categorization block, packets are distributed and categorized into the number of RST, SYN, ACK, and ICMP packets per second, along with other TCP packets. After categorization, a Chi-square approach is employed to carry out anomaly detection, where a χ^2 value is estimated using (21) and compared to a tabular χ^2 value. When there is a large difference between these two values, an intrusion alert is triggered.

Leu and Lin [88] used the goodness-of-fit test of the χ^2 approach to detect attacks. The method examines the number and variation of packets sent from sources, as well as IP address distribution statistics. When an attacker floods the system with many packets from random source IPs, the approach estimates its Chi-square value and checks to see if it exceeds a predetermined threshold to trigger an attack alarm. Experimental findings demonstrate the approach's capability to quickly identify DoS and DDoS attacks. Other studies reporting the use of χ^2 value for attack detection may be found in [89]. In [89], χ^2 was estimated based on moving averages while considering how frequently events appeared in the Solaris BMS audit record. The results obtained showed that the χ^2 values based on the moving average were sufficient to detect anomaly attacks.

***Statistical forecasting models*:** A conventional statistical forecasting model makes predictions about future occurrences using statistics derived from historical data. Using historical data to analyse and observe past network traffic patterns, this method forecasts future observations. A plethora of statistical forecasting models have been developed for attack detection. Moving average (MA), weighted moving average (WMA), simple exponential smoothing (SES) or exponential WMA, double exponential smoothing (DES), and triple exponential smoothing (TES) are a few examples of this. Each of these models has its own accuracies and deficiencies. MA is a smoothing technique that observes the underlying pattern of a data set to forecast future values. While SES, EWMA, and DES consider both historical observations and historical forecasts, MA and WMA base their forecasts solely on prior observations. The authors in [90] revealed that the EWMA models could be used for detecting rapid changes in event intensity when demonstrated on the publicly available DARPA dataset. In [90], an adaptive threshold algorithm based on the EWMA model was developed for detecting SYN flooding attacks. Real traffic traces were employed to analyse the effectiveness of the algorithm. A satisfactory result was observed when high-intensity attacks were considered. However, the algorithm performs terribly when handling attacks of low intensity. Similar to the studies presented in [90], Machaka et al. [91] assessed the use of the EWMA algorithm for DDoS attack detection in IoT infrastructure. A high detection rate was achieved with a 40 s delay when demonstrated on an artificially generated dataset for a high-rate attack. While the detection rate of this approach is relatively high for attacks with high intensity, its performance deteriorates for attacks with low intensity. The use of an adaptive fusion of multiple characteristics (MAF–ADM) for the detection of low-intensity attacks was suggested by Zhan et al. [92] as a solution to this problem. Under a low-intensity attack, the time-frequency joint distribution of the legitimate TCP traffic changes; therefore, several statistical features of this distribution were selected to create isolation trees. The potential to isolate samples containing low-intensity attacks was then combined to create an anomaly score. The anomaly score was smoothed using a WMA to lower the potential number of errors that may result due to noise in the network traffic. The result shows that the method can effectively detect low-intensity attacks with a relatively low false negative rate when demonstrated on the WIDE2018 and LBNL datasets. The approach has two shortcomings. First, neither of the two datasets had evidence of low-intensity attacks; instead, this was simply assumed to exist. Second, the extraction of features requires very high data processing expenses and is time-consuming. These two drawbacks constrain its use for real-time online detection of low-intensity attacks. The authors in [93] used SES and wavelet analysis to track incoming bytes, packet counts, and the ratio of incoming to outgoing packets to detect UDP flooding DDoS attacks. This approach detects multiple attack scenarios without producing any false positives. The application of TES for TCP–SYN flood and slammer worm detection was reported in [94]. In this investigation, the traffic packets' source IP, destination IP, and ports were examined within a 900 s interval. The effectiveness of the approach was verified using the Brazilian National Research and Education Network dataset, which has 5 days of network traffic. Results indicated that the approach was successful in identifying TCP–SYN flooding. For this approach, the false alarm rate and detection accuracy were not assessed.

Some other time series models, such as auto-regressive (AR), autoregressive integrated moving average (ARIMA), and linear regression model (LRM), are reportedly used for DDoS attack detection. Zhang et al. [95] used the ARIMA framework to identify DDoS attacks via the NS2 simulator. Yaacob et al. [96] introduced a novel algorithm with the use of the ARIMA technique to detect possible attacks that may occur in computer networks. Their approach offers the network administrator a means for early warning. In [97], a combination of ARIMA and a chaotic system was used to detect attacks, with a true positive rate of 94.4%. Additionally, false positives and false negatives of 0.1% and 5.6%, respectively, were recorded. The authors in [98] looked at the relationship between the average and standard deviation of the network traffic throughput to evaluate DDoS attacks. The research demonstrates that, in non-attack situations, the rise in standard deviation caused by a traffic surge increases the average network throughput, as seen in Figure 13. However, in a DDoS attack scenario, the standard deviation is not affected by the increased network throughput because of the attack. This hypothesis was used to produce an attack detection method with linear regression. The efficacy of the developed approach was confirmed using the CAIDA dataset. The results obtained revealed that DDoS attacks may be accurately identified with a low proportion of false positives.

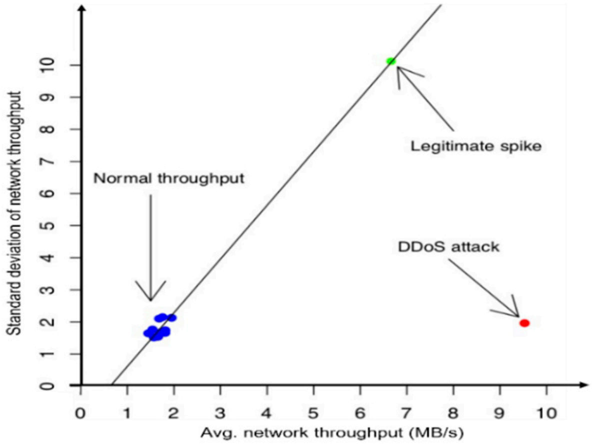

Figure 13. Correlation between the mean and standard deviation of traffic throughput, adapted from [98].

In [99], the authors used several forecasting algorithms, including MA, WMA, EWMA, and LRM, to predict the intensity of SYN, DNS, and ICMP floods. For the predictions in this study, a window size of 60 s was established to track the number of packets. The results for the three attack types demonstrate that LRM best detects the magnitude of TCP–SYN flood, while EWMA is best at detecting the attack intensity of DNS, TCP and ICMP floods. The algorithms are said to produce an overall error rate of 1%.

The research studies presented have shown that forecasting models can be used for attack detection with a relatively high level of accuracy. The inherent errors generated by these models are a significant disadvantage. Every forecast model generates a succession of errors between the predicted and overserved values. Accordingly, if the network traffic data contains one forecast for each feature, then the total errors from all the features amount to the following:

$$e_T = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{22}$$

where e_i is the error generated by the n-feature. Therefore, significant forecast errors are anticipated, given the intermittent nature of internet traffic. In addition, because the models depend on prior statistical features, they may not work well with actual internet data series [100].

Other statistical approaches are also reported in the literature. These range from the use of a simple *t*-test to a more sophisticated cumulative sum (CUMSUM) approach.

Additional reported techniques are the statistical segregation method (SSM), multivariate correlation analysis (MCA), and analysis of covariance, among others. In the two-sample *t*-test proposed by Chen [101], the authors examined the statistics of the normal SYN arrival rate (SAR) by sampling the input traffic flow. Then, the dissimilarity between the number of SYN and ACK packets as well as the dissimilarity between the arriving SAR and normal SAR are estimated to ascertain the occurrence of an attack. Simulation results revealed that the proposed method has a quick detection rate and a low likelihood of both false positives and false negatives. Additionally, computational overhead is not too high and, in the event of a sudden shift in traffic, can detect DDoS flood attacks. In low-traffic areas, it might, however, miss an attack.

The CUMSUM method estimates the cumulative sum of the difference between an input sequence's actual and expected values, which is then compared to a threshold value [102]. A CUMSUM value above the threshold signifies a change in statistical features in the network traffic over time series. This method is used to estimate variations in traffic features. The authors of [103] suggested a simple method for detecting SYN flooding attacks using the non-parametric CUMSUM model. The results of the simulation showed that TCP–SYN flooding may be accurately detected with a low false alarm ratio and a high detection ratio. In the research study presented in [102], the authors investigated the use of CUMSUM for detecting DDoS attacks in an IoT network. The algorithm has a good detection rate for high-rate attacks but is poor for low-intensity attacks. A combination of CUMSUM and an entropy-based method is reported in [104]. The CUMSUM is used in this work to handle traffic entropy. For the observed entropy data, additional signal processing, utilizing wavelet pre-filtering, is used. This helps to improve detection efficiency over other CUMSUM approaches that rely solely on the entropy of the packet header field without further processing. The findings revealed high detection accuracy and a low proportion of false positives. The authors in [105] investigated the potential of SSMs to minimize false detection during DDoS attacks. With this method, the traffic flow is sampled at regular intervals to identify the distinctions between legitimate and malicious traffic. After that, correlation analysis is used to separate attacks from legitimate flows by comparing the samples to pre-specified attack state conditions. Evaluation results reveal a mix of segregation methods that could significantly lower the likelihood of false detections during DDoS attacks. The results also demonstrate higher detection latency and increased computational overhead. Tan et al. [106] used MCA for attack detection. In this study, the patterns of legitimate network traffic are examined, and the traffic is then classified by obtaining the geometrical correlations between network traffic parameters. To expedite the MCA procedure, a triangle-area-based approach was also added. The effectiveness of this method was assessed with the KDD Cup'99 dataset. The approach has excellent performance with 99.95% detection accuracy. However, some poor performance was observed when the false positive rate was analysed. The proposed approach could be further validated on some other dataset with the updated DDOS attack vectors. More advanced categorization algorithms and the use of real-world data would also reduce the false positive rate. In [107], the analysis of covariance was proposed. The method's efficacy in detecting SYN flood attacks has been established. The approach has relatively high detection accuracy. The approach does, however, have certain drawbacks. There is no theoretical justification for the high detection rate. The method also faces the difficult task of choosing an adequate observed time window for the covariance analysis. An improvement on the covariance approach is presented in [108]. In this study, a covariance criterion was used to generate a profile of typical network traffic and identify anomalous activities. Then, a decision-making rule that considered all the data in the covariance matrix was integrated using the Chebyshev difference. The results show that the detection rate improved. However, because there is so much data to handle in the covariance matrix, the approach has a huge computational complexity problem. Peng et al. [109] use a sequential-non-parametric change point for detecting bandwidth attacks. The approach involves tracking a rising number of new IP addresses, followed by a statistical analysis of the incoming traffic over a period to determine the typical time interval. The arrival rate of new IPs is then compared to the normal value, and, when this exceeds the average arrival time, a bandwidth attack alarm is generated. Although the technique records a relatively good detection rate, a significant drawback was noticed. Since the detection method can only identify an attack when there is a dramatic change in the volume of existing network traffic, a spoof IP address will bypass the detection approach.

The use of feature–feature score (FFSc) for DDoS attack detection was suggested by Hoque et al. [110]. In this method, the behaviour of network traffic was examined using three fundamental network traffic parameters: packet rate, entropy of source IPs, and changes in source IPs. Next, a similarity value is calculated for each network traffic sample using the FFSc. The FFSc is contrasted against a pre-defined threshold, and, if it is above the threshold, an attack alarm is raised. The viability of the approach was verified using CAIDA and DARPA datasets. It was noticed that the pre-defined threshold greatly affects the detection accuracy. The accuracy of the method substantially declines as the detection threshold increases. Thus, the choice of the detection threshold is a major concern. Continuous rand probability score (CRPS) has also been used in recent years to distinguish between legitimate and attack traffic. The CRPS is primarily employed to assess the correctness of a statistical forecasting methodology [111]. It is presently utilized for anomaly detection since CRPS can compare a whole distribution with an observation [112–114]. To use the CRPS for DDoS attack detection, the CRPS is generated for every traffic measurement. Each incoming traffic network measurement is then contrasted with the traffic distribution under no attack. It is assumed that, in an attack-free network, the traffic distribution is Gaussian [112] with mean μ and variance σ^2. The CRPS is computed using (23) [112,113]:

$$CRPS\left(N(\mu,\sigma^2),x\right) = \sigma\left[\frac{x-\sigma}{\sigma}\left(2\Phi\left(\frac{x-\sigma}{\sigma}\right)-1\right)+2\varphi\left(\frac{x-\sigma}{\sigma}\right)-\frac{1}{\sqrt{\pi}}\right] \quad (23)$$

where Φ and φ are the Gaussian probability and cumulative density functions, respectively. The monitored traffic is considered normal when the CPRS exhibits values that are very small and close to zero. Higher CRPS values, however, point to the existence of malicious traffic in the network traffic being monitored. With these metrics, it is feasible to know which traffic is legitimate or an attack. The CRPS results are subjected to an exponential smoothing approach by Bouyeddou et al. [112] to set a decision threshold and increase the existence of attack traffic. The proposed approach performed well when demonstrated on ICMPv6 and DARPA datasets, with a 100% detection rate. The authors argued that the proposed attack detection method is for a single timescale and may be inappropriate to identify malicious activities at different scales. Motivated by [112], Sharma et al. [114] used a similar procedure to detect attacks in fog-enabled IoT, with better detection accuracy when validated using the DARPA'99 dataset.

3.3.4. Attack Detection Methods Based on Chaos Theory

Chaos theory is the mathematical study of nonlinear phenomena that are challenging or practically impossible to anticipate. As mentioned earlier, every forecasting technique will inevitably produce some errors. Hence, the resultant error needs to be carefully examined. It is possible to know if the error generated exhibits chaotic behaviour or not by computing the Lyapunov exponent, using (24) [115]:

$$\lambda_i = \lim_{x\to\infty}\frac{1}{t_i}\ln\left|\frac{\Delta X_i}{\Delta X_0}\right| \quad (24)$$

A positive Lyapunov exponent reveals the existence of chaotic behaviour in the prediction error values, which is an indication that the forecast prediction is significantly different from the typical observed values. Because there is little variation between the predicted and actual observations, a negative value indicates that the error is not chaotic. In chaos-based DDoS attack detection, Lyapunov exponents are used to know whether an event is normal behaviour or an attack [116–121]. Chonka et al. [116] provided one of the early efforts in this field. To ascertain whether there was attack traffic, the authors looked at the average exponential rate of dispersion across two nearby orbits (normal and new traffic). This work used (24) to compute and analyse the Lyapunov exponent. DDoS attack detection is made possible by (25):

$$\lambda_i = \begin{cases} > 1 & \Rightarrow DDoS \quad attack \\ \leq 1 & \Rightarrow Normal \ traffic \end{cases} \tag{25}$$

If $\lambda_i > 1$, it demonstrates how chaotic and unstable the network traffic orbit is. As a result, the closest points disperse at any arbitrary range. This indicates that an attack is the cause of the change in traffic. This network activity is categorized as DDoS attack activity. In the absence of an attack, the orbits are drawn to a fixed point from where they dispersed; therefore, the traffic change is caused by new legitimate traffic rather than an attack. The study recorded a detection rate ranging from 88% to 94%, with a lower false positive rate, from 0.455 to 0.05%, when demonstrated on the DARPA dataset.

Similarly, in [117], chaos theory and Lyapunov exponents were employed to detect attacks in cloud computing environments. Inspired by [116], the study records an average detection rate of 89% with 11% false positive rates. The authors of [118] presented DDoS intrusion detection via network traffic prediction and chaos theory. To train a neural network to detect anomalies caused by either bursty legitimate traffic or DDoS flooding attacks, results from a local Lyapunov exponent are used. Computer simulations conducted on the DARPA network traffic dataset show that a detection accuracy of 93.75% was recorded. The accuracy is marginally higher than that noted in [116,117]. There was no report on the evaluation of the false alarm rates. The research provided in [119] is comparable to that in [118]. However, in this study, a change in the Lyapunov exponent is suggested for detecting anomalies in the network traffic. Chaotic analysis is performed on the entropy of source and destination IPs to achieve attack detection. This method performed better than the previous studies, with a record true positive rate of 98.56% when demonstrated on the DARPA'99 dataset. In [120], the chaos-based hypothesis for DDoS attack detection is validated. The study involves the prediction of network traffic using SES. Thereafter, prediction errors were evaluated with chaos theory and a back propagation neural network. The results indicated that a detection accuracy of up to 98.04% can be achieved using the chaos hypothesis. The authors in [89] presented a similar procedure for detecting DNS amplification. A Lyapunov exponent is determined over a window size. Following that, network traffic is categorized as either normal or abnormal, using the exponent analysis described in [116]. This study records a lower detection accuracy of 66%. Comparing this accuracy to the earlier research mentioned in [116–120], it is significantly lower. Nevertheless, the results from this study revealed that a smaller window of packets offers enough information to identify a DDoS amplification attack. This will, however, increase the cost of processing power. To improve the accuracy of the chaos method, Procopiou et al. [121] combined a forecasting approach with chaos theory for the detection of attacks in smart home networks. Additionally, using a pre-determined window size, the Lyapunov exponent hypothesis was used to identify whether the incoming traffic was legitimate or anomalous. Simulation results show that a detection accuracy of 94.3% was achieved with a true positive rate of 87%. The algorithm, however, only records a precision value of 81%, which restricts its practical usage. Additionally, when the low-rate DDoS attack was considered, additional false positive rates were produced. The strategy is also complex for a typical smart home setup.

3.3.5. Heuristic-Based Detection System (HBDS)

In this method, detection threshold decisions are made by using algorithmic logic to analyse the statistical features of network traffic. The network traffic is adaptively fine-tuned, and the detection thresholds are optimized using an optimization model to minimize false positives and false negatives. To obtain better results, the HBDS heavily relies on the use of a classification method in conjunction with optimization. Recently, studies have investigated attack detection and classification, employing both machine learning and optimization methods. Due to the high computational cost incurred by most HBDSs, Kumar and Selvakumar [122] suggested an adaptive hybrid neuro-fuzzy system for DDoS attack detection with substantially reduced cost. In this study, the adaptive neuro-fuzzy inference system (ANFIS) was employed as the basis classifier. The approach is suited to real-time network datasets, since it can handle both discrete and continuous features in a dataset. The proposed approach also records a detection accuracy of 99.2% using fewer statistical features from the traffic pattern in the dataset. However, it has been demonstrated [61–67] that using a multi-feature will improve the precision of any detection method against a variety of attack features. Therefore, the use of less advanced features may restrict its application to current and next-generation attacks. Furthermore, some false alarm cases were recorded. In [123], the authors integrated a multi-objective optimization method with a convolutional neural network (CNN) for DDoS attack detection in IoT networks. In this study, CNN's LSTM deep learning technology was combined with the Jumping Gene-adapted NSGA-II multi-objective optimization method to categorize network traffic as attack or legitimate. The applicability of this approach was demonstrated on the CISIDS2017 datasets, with 99.03% accuracy and a reduced training time. The records of the false positive rates were not analysed. The authors of [124] went on to demonstrate that multi-objective feature selection can increase attack detection accuracy while having a very low false alarm rate. One of the biggest problems with most attack detection studies is the feature selection of network packets, which may result in some false alarm rates. In [124], a multi-objective optimization problem was developed to improve on the earlier challenges. The optimization problem was solved using a non-dominated sorting genetic algorithm with an adapted jumping gene. Thereafter, a machine learning model was employed as a classifier to select features. The method records a detection accuracy of 99.9% with just 6 features and a runtime of 0.02 s, which indicates that the detection speed is better. Despite the better detection speed, the authors in [125] used 10 more features to attain the same accuracy. Additionally, there was no discussion of the feature selection in this study. The results of the false positive and false negative rates were also not discussed in either study. In [126], the authors attempted to solve DDoS attack detection feature selection problems with an intelligent wrapper feature selection model that incorporates binary-particle swarm optimization. Out of the 76 features in the CICIDS-2017 dataset, 19 were the best chosen. Various classification techniques were used to train and assess these features. Among the classifiers investigated, the decision tree classifier gave the best performance, with the highest accuracy of 99.52%.

The research study in [127] was also based on analysing traffic features and optimally selecting the best features for DDoS attack detection in cloud-based networks. This method gathers service requests from users, groups them as log information, and then uses the Bhattacharya distance metric to classify the most significant features. Following that, a Taylor–Elephant Herd Optimization was developed to select the features optimally. The selected features were used to train a deep belief network. A detection accuracy of 83% recorded by this approach does not guarantee its use for real-time attack detection. Similar to the research study in [127], Varghese and Victor Jose [128] presented an improved radial bias function neural network where flow-based features, and higher-order statistical features with improved holoentropy were extracted from the CICDDoS2019 and UNSW-NB-15 datasets. This approach has a detection accuracy of 90.86%. However, the false positive and negative rate results are deemed too high for practical applications. Studies in [129] proposed the integration of support vector machines (SVM) with hybrid Harris Hawks optimization to achieve 97.05% detection accuracy on the NSL–KDD dataset. In [130], a vector convolutional deep neural network was optimized using binary and real cumulative incarnations. The study recorded different accuracy levels for the datasets used. For the NSL–KDD dataset, an accuracy of 99% was observed. However, for the KDD Cup'99, the accuracy dropped to 97.5%. A relatively high error rate was observed when demonstrated on the KDD Cup'99, which justified the reason for reduced accuracy.

In Table 7, a summary of some of the existing research studies on anomaly-based approaches, focusing on chaos theory, the queuing model, and statistical approaches for DDoS attack detection, is presented. The comparison was conducted using the method, the dataset, and the application domain. Additionally, a summary of the evaluation results in each study is presented. The findings indicate that these methods rely heavily on the application of a detection threshold to increase accuracy. They also experience feature selection problems.

Table 7. Summary of some notable studies on anomaly-based methods for DDoS attack detection.

Study	Method Used	Description	Application Domain	Dataset	Results
[11]	Entropy (UWPEE)	• The energy entropy is estimated to know if the traffic data exhibit distinct properties at various scales.	UAV network	Real	• The scheme could identify traffic-based attacks with an accuracy rate of 84.47%.
[66]	Joint entropy metrics	• A three-module detection system. • Estimate and compares entropy against a threshold.	IoT	DARPA'99 and CICDDoS2019	• Achieve 90% detection rate. • A static threshold utilized limits its applicability to real-life packets.
[74]	M/M/1 queue theory	• Investigate the potential of queuing theory for attack detection. • Model attack based on first-come first-served queuing analogy having single server.	IoT	Simulated	• Detect UDP flood. • Records 46% packet loss during detection.
[81]	M/M/1/K queue	• Examine the effect of attacks on queue growth rate.	SDN-fog computing	ISCX2012, and real data	• Queue growth rate rises gradually as the frequency of attacks increases.
[87]	Chi-square	• Has data processing and feature extraction, data distribution and categorization, and chi-square test modules.	IoT	CAIDA	• The distribution and categorisation of the input data have an impact on performance.
[91]	EWMA	• Assess EWMA for DDoS attack detection.	IoT	Traffic data from the MIT Lincoln Laboratory	• Record high detection rate. • Achieve false positive rate less than 40%.
[100]	Chaos theory	• Employ Lyapunov exponent analysis to categorize network traffic as normal or malicious.	–	DDoS amplification dataset	• Record an accuracy of 66%.
[101]	Two-sample *t*-test	• Statistics of normal SYN arrival rate were investigated.	IoT	Simulated	• Fast detection rate. • Relatively low probabilities of false positive and false negative rates.
[103]	CUMSUM	• A lightweight approach for TCP–SYN flooding attack detection.	IoT	DARPA'98/'99	• High detection ratio. • Record 2.46% false alarm ratio.
[105]	SSM	• Filters the traffic flow at regular intervals to identify malicious and legitimate traffic.	IoT	CAIDA	• Reduce false detection rate. • Add more computational overheads.

3.3.6. Machine Learning-Based Detection Methods

This approach involves the use of algorithms to identify malicious traffic from a pool of network traffic simply by learning the characteristics of the network traffic. After learning the characteristics of traffic features, these algorithms may develop an extremely accurate model for identifying these features. Several machine learning algorithms are employed for attack detection. In this section, we investigate attack detection studies in the IoT and some other emerging networks.

Attack detection in IoT and SDN using machine learning**:** The research work discussed in [22] demonstrates the use of ANN for DDoS attack detection. Based on distinctive patterns that distinguish DDoS attacks from normal traffic, the ANN can identify TCP, UDP, and ICMP DDoS attacks. The efficacy of the approach was compared to backpropagation (BP), Chi-square, SVM, and Snort based on accuracy, precision, and sensitivity. The results demonstrate that it outperforms its competitors (BP, Chi-square, SVM, and Snort), with a detection accuracy of 98% attained. Additionally, the method records a precision value of 100% and a sensitivity value of 96%. As the accuracy only totals 98%, the precision value of 100% still needs further substantiation. This method has a significant flaw in that it cannot detect DDoS attacks when the protocol headers are encrypted using any encryption scheme. Alshamrani et al. [131] suggested employing SVM to detect attacks in an SDN-based network. This method involves the routine collection of network packets, from which 24 features are extracted. SVM is then used to categorize these features and find abnormalities. The method was validated through the NSL–KDD dataset, and its performance was compared to that of the J48 and Naïve Bayes (NB) classification methods. The approach has a detection accuracy of 99.4%, compared to 99.75% and 95.87% for the J48 and NB algorithms, respectively. It can be shown that the J48 classification method continues to perform better in terms of accuracy than the suggested approach. The method also has a significant processing overhead. The study in [132] uses SVM to classify additional traffic features that are periodically obtained from a flow table. These are aggregated features that pertain to DDoS attacks. They include the speed of the source IP and port, the speed of flow entries, the standard deviation of the flow bytes and packets, and the ratio of pair–flow. The validity of the approach was verified by simulation. Evaluation results show that a detection rate of 95.24% was achieved, even with a small amount of flow data. The method does, however, record some false alarms. The average false alarm rate generated was 1.26%.

The effectiveness of SVM and deep feed forward (DFF) algorithms for detecting DDoS attacks in IoT networks was examined in [133]. The effectiveness of these algorithms was analysed using the DARPA’09 dataset. Evaluation results revealed that DFF has superior accuracy to SVM. However, the SVM method outperformed the DFF algorithm in terms of processing time. The detection accuracy recorded by the DFF is 99.63%, compared to 81.23% for the SVM. This demonstrates that DFF performs roughly 22% better than SVM. The findings also showed that the DFF has a higher computational overhead, which has a big impact on its detection speed. Rahman et al. [134] designed an SDN framework and applied four machine learning classifiers independently for the detection and mitigation of ICMP and TCP floods. J48, RF, SVM, and k-nearest neighbours (k-NN) were the machine learning classifiers considered. In this approach, a synthesized dataset having normal and DDoS traffic with 24 packet-level features was employed to assess the effectiveness of the classifiers. Aside from DDoS attack detection, a mitigation code was also developed to restrict the attackers’ switch ports for 30 s. Evaluation results show that the J48 classifier outperforms the other algorithms in terms of processing speed. While SVM and k-NN have zero errors, some cases of errors were observed with J48 and RF. In addition, there is a great deal of variation in the processing times of these algorithms. It is noted that SVM requires the most testing time, whereas k-NN requires the least training time. Similar to this, the authors of [135] concentrated on the employment of more machine learning classifiers to detect DDoS attacks. In this method, six classifiers were evaluated for attack detection: logistic regression (LR), NB, k-NN, SVM, decision tree (DT), and RF. This contrasts with a previous study [134] that simply used conventional flow features for attack detection. This approach used extended features. Evaluation results show that RF performs best in terms of detection accuracy, whereas k-NN records the worst performance. The accuracy of detection by RF is 99.76%, compared to 86.41% for k-NN. While the models can accurately detect attacks within a few seconds (specifically, less than 1 s), the probability of dropping normal traffic is also observed. Koroniotis et al. [136] investigated the application of machine learning models with recurrent neural network (RNN) and LSTM for attack detection on the BoT–IoT dataset. The features extracted from the dataset were classified into two categories. The top 10 features selected from a filter with a correlation coefficient and joint entropy make up the first category, while the second category has all 35 features in the dataset. The effectiveness of these models was evaluated based on this category. When the top 10 features were considered, the RNN performed better, with an accuracy of 99.74%, while the SVM performed the worst, with an accuracy of 88.37%. However, the SVM outperforms other models in terms of precision and processing time. When all 35 features were considered, the SVM had superior performance in terms of accuracy, precision, and processing times. The SVM records an accuracy of 99.99%. One of the shortcomings of this study lies in the dataset used. Gopalan [137] reported that the dataset is unbalanced, which may have positively affected the identification of attacks due to data bias. The studies in [138] tried to solve the issue of dataset class imbalance resulting from the use of the BoT–IoT dataset in [136]. Using the same dataset as the previous study, several machine and deep learning methods were employed to create a novel DDoS and DoS attack detection method on an IoT network. These models included RF, DT, LSTM, gated recurrent units (GRU), MLP, RNN, and SVM. To prevent feature dependencies, the binary and multiclass classifications in this study used three separate feature sets. Evaluation results show that the RF and DT are more accurate for both binary and multiclass classifications. Both versions exhibit excellent performance across the board for every feature set. For instance, when the initial feature set for multiclass classification is considered, the DT model has an accuracy of 99.95%, whereas the RF model has an accuracy of 99.92%. When binary classification is considered, the accuracies for DT and RF rise to 99.97% and 99.95%, respectively.

Chen et al. [139] employed DT for DDoS attack detection in a multi-layer IoT environment. IoT gateways, cloud servers, SDN switches, and IoT devices make up the multi-layer IoT environment, as shown in Figure 14. In this study, eight smart poles were used to build a wireless sensor network that collected sensor data across a campus. Each smart pole was equipped with an LED lamp, an access point, a camera, smart signage, a communication box, and an equipment box. The study used Wi-Fi, Bluetooth, ZigBee, and LoRa to transmit sensor data. Smart poles SP1–SP4 equipped with Wi-Fi access points connected to the internet without a backbone connection, while SP4–SP8 smart poles required Ethernet to send and receive packets. Additionally, SP2 and SP3 supported ZigBee, SP5 supported Bluetooth, and SP2 and SP8 supported LoRa. Each pole was fitted with Raspberry Pi 3, which oversaw gathering of the sensor data because it can communicate with Ethernet, Bluetooth, ZigBee (via the I^2C protocol), and Wi-Fi devices as a heterogeneous gateway.

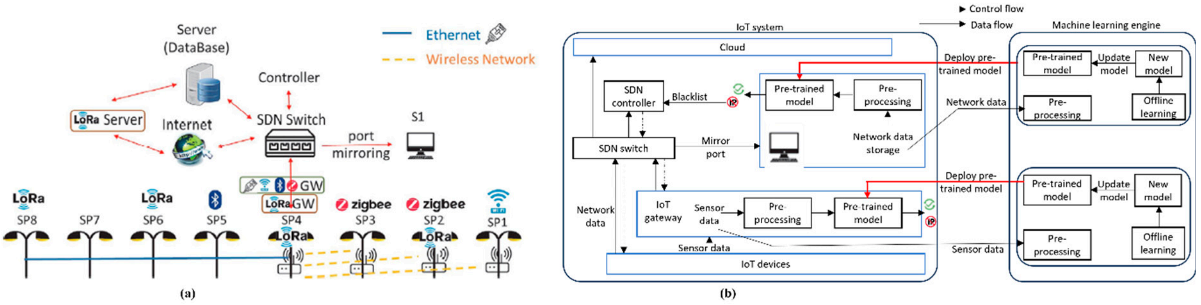

Figure 14. A typical experimental setup for DDoS attack detection machine: (**a**) architecture of the smart pole, (**b**) multi-layer IoT architecture, adapted from [139].

The IoT gateway gathered and pre-processed sensor data packets. As part of the procedure, DT classifiers were trained using the extracted packet attributes, such as packet length, timestamp, protocol, source IP and MAC, and destination IP and MAC, based on the types of DDoS attacks. The classifier determined whether packets are normal or anomalous. It is thought that, in normal conditions, sensors send data at a set frequency; however, in attack scenarios, hackers infect IoT devices and control the sensors, causing them to send data continuously. To determine whether the IoT devices are in a normal state, a timestamp is chosen for the sensor data, and an estimation of the total number of packets for a set period is made. Since ICMP packets are typically sparse on the network, it is simple to detect ICMP attacks. Therefore, receiving many ICMP packets quickly indicates that a device has been compromised. Experimental results show that ICMP flooding, SYN flooding, and UDP flooding were detected with 97.39% accuracy and an F1-score above 97%. The gateway alerts the user as soon as it identifies the flood of sensor data attacks. The SDN controller equipped with the RT166P SDN switch is used to blacklist the compromised device so that IP and MAC addresses from such devices are blocked and network traffic is restricted. Blacklisting compromised devices is performed by the SDN controller via bandwidth control and port mirroring. In port mirroring, a switch sends a copy of every packet received on one port to another port, which is used to store the packets and analyse the data. The bandwidth in the mirror port, which gathers all packets from the eight smart poles for varying numbers of malicious devices that perform DDoS attacks on the Internet, is then examined. The outcome demonstrates that, in an attack-free environment, the mirror port's capacity is approximately 10 Mbps, whereas a single UPD flood records a bandwidth between 80 and 100 Mbps. Therefore, the bandwidth control rule of the SDN switch was set to 800 Mbps to prevent devices from entering overload status when attackers conduct a significant DDoS attack. For a huge DDoS attack, the bandwidth is predicted to be in the range of Gbps or even Tbps.

Mihoub et al. [140] presented an attack detection and mitigation architecture for IoT networks using machine learning. In this study, a multi-class classifier was developed using DT, RF, k-NN, multi-layer perception (MLP), RNN, and LSTM to classify the extracted features from the BoT–IoT dataset. This classifier follows the looking-back idea, where the sub-categories of the attacks are also localized. Evaluation results show that looking-back-enabled RF has the highest accuracy, while the lowest is observed with the k-NN under the same concept. The authors of [141] implemented k-NN, SVM, NB, DT, RF, and LR machine learning algorithms in WEKA tools to analyse their detection performance using the CICDDoS2019 datasets. Evaluation results show that both DT and RF record the highest accuracy, while the NB has the lowest detection accuracy. Nevertheless, the DT has superior performance in terms of processing time. The DT classifier requires 4.53 s to process the data, whereas the RF classifier needs roughly 84.2 s. Similar to the earlier studies in [141], the authors of [142] analysed the potential of SVM, MLP, DT, and RF classifiers for attack detection in a simulated SDN environment using Scapy tool with a list of valid IPs. Results show the superiority of the RF over other classifiers in terms of detection accuracy. The DT, however, has a quicker processing time. The primary drawback of this study is that all traffic was generated artificially and that some traffic characteristics, including IP, protocols, and packet size, were randomly selected. The choice of these features was not discussed. Additionally, these features were insufficient to provide successful detection performance. Aslam et al. [143] proposed a three-layer adaptive machine learning framework for attack detection and mitigation in IoT networks. An adaptive multi-layered feed-forwarding scheme was developed using machine learning classifiers to examine the static features of SDN-enabled IoT network traffic to detect attacks. Within the first layer, classifiers were used to develop a DDoS detection model from the datasets. The findings from this layer were then compiled using an ensemble voting (EV) approach that was applied to the classifiers. The last layer was where live network traffic was measured and compared with the accumulated output of the classifiers to detect anomalies. Results indicate that the framework achieved accuracy ranging from 95% to 98.8%. The accuracy is found to be proportional to the number of test flows. The highest test flows produce the best accuracy. As observed in most of the previous studies [131–143], the choice of feature selection has a considerable effect on the detection accuracy. To improve this, a plethora of multi-class machine learning approaches have been proposed. However, feature selection is still a difficult task. The study presented in [144] applied a hybrid methodology of feature selection to RF, DT, k-NN, and Extreme Gradient Boosting (XGBoost) classifiers. The hybrid feature selection methods are Chi-square, Extra Tree, and ANOVA. The effectiveness of this approach was validated using the CICDDoS2019 dataset. Evaluation results show that XGBoost with ANOVA has superior performance, with an accuracy of 98.35%. This performance was achieved with just 15 features and an 82.5% feature reduction ratio. When all the features were considered, the accuracy of XGBoost dropped to 96.7%.

Attack detection in UAV/IoD/FANET using machine learning: In [13], a machine learning-based approach was suggested for the detection and categorization of GPS spoofing attacks on UAVs. This study implemented three testing scenarios in an outdoor setting, where a sequence of GPS signal characteristics was gathered, and the UAV was subjected to spoofing attacks using an SDR transceiver module. Thereafter, a variety of machine learning classifiers were developed utilizing the datasets obtained from the testing setups of authentic and spoofed flight scenarios. The results presented revealed that the approach permitted detection of GPS spoofing attacks in UAV networks with a detection rate (DR), misdetection rate (MDR), and false alarm rate (FAR) better than 92%, 13%, and 4%, respectively. In [145], a hybrid of LR and RF was utilized to address security concerns with IoT-enabled drones. In this method, cybersecurity vulnerabilities were reduced by incorporating tactics inspired by artificial intelligence within the framework of a drone network. The performance of the developed approach was verified using KDD drone data and GPS characteristic data. Evaluation results indicated that an accuracy of 98.58% was achieved. Additionally, the approach was also evaluated using precision, recall, and F-measure metrics of 97.68%, 98.59%, and 99.01%, respectively. In [146], the authors modelled SYN traffic using Bayesian inference and created an algorithm to identify SYN flood attacks in several wireless ad hoc networks, including MANET, VANET, and FANET. Other than SYN flooding attack detection, the proposed algorithm also detects Hello flooding attacks, RREQ flooding attacks, data flooding attacks, and UDP flooding attacks without a dedicated server node. The DARPA 1999 dataset was used to demonstrate the effectiveness of the proposed scheme. The findings show that the algorithm records a higher true positive rate and precision level. Ouiazzane et al. [147] proposed a multi-agent and DT-based machine learning approach for the detection of DoS attacks in IoD networks. The approach permits the detection of both unknown and known DoS attacks in UAV networks, with low false positive and negative rates and good performance when demonstrated on the CICIDS2017 dataset.

Attack detection in RPL-based IoT using machine learning: IoT devices have limited resources, so they cannot use conventional Internet routing protocols. As a solution to this problem, the RPL defined by the IETF is seen as a viable method to satisfy the routing requirements of Internet of Things networks and reduce resource consumption along the routing path. This protocol can adapt to various situations and has several secure modes. The RPL is a distance-vector routing protocol that builds its topology on a Destination-Oriented Directed Acyclic Graph (DODAG) [148]. In this architecture, traffic is routed to one or more root nodes using a point-to-point (P2P), multi-point-to-point (MP2P), or point-to-multipoint (P2MP) network topology, since all nodes are connected in such a way that there are no round-trip pathways. It has been shown that malicious nodes can carry out their operations while the packets are being routed and forwarded; thus, the security of RPL routing data in the IoT has been a significant challenge. This enables several attack types to take place within the routed data [149]. One of the primary attacks on RPL is flooding attack. In this attack, an excessive amount of traffic is created in a network using the Hello message to disable nodes and links. This attack consumes the resources of the nodes, such as storage, energy, and processing, to cause a denial of service. Other notable forms of attacks in RPL-based IoT networks, such as Sybil attack, wormhole (WH), selective forward (SF) attack, sinkhole (SH) attack, clone ID (CID), version number (VN), and blackhole (BH) attack, have been reported [150–152]. The Sybil attack is extremely similar to a clone ID attack, which allows for the control of a sizable portion of a network without the use of actual physical nodes. Due to variations in topology and complexity, as well as dissimilarity in traffic patterns, conventional security measures, such as those centred around encryption and threshold, are ineffective in detecting attacks in RPL-based IoT networks. Machine learning and deep learning models has been used because of this. Mehbodniya et al. [153] investigated the use of machine learning for Sybil attack detection in RPL-based IoT networks. In this study, NB, RF, and LR classifiers were applied to the data that were generated using the Contiki–Cooja simulator. The performance of the algorithms was evaluated using the accuracy and packet delivery ratio. Experimental results show that NB has the best performance, with 92.14% accuracy, and the best packet delivery ratio, while LR has the worst performance in terms of both metrics. In Osman et al. [154], an attack detection framework was proposed for VN attack detection in RPL-based IoT networks using light gradient boosting machine (LGBM). The dataset for this work was largely created through simulations using Cooja tools for VN attacks. This was followed by a feature extraction module, an LGBM-based classification algorithm, and model parameter optimization. Simulation results proved that the proposed method is effective in identifying VN attacks, with a 99.6% accuracy rate. Additionally, 99.6% precision and a 99.6% F1-score were recorded. The proposed approach occasionally generates false alarms. Additionally, only one attack can be localized. Moreover, the method requires lengthy processing times and high computational overhead.

In [155], ANN was employed in RPL-based IoT networks to identify HF, DR, and VN attacks. Ten-fold cross-validation techniques were utilized in this study to prevent over-fitting. The performance of the proposed model was compared, considering both the holdout approach and the ten-fold cross-validation technique. The results of the simulations demonstrate that the proposed model is 100% accurate in localizing the attacks in each scenario. To generate datasets, the authors, however, employed a small network. Similarly, Verma and Ranga [156] examined the efficacy of machine learning classifiers for the detection of SH, BH, Sybil, CID, SF, and Hello flooding (HF) attacks in RPL-based IoT networks using Boosted Trees, subspace discriminant, RUSBoosted Tree, and bagged trees. Hold-out and cross-validation methods were used to investigate this performance. According to simulation results, the subspace discriminant model performs the worst, with an accuracy of 77.8% and an area under the curve (AUC) of 0.87 for a 40% hold-out validation, while the ensemble of Boosted Trees performs the best, with an accuracy of 94.5% and an AUC of 0.98. The Boosted Trees and RUSBoosted Trees have the best accuracy and AUC, respectively, in the cross-validation scenario. In Sharma et al. [157], the potential of three machine learning classifiers was examined for attack detection in RPL networks. The Cooja network simulator was used in this study to create a multi-class dataset that included a standard traffic pattern and four RPL attacks, including HF, flooding, VN, and DR attacks. These datasets were evaluated independently using the classifiers RF, NB, and J48. Experimental results indicated that the RF classifier has the best performance. Superior to NB and J48, RF achieved precision, recall, and accuracy values of 99.4%, 99.3%, and 99.33%, with the J48 classifier performing the worst.

Attack detection in NDN using machine learning: Named Data Networking (NDN) is an emerging next-generation network architecture that is anticipated to replace the current IP-based internet infrastructure. It employs the content-centric networking paradigm, where content is retrieved using names rather than the network addresses of the servers hosting it [158]. In this architecture, a source can request content by employing name prefixes rather than the present IP prefixes to route an interest request. Interest packets are routed to the location of the original source of the content. Any router and intermediary node, along the path, search their cache for identical copies of the requested content. Any piece of interest request that has a cached copy is returned to the requester along the path it originated from. All middle nodes keep a copy of the content in their caches on the way back to prepare for potential same-interest requests from incoming requests [158]. Each NDN router maintains three fundamental data structures: the Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS) [159]. NDN was initially projected to address the basic shortcomings of the existing internet-based network, but attackers are now using the two unique features in NDN routers, CS and PIT, to launch new variants of DDoS attacks against it. Thus, they are susceptible to new types of attacks. The two most prominent categories of DoS/DDoS attacks in NDN infrastructure are the interest flooding attack (IFA) and content/cache poisoning attack. Other forms of attack in NDN, such as cache privacy attacks, cache pollution attacks, and false locality attacks (FLA), have been reported [159,160]. The goal of content poisoning is to prevent users from accessing legitimate content by forcing routers to forward and send spoof data packets [158]. The IFA is one of the most severe attacks in NDN. This attack is an extended feature of DDoS in NDN, whereby the attackers flood the network with many non-existing interest packet requests. These requests are stored in the PIT of the NDN routers in between. Due to the persistence of these entries in PITs of NDN routers, valid requests are denied space in the PITs [158,159]. Despite the NDN's potential, it still lacks a good defence scheme against DoS and DDoS attacks. Deep learning and machine learning techniques have recently been developed for NDN attack detection. Kumar et al. [160] proposed a machine learning framework for IFA attack detection in NDN. In this study, IFA was modelled and simulated to gather attack features. The most prominent features were selected based on information gain-based ranking; thereafter, DT, J48, and MLP with backpropagation machine learning classifiers were used for IFA detection. According to experimental data, MLP with BP is more appropriate in terms of identifying and mitigating IFA, while the J48 classifier works better for large network topologies.

In [161], the authors used an SVM classifier to characterize the entropy of interest names, the satisfaction ratio, and the PIT usage of interfaces that are continuously acquired from a router. The Jensen–Shannon divergence was used to extract malicious prefixes, and an IFA activity was notified when anomalies were found. When the SVM was applied without the Jensen–Shannon divergence, a high misjudgment rate was seen. However, with the inclusion of this entropy scheme, experimental results revealed that the approach achieved high accuracy. The fact that this study can only identify one kind of attack in NDN poses a significant constraint. Other attack types cannot be identified and require further development. As with the prior solution [160], the detection process may consume a lot of resources. In [162], the authors set up sample sets with various detection granularities to improve detection accuracy using an RF classifier. Experimental results show that the scheme could detect IFA attacks with a high detection rate. It was possible to attain detection probabilities of 97.5% and false negative probabilities of 1.2%. Additionally, some error cases with a 3% error rate were noted.

Table 8 provides a summary of the research papers and thus far in the application of machine learning models for attack detection. These studies were compared based on the method employed, the dataset, and the application domain. A summary of the evaluation findings from each study is also provided. As shown in Table 8, a number of studies have discussed the use of machine learning for detecting network anomalies, with varying levels of accuracy and false alarms.

Table 8. Summary of studies on the application of machine learning models for DDoS attack detection.

Study	Method Used	Description	Dataset	Application Domain	Remarks
[13]	RF, k-NN, MLP, LR, DT, SVM, NB	• Gathers sequence of GPS signal characteristics from UAV setup. • UAV is subjected to spoofing attacks using an SDR transceiver module.	Synthesized	FANET	• Detection of GPS spoofing attacks in UAV networks was achieved. • Gives a detection rate, misdetection rate, and false alarm rate of 92%, 13%, and 4%.
[131]	SVM	• Collect network packets periodically. • Extract 24 features from each packet. • Classify the features using SVM.	NSL–KDD	IoT–SDN	• Achieves a detection accuracy of 99.4%. • High computational overhead.
[139]	DT	• Capture sensor data packets via IoT gateway. • Data are used to train a DT classifier.	Generated	Multi-layer IoT	• ICMP flood, SYN flood, and UDP flood are detected with 97.39% accuracy. • F1-score above 97% is achieved.
[144]	RF, DT, k-NN and XGBoost	• Apply feature selection approach on four classifiers to assess the detection accuracy.	CICDDoS2019	IoT	• XGBoost with ANOVA has the superior performance, with an accuracy of 98.35%.
[145]	LR and RF	• A hybrid of LR and RF was utilized to address security concerns with IoT-enabled drones.	KDD drone data	IoD	• Records an accuracy of 98.58%. • Precision, recall, and F-measure values of 97.68%, 98.59%, and 99.01% are recorded.
[146]	Bayesian inference	• Model SYN traffic using Bayesian inference. • Develop an algorithm to identify SYN flood attacks.	DARPA'99	MANET, VANET, FANET	• Detects SYN flooding, Hello flooding, RREQ, and UDP flooding. • Has high true positive and precision level.
[147]	Agent-base and DT	• A multi-agent and DT-based machine learning approach was developed for attack detection.	CICIDS2017	IoD	• Detects unknown and known DoS attacks. • Has low false positive and negative rates.
[153]	NB, RF, LR	• Investigate the effectiveness of machine learning classifiers for Sybil attack detection.	Simulated	RPL-based IoT	• NB has the best performance with 92.14% accuracy. • LR has the worst performance.
[154]	LGBM	• Propose a framework for VN attack detection using LGBM.	Simulated	RPL-based IoT	• Achieves 99.6% accuracy. • Precision of 99% and F1-score of 99.6%. • Only one attack can be detected. • Large computational overhead.

Table 8. *Cont.*

Study	Method Used	Description	Dataset	Application Domain	Remarks
[155]	ANN	• ANN for detecting HF, DR, and VN attacks in RPL-based IoT networks.	Simulated	RPL-based IoT	• Attacks are localized with 100% accuracy. • Small network for dataset generation.
[157]	RF, NB, J48	• Evaluate the performance of three classifiers for attack detection.	Simulated	RPL-based IoT	• RF has the best performance in terms of accuracy, precision, and recall. • J48 has the worst performance.
[160]	DT, J48, MLP + BP	• Simulate IFA to gather attack features. • Use DT, J48, and MLP + BP on the most prominent attack features.	Simulated	NDN	• J48 classifier performs better for large topology. • MLP + BP is more suitable in context to detecting and mitigating IFA. • The detection process may consume a lot of resources.
[162]	RF	• Set up sample sets with different detection granularity to improve detection accuracy	Simulated	NDN	• Detects IFA attack. • Records detection probability of 97.5%. • Error rate of 3%.

3.3.7. Deep Learning-Based Detection Methods

The deep learning structure makes use of the supremacy of both supervised and unsupervised learning with its feature extraction and classification module [163,164]. Due to this advantage, research studies are being tailored to deep learning for DDoS attack detection in internet-enabled networks.

Attack detection in IoT and SDN using deep learning: Hassan et al. [165] proposed a deep convolutional neural network (DCNN) model for DDoS attack detection in an optical switching network. The performance of this model was compared to SVM, k-NN, and NB. The results demonstrated that DCNN outperformed SVM, k-NN, and NB, achieving 99% detection accuracy, as compared to 88%, 93%, and 79% detection accuracy for SVM, k-NN, and NB, respectively. Additionally, a misclassification rate of 1% was observed. In [166], the use of ANN with a signature-based method was investigated. The results presented showed that the combined approach has an accuracy of 99.98% with false positive rates of zero. Zhu et al. [167] utilized CNN and feed-forward neural networks (FNN) models for network traffic analysis and anomaly detection. When tested on the NSL–KDD dataset, the findings revealed that an accuracy of 77.84% was recorded. When compared to the other classifiers tested, including NB, RF, J48, RT, and SVM, the accuracy value was deemed to be higher. However, a detection accuracy of 77.84% is still rather low compared to other comparable research results [165,166]. The study by the authors in [168] used LSTM for DDoS attack detection in fog computing environments. Network packets recorded at a specific time interval wre employed to train the LSTM. The number of hidden layers in the LSTM was investigated for detection accuracy. The LSTM with three hidden layers and 128 units was found to be appropriate, with a detection accuracy of 98.88% when demonstrated on the ISCX 2012 dataset. The DeepDefense system proposed by Yuan et al. [169] leverages CNN, RNN, LSTM, and a gated recurrent unit neural network (GRUNN) to localize attacks in IoT networks. The approach was demonstrated on the ICX2012 dataset. A substantial decrease in error rate was achieved when compared to the conventional machine learning approach. With a 98% detection accuracy, the deep learning model lowers the error rate by 39.69%. In research by Shurman et al. [170], the utilization of two methodologies for attack detection in an IoT network was examined independently. The first method makes use of a hybrid intrusion detection system, while the second approach uses an LSTM deep learning model. The applicability of these approaches was demonstrated on the CICDDoS2019 dataset. The two methods had a detection accuracy of 91.9% for both DDoS and DoS attacks. In this study, a few cases of false alarms were recorded.

Ge et al. [171] proposed a tailored deep learning approach for detecting attacks in an IoT environment. An embedding layer and an FNN approach were used in this study to perform multiclass attack prediction. Additionally, an FNN model was also developed to perform binary classification. Evaluation results reveal the success of the method. Both classifiers performed better. Particularly, the binary classifier showed detection accuracy close to 99.99%, while the multi-class classifier recorded about 99.79% accuracy. In this study, only a few attack classes were reportedly detected. Thus, detecting other forms of attack is not guaranteed. The study presented in Elsayed et al. [172] discusses the use of an RNN with an autoencoder (AE) to improve detection accuracy during a DDoS attack in SDN. The success of this scheme was evaluated in comparison to NB, RF, DT, SVM, and linear regression classifiers. The scheme has a significant enhancement in terms of accuracy when demonstrated on the CICDDoS2019 dataset compared to existing approaches. The approach records an accuracy of 99%. The computational overhead was slightly reduced; the study, however, excluded reporting performance parameters such as model training time or samples classified. Roopak et al. [173] evaluated the effectiveness of CNN, LSTM, MLP, and a hybrid of CNN and LSTM (that is, CNN + LSTM) for attack detection in IoT networks. The effectiveness of these models was demonstrated using the CICIDS2017 dataset. The CNN + LSTM has the highest accuracy among these models, whereas the MLP has the lowest accuracy. LSTM performs second-best, with an accuracy of 96.24%, while CNN + LSTM achieves a detection accuracy of 97.16%. However, in terms of precision, the LSTM is observed to have the lowest precision results, while the MLP has the second-best performance. The study offers no justification for this obvious disparity. An AE-based unsupervised deep learning framework was proposed by Abeshu et al. [174] for the fog computing layer. The fog node is where training and parameter updates are carried out. The stacked AE model was pre-trained with unlabelled data, and was subsequently used to classify test data. The effectiveness of the method was demonstrated on the NSL–KDD dataset, considering 41 features. An excellent accuracy of 99.2% and a detection rate of 99.27% were recorded. In this study, only a very few cases of false alarms were recorded. The performance of the suggested approach needs to be proven on more recent data because the dataset utilized are outdated. The study presented by the authors in [175] shows that a bidirectional long short-term memory-based RNN (BLSTM-RNN) could be effectively employed for attack detection. The performance of this approach was also compared to a unidirectional LSTM-RNN for the detection of botnet attacks. In this study, four attack vectors were considered in the generated dataset used for validation. These attack vectors include Mirai, UDP, DNS, and ACK. The method performed well in detecting Mirai, UDP, and DNS attacks with 99.0%, 98.0%, and 98.01% accuracies. When the ACK attack was considered, though, its performance deteriorated. This strategy has the significant drawback of adding computing overhead to each epoch, which increases the processing time.

***Attack detection in UAVs/IoD/FANET using deep learning*:** In [14], the authors proposed a sea turtle foraging algorithm with a hybrid deep learning-based intrusion detection scheme (STFA-HDLID) for attack detection in an IoD environment. In this approach, the feature selection process was achieved with the STFA. Additionally, classification was performed using a Deep Belief Network (DBN) and the Sparrow Search Optimization (SSO) algorithm. The performance of the approach was demonstrated using the TON_IoT and UNSW-NB15 datasets. The results presented showed that an accuracy of 99.51 was recorded for the TON_IoT dataset, while 98.85% was achieved when the UNSW-NB15 dataset was considered. The authors of [176] investigated a framework for attack detection in FANET utilizing recurrent neural networks. The framework has both the data collection and the data stream processing modules. The latter gathers communication data from the drones, including data relevant to intrusion detection, which is subsequently put into two RNN modules for data processing. The efficacy of the proposed approach was verified using the KDD Cup'99, NSL–KDD, UNSW-NB15, Kyoto, CICIDS2017, and TON_IoT datasets. The results showed that the framework has excellent performance. In [177], a deep convolutional neural network (DCNN) was utilized for attack detection in UAV networks. This method made use of encrypted wireless traffic records that were gathered from three different types of frequently used UAVs: DJI Spark UAVs, Parrot Bebop UAVs, and DB Power UAVs. The performance of the proposed approach was demonstrated using the UAV-IDS-2020 dataset, which has numerous attacks against UAV networks. Experimental results show that a detection accuracy of 99.50% was achieved with a 2.77 ms prediction time. The authors of [178] concentrated on crystal structure optimization for attack detection in the IoD environment using a deep autoencoder-based model. In this work, the feature subsets were selected using a modified deer hunting optimization-based feature selection strategy, and the attacks were classified using an AE method. The model was simulated, and the results obtained showed that an accuracy of 99.12% was achieved. However, the proposed model needs to be tested on a large-scale, real-time dataset. Zhang et al. [179] developed an open-CNN model for the detection of unknown attacks in drone networks. Extensive experimental demonstrations showed that the developed model could detect DDoS, DoS hulk, botnet, and web attacks when tested on the CICIDS 2017 dataset. In addition, the authors compared the performance of the developed model with CNN and CNN–LSTM. The results presented revealed that the accuracy was improved by 9–30% when compared to the CNN and CNN–LSTM models.

***Attack detection in RPL-based IoT using deep learning*:** A framework that employs stacked AE-based DNN for CID attack detection in RPL-based IoT networks was proposed by Molina et al. [180]. To create a dataset for this study, CID attacks were implemented using the Cooja network simulator, considering three different network structures with a number of benign and malicious nodes. The SAE + DNN model was used to process and categorize these data. Experimental results showed that the framework detects CID attacks with an average accuracy of 99.65%. The proposed framework, however, is only capable of detecting CID attacks and cannot be utilized to identify other types of attacks in RPL-based IoT networks. Additionally, more computational overhead is observed, which is not good for resource-constrained IoT devices. In [181], an ANN-based attack scheme was proposed, using MLP for attack detection in an RPL-based IoT network. The proposed approach has three stages: simulation, pre-processing, and classification. In the simulation stage, packet data are generated from the Contiki network simulator; the features of these data are extracted during pre-processing, while the classification stage involves the application of the MLP to the extracted features to identify attacks. Simulation results show that the approach could identify a VN RPL attack. Additionally, the method records a root mean square error (RMSE) of 0.0003 and a mean absolute error (MAE) of 0.0002. This study does not cover other well-known performance metrics such as accuracy, precision, recall, and F-measure. Additionally, the method has a higher computational burden, which could restrict its application to constraints devices.

In Cakir et al. [182], a deep learning approach to detect hello flooding attacks in RPL was presented, using a gated recurrent unit (GRU) network with RRN. Similar to the studies presented in [181], the approach also consisted of three stages: network simulation, preprocessing, and detection. The network simulation made use of the Contiki–Cooja simulator to generate datasets, which were processed and fed to the input of the GRU + RNN to differentiate between legitimate and malicious nodes. The performance of the proposed model was verified using five and four feature sets. According to the results, an accuracy of 99.96% was attained for the five-feature set and 99.90% for the four-feature set. Additionally, a mean square error of 0.05 was achieved. Similar to the research presented in [181,182], the authors in [183] used an MLP classifier to distinguish between normal and malicious behaviour using a dataset generated from the Cooja simulator. Hello flooding, VN, and decreased rank (DR) attacks were all included in this dataset. Results from the simulation indicated that a 99.5% accuracy rate was attained. Additionally, according to the F1-score results, the approach has a detection rate of 94.7%, 99%, and 95% for DR, HF, and VN attacks, respectively. Analysis of other critical parameters, such as end-to-end delay and processing times, may also need to be investigated. Additionally, the used traffic data are static and do not accurately reflect the dynamic nature of internet traffic.

Attack detection in NDN using deep learning: Zeng et al. [184] proposed a scheme based on CNN for detecting FLA in NDN. In this study, the regularity of previous requests was harnessed, and the inherent features of the cached contents, such as the request ratio, the standard deviation of repeated interests, the variance of the request interval, and the change in cache hit ratio, were used as input data to the CNN. CNN was able to classify the attacks and report whether an attack had been executed. The scheme was simulated using different network topologies, and the results revealed that the scheme is effective in detecting FLA with a detection ratio of 26.3% and a cache hit ratio of 12.2%. It also records a lower hop count. Unlike previous solutions, the authors in [154] developed a hybrid multi-objective strategy employing optimization and a deep learning model for DoS attack detection in NDN. This was accomplished by merging the multi-objective evolutionary optimization technique with particle swamp optimization, while the prediction accuracy was improved using the radial basis function (RBF) neural network. When malicious traffic is recognized, the router notifies the source interfaces. The performance of the hybrid scheme was demonstrated in a simulated environment consisting of different network topologies. Evaluation results showed that the scheme can respond to and mitigate DoS attacks with good accuracy. An accuracy of more than 90% was recorded in terms of the average interest satisfaction ratio for legitimate users, the PIT usage, and the number of received contents. Moreover, a very low false positive rate was achieved. A feature analysis of detection parameters was not discussed in this study. Similar to the study presented in [154], Kumar et al. [185] applied deep learning models for IFA detection in NDN using linear and DFN network topologies in the ndnSIM and CCNx code bases. These simulated network topologies were used to generate the dataset, which was then used on MLP with back propagation (MLP + BP) and RBF with computed k-means clustering. In addition, the RBF was combined with other optimization algorithms such as PSO (RBF + PSO), RBF + JAYA, and teaching learning-based optimization (RBF + TLBO). To localize the IFA attacks in the dataset, SVM and k-NN classifiers were also created individually, and the effectiveness of these techniques was evaluated. According to experimental findings, MLP + BP, RBF + PSO, RBF + JAYA, and RBF + TLBO have more accurate detection than k-NN and SVM. The MLP + BP offers the highest precision (97.5%) and accuracy (97.3%) while using CCNx code. Additionally, a few instances of false alarms were noted.

In Table 9, a summary of the existing research studies on the use of a deep learning model for attack detection is presented. These studies are compared based on the deep learning model used, the dataset, and the application domain. A summary of the evaluation findings from each study is also provided.

Table 9. Summary of some studies on the application of deep learning models for DDoS attack detection.

Study	Method Used	Description	Dataset	Application Domain	Results
[14]	DBN	• Propose a sea turtle foraging algorithm with a hybrid deep learning-based intrusion detection.	TON_IoT and UNSW-NB15	IoD network	• An accuracy of 99.51 is recorded with the TON_IoT dataset. • Accuracy of 98.85% with UNSW-NB15.

Table 9. *Cont.*

Study	Method Used	Description	Dataset	Application Domain	Results
[158]	RBF with PSO	• Use a hybrid multi-objective scheme with optimization and deep learning.	Simulated	NDN	• Records accuracy more than 90%. • A very low false positive rate.
[170]	LSTM	• Investigate the use of hybrid-based intrusion detection system and LSTM.	CICDDoS2019	IoT environment	• The two approaches achieve an accuracy of 91.9%.
[171]	FNN	• Combine FNN model with an embedding layer for multiclass attack prediction.	Generated	IoT	• Only a few classes of attack are reportedly detected; thus, detection of other form of attack is not guaranteed.
[172]	RNN, AE	• Combine RNN with AE to enhance accuracy.	CICDDoS2019	SDN	• The approach records an accuracy of 99%. • Reduces computational overhead.
[174]	AE	• Propose an AE-based unsupervised deep learning framework.	NSL-KDD	IoT	• Accuracy of 99.2% is achieved. • Records very few cases of false alarms.
[177]	DCNN	• Develop a deep leaning approach for attack detection in UAV networks.	UAV-IDS-2020 dataset	UAV network	• Has a detection accuracy of 99.50%. • The approach has 2.77 ms prediction time.
[179]	CNN, CNN-LSTM	• Develop an open-CNN model for the detection of unknown attacks.	CICIDS2017	IoD network	• Detects DDoS, DoS hulk, botnet, and web attacks. • Detection accuracy improves by 9%–30% when compared to the CNN and CNN–LSTM models.
[180]	SAE + DNN	• Propose a framework for CID attack detection.	Simulated	RPL-based IoT	• Records average accuracy of 99.65%. • Limited to CID attack detection only.
[181]	MLP	• ANN-based scheme for attack detection in RPL-based IoT network.	Simulated	RPL-based IoT	• Detects VN attack. • Large computational overhead.
[182]	GRU + RNN	• Use Contiki–Cooja simulator to generate dataset used as input for GRU + RNN.	Simulated	RPL-based IoT	• Accuracy of 99.96% for five-feature set • Accuracy of 99.90% for four-feature set.
[183]	MLP	• Use MLP classifier to identify normal and malicious behaviour from a dataset generated from Cooja simulator	Simulated	RPL-based IoT	• High accuracy for HF attack. • F1-scores of 94.7%, 99%, and 95% for DR, HF, and VN attacks. • Uses static traffic data.

Table 9. *Cont.*

Study	Method Used	Description	Dataset	Application Domain	Results
[184]	CNN	- Propose a scheme based on CNN for detecting FLA. The scheme was simulated using different network topologies.	Simulated	NDN	- A detection ratio of 26.3% and cache hit ratio of 12.2%. It also records a lower hop count.

Table 10 displays a comparison of anomaly-based attack detection methods. This comparison is based on the features of each method, their advantages, and limitations.

Table 10. Comparison of the anomaly-based detection methods.

Methods	Features	Advantages	Limitations
Entropy -based	Compares estimated entropy of traffic features against a pre-defined threshold.	- Quick. - Good accuracy.	- Threshold issue. - Poor response time for attacks with many packets.
Chaos-based	Uses an estimate of Lyapunov exponent in network traffic orbit to determine attack.	- Has a relatively better error rate.	- Accuracy depends on the choice of window size. - High computational overhead.
Queuing theory	Uses queue management algorithm.	- Fast response time.	- Very poor detection accuracy for large-scale DDoS attacks. - Setting the length of the queue. - High false negative rates.
Statistical approach	Statistical tests are performed to verify if the observed pattern is different from the expected pattern based on historical data.	- Simple. - Good detection accuracy.	- Accuracy depends on the mathematical model. - Setting optimal threshold. - Misclassification
Heuristic-based	Uses algorithmic logic to analyse statistical features of network traffic.	- Minimizes false positives and false negatives. - Good detection accuracy.	- Accuracy depends on detection thresholds. - High computational cost.
Machine learning	Uses algorithms to identify malicious traffic from a pool of network traffics just by learning the characteristics of the network traffic.	- Identifies traffic patterns quickly. - Superb detection accuracy.	- Feature engineering problem. - Long training time. - Larger dataset is required for better accuracy.
Deep learning	Utilizes the advantages of supervised and unsupervised learning with its feature extraction and classification module.	- Ability to learn high-dimensional features. - Flexible adaptation to novel problems. - Superior layer feature learning ability. - Ability to directly process raw data.	- Generalization issue. - Sometimes leads to overfitting. - Computational overhead. - Larger dataset is required for better accuracy.

4. Benchmark Databases and Performance Evaluation Metrics

4.1. Dataset Used

Datasets are used to train and verify the applicability of most DDoS attack detection methods. The availability of datasets is one of the greatest hurdles for these methods. Access is restricted because of worries about privacy and unauthorized use, as well as sporadically occurring legal conflicts amongst the parties involved. The network traffic data contain sensitive information that might not be made available to the public outside of a research setting. Most of the time, researchers create their own datasets for DDoS attack detection. Regrettably, a significant proportion of the generated datasets might not be sufficiently complete and, in some situations, might not fit the application domain. In this section, we describe a few of the well-known datasets that have been frequently used to identify DDoS attacks.

4.1.1. DARPA'98/99

This is one of the most widely used for intrusion detection studies. It was generated in a simulated network environment at the MIT Lincoln Lab. It includes packet-based records of network traffic for seven weeks and five days. The dataset is available for download at [186] and is open to the public. Despite its widespread use, investigations in [187] discovered numerous redundancies that prevent it from being used for more realistic and real-world attack detection.

4.1.2. KDD Cup'99

This dataset was created by [188] and was derived from the DARPA'98/99 dataset. It is also famous for its use in this field and has a wide range of attacks. The dataset, which is in packet or flow format, has been utilized to verify the signature of a known attack detection approach [189]. The dataset, which covers basic TCP connection information, has more than 20 different forms of attack. The dataset contains 5 million data points and is freely available for download at [188].

4.1.3. NSL–KDD

Due to the enormous number of redundancies that the KDD Cup'99 dataset revealed, this dataset was developed to enhance it. Replicas from the KDD Cup'99 were removed to create a more complex sub-set. The final dataset contains over 150,000 data points. This dataset comprises 7853 DoS attack testing results and 53,385 DoS attack training results [190]. The dataset is available for free download at [191] in flow-based format.

4.1.4. SSENeT-11

Vasudevan et al. [192] generated the SSENet-11 dataset using the Tstat tool. The dataset was captured in a simulated environment for 4 h. It has numerous DoS or port scan attack types. Each data point in this collection is identified by 24 attributes that are preserved as packet-based traffic. The dataset is not freely accessible.

4.1.5. SSENet-14

This was developed by [193] by extracting features from the packet-based files of SSENet-11 [192]. The dataset was developed in a realistic network environment. Attack tools were used to generate the attacks while performing routine tasks. Each data point is distinguished by 28 attributes recorded as packet-based traffic, similar to the SSENet-2011 and KDD Cup'99 datasets. SSENet-14 has 200,000 data points that have been annotated. The dataset is not accessible to the public, the same as for SSENet-11.

4.1.6. Kent2016

This dataset was collected over a period of 58 days at the Los Alamos National Laboratory [194] using de-identified network event data. It contains 130 million flow-based traffic records and various host-based log files that have been anonymized due to privacy concerns. The dataset includes 62,974 processes, 12,425 users, 17,684 computers, and 1,648,275,307 events, totalling 12 GB in size. The dataset is publicly available and can be downloaded at [194].

4.1.7. ISCX2012

Shiravi et al. [195] generated this dataset in 2012. This was accomplished by recording one week's worth of network traffic in a simulated environment. The attacks include DDoS, SSH brute force, infiltration, and DoS. The dataset is freely accessible online in packet and bi-directional flow-based formats.

4.1.8. CIC DoS

This dataset was generated by the Canadian Institute for Cybersecurity (CIC). The dataset has a total size of 4.3 GB and consists of several HTTP-based attacks. The attack traffic was produced using Ddossim, Goldeneye, and Hulk. Normal traffic was created from non-attack traffic in the ISCX2012 dataset. The dataset is available in packet-based format at [196] for public access.

4.1.9. DDoS2016

DDoS2016 was produced in 2016 by [197] via the NS2 simulator. Along with typical network traffic, the dataset also has UDP flood, Smurf, HTTP flood, and SIDDOS attacks. The dataset has 2.1 million packets and is freely available to the public.

4.1.10. NDSec-1

Beer et al. [198] produced this dataset in 2016. A variety of cyberattacks are included, such as a botnet, brute force attacks (FTP, HTTP, and SSH), HTTP flood, SYN flood, UDP flood, exploits, port scanning, spoofing, and SQL injection. There are a total of 5838 records in this dataset. From these records, 3558 records (or 60.1%) are categorized as normal traffic and 2380 records (or 39.9%) as attack traffic.

4.1.11. CICIDS2017

This dataset closely resembles actual real-world data and includes several legitimate and recent attacks. It represents the outcome of network traffic studied using a CIC flow meter. Eight separate files made up the dataset, which covered five days of both normal and attack traffic created between 3 July and 7 July 2017. CICIDS2017 meets all critical criteria for an intrusion detection dataset [199]. It has prevalent attacks, including DoS, DDoS, brute force, XSS, SQL Injection, web infiltration, port scanning, and botnets. It contains labelled network flows in flow-based formats as well as entire packet payloads in packet-based formats for machine and deep learning [199]. In this dataset, there are a total of 225,745 records. There are 97,718 records classified as "normal traffic", which amounts to 43.3%, and 128,027 (56.7%) classified as "attack traffic". The dataset is publicly available for researchers and can be downloaded from [200] in both pcap and csv formats.

4.1.12. CICIDS2018

This dataset was produced in cooperation with the CIC and the Communications Security Establishment (CSE) [201] as an expansion of the CICIDS2017 dataset. The objective was to develop a new dataset that was scalable and more accurate. The authors used the Amazon Web Services (AWS) platform instead of the obsolete network architecture used in the CICDIS2017 dataset to build the normal and attack classes using the same notion of network profiles. The network traffic in the CICIDS2018 dataset was compiled over a 10-day period. It has attacks such as brute force, Heartbleed, botnets, DoS, DDoS, web attacks, and infiltration of the network. There are a total of 1,046,845 records in this dataset. There are 360,833 records classified as "normal traffic", which amounts to 34.5%, and 686,012 (65.5%) classified as "attack traffic". This is freely accessible to researchers in flow and packet formats.

4.1.13. CICDDoS2019

This is one of the most recent datasets produced for DDoS attack detection. It contains a legitimate, most recent type of DDoS attack. CICDDoS2019 includes more distinct DDoS attacks with high traffic [202]. It has modern reflective DDoS attacks such as Portmap, NetBIOS, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. In this dataset, there are a total of 294,627 records. There are 121,980 records classified as normal traffic, which amounts to 41.4%, and 172,647 (58.6%) records classified as attack traffic. The dataset is publicly available and can be downloaded from [203] both in pcap file format and flow-based format.

4.1.14. IoTID20

This is a recent dataset employed for attack detection in IoT networks. Ullah and Mahmoud [204] created the IoT intrusion detection dataset 2020 (IoTID20). There are eight IoT cyberattacks in this dataset. There are also 79 features in the dataset that distinguish between legitimate and malicious traffic. The dataset has been employed to verify the applicability of machine learning for the detection of zero-day attacks [205].

4.1.15. UAV-IDS-2020

This is a recently compiled dataset utilized in UAV communication networks to detect or predict attacks. It contains updated and non-redundant encrypted Wi-Fi traffic logs, with binary output labels listed for each record as 0 for UAV normal and 1 for anomalous [177]. The dataset has several interpretations of the data logs as having genuine and anomalous networked UAV communication traffic. The original dataset is available for the bidirectional-flow mode, which includes the uplink flow, downlink flow, and total traffic flow, as well as the unidirectional-flow mode for UAV communication. In total, there are 55 attributes and 17,256 instances in the dataset. UAV-IDS-2020 is publicly available in flow-based format.

Table 11 provides an overview and comparison of the well-known datasets utilized in studies on attack detection. In this table, the details of the various attacks within the dataset are provided. Additionally, details regarding the type of attack, file format, volume and duration of the traffic, and test networks are shown. It has been noted that most of the datasets are freely accessible in both packet and flow formats, provide information about legitimate and malicious traffic, and are, thus, helpful for attack detection. It is also observed that CICDDoS2019 and IoTID20 are the most recent datasets for DDoS attack detection. Of note, the most recent types of attacks are included in the CICDDoS2019 dataset.

Table 11. Overview of some of the frequently used datasets.

Dataset	Year	Publicly Available?	Traffic Category	Format	Traffic Volume	Span	Traffic Present	Attack Type
DARPA'98/'99	1998/1999	Yes	Simulated	.pcap, logs	n/a	7.5 weeks	Normal and attack traffic	DoS, privilege escalation, probing
KDD Cup'99	1999	Yes	Simulated	-	5 M points	-	Normal and attack traffic	TCP, DoS, privilege escalation, probing
NSL–KDD	1999	Yes	Simulated	-	150 k points	-	Normal and attack traffic	DoS, probing
SSENet-11	2011	No	Simulated	-	n/a	4 h	Normal and attack traffic	DoS, port scan
ISCX2012	2012	Yes	Real	.pcap, .csv	2 M flows	7 days	Normal and attack traffic	Infiltration, DDoS, SSH brute force, HTTP DoS
CIC DoS	2012	Yes	Simulated	.pcap	4.6 GB packets	24 h	Normal and attack traffic	Slowloris, slowbody, slowread, Hulk, app. layer DoS
SSENet-14	2014	No	Simulated	-	200 K points	4 h	Normal and attack traffic	Botnet, flooding, port scan

Table 11. *Cont.*

Dataset	Year	Publicly Available?	Traffic Category	Format	Traffic Volume	Span	Traffic Present	Attack Type
NDSeC-1	2016	No	Simulated	.pcap, logs	3.5 M packets	-	Attack traffic only	Botnet, HTTP flood, SYN flood, UDP flood, SSL proxy, SQL injection, spoofing, exploits
DDoS 2016	2016	Yes	Synthetic	.pcap	2.1 M packets	-	Normal and attack traffic	HTTP flood, Smurf ICMP flood, UDP flood
CICIDS2017	2017	Yes	Simulated	.pcap, .csv	3.1 M flows	5 days	Normal and attack traffic	Botnet, LOIC, SQL injection, slowloris, SSH brute force
CICIDS2018	2018	Yes	Simulated	.pcap, .csv	6.89 GB packets	10 days	Normal and attack traffic	Brute force, botnet, Heartbleed, DoS, DDoS, web attacks, infiltration
CICDDoS2019	2019	Yes	Simulated	.pcap, .csv	13.01 GB packets	2 days	Normal and attack traffic	PortMap, NetBIOS, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, SNMP, SSDP, TFTP, Web-DDoS
IoTID20	2020	No	Simulated	.csv	-	-	Normal and attack traffic	SYN, UDP, HTTP, ACK floods, Host brute force, port scan
UAV-IDS-2020	2020	Yes	Real	.csv	-	-	Normal and attack traffic	GPS spoofing, jamming, DoS

4.2. Evaluation Metrics

Researchers have evaluated the effectiveness of attack detection methods using a variety of criteria. In this section, we present some of the measures that are typically employed to assess the effectiveness of these methods.

4.2.1. Detection Accuracy

Any method used for attack detection should be able to accurately detect attacks. This metric determines the percentage of accurate predictions across all the cases considered. The detection accuracy (D_A) is expressed using (26):

$$D_A = \frac{T_P + T_N}{T_P + F_P + T_N + F_N} \tag{26}$$

In (26), T_N stands for true negative and signifies the number of instances of normal traffic that the detection method correctly classifies as belonging to the normal class; F_N denotes false negative, indicating the number of instances of attack traffic that are identified as normal traffic; true positives (T_P) signifies the number of attack instances accurately categorized; and false positives (F_P) signifies the number of instances of normal traffic that are wrongly classified as attack instances.

4.2.2. Error Rate

The error rate (E_R) is also known as misclassification or misidentification error. It calculates the proportion of inaccurate predictions across all instances. It is written as shown in (27):

$$E_R = \frac{F_N + F_P}{F_N + F_P + T_N + T_P} \tag{27}$$

4.2.3. Specificity

It is a measure of the normal traffic that is accurately predicted. The specificity (S_p) is expressed as below:

$$S_p = \frac{T_N}{T_N + F_P} \tag{28}$$

4.2.4. Precision

It is an estimate of the proportion of positive patterns in a positive class that are successfully predicted out of all the anticipated patterns. The precision (P) is expressed as shown:

$$P = \frac{T_P}{T_P + F_P} \tag{29}$$

4.2.5. Sensitivity/Recall(s)

It measures the proportion of positive patterns that are classified properly. It is expressed as follows:

$$s = \frac{T_P}{T_P + F_N} \tag{30}$$

4.2.6. F-Measure/F1-Score

This metric denotes the harmonic mean between s and P results:

$$F - measure = \frac{2 \times s \times P}{s + P} \tag{31}$$

5. Key Findings and Discussions

In this section, the lessons learned from the survey are presented. An increasing amount of research shows the significance of this field. The research studies cover the use of traditional and more sophisticated deep learning approaches for attack detection. Traditional approaches such as LADS and MULTOPS have a fast detection speed, since only the measured traffic volume is compared against a pre-defined threshold. The major challenge is the choice and selection of the right threshold. As observed in the study presented in [55], only a few traces of DDoS attacks can be detected when a threshold established at 26 Mbps is used on an ISP network. The threshold selection issues make this approach prone to false alarms, which affect the detection accuracy. Therefore, the applicability of this approach to present-day security threats is constrained by detection accuracy and false alarm rates. The approach based on the correlation of the attacker's and the target server's IP addresses is very simple, as it only compares non-uniformity in the IP addresses. This makes the approach record excellent accuracy for attack detection, as noted in the research studies in [55–58]. This is not the case with modern attack strategies, where the same IP address can be used to launch attacks. The attack is launched in such a way that there is no significant dispersion of the IPs, which makes the correlation coefficient unaffected. This makes it difficult for an attack to be detected using this approach. Several other studies [49–54] have focused on the use of network traffic patterns to identify anomalies, since attack traffic will have a different pattern from that of legitimate ones. Despite the good precision recorded by this method, as reported in [50–52], the approach suffers from inaccurate flow feature selection. In view of this, machine learning, for example [53], is used to overcome this limitation. A major bottleneck reported is the difficulty in detecting attacks when the protocol headers are encrypted.

Research studies have also demonstrated the use of queue modelling [76–82], Chi-square [86–89], and chaos-based [116–121] approaches for DDoS attack detection, with some level of success. DDoS attack detection based on queue modelling has a fast response time once the maximum length of the queue is exceeded. The method, meanwhile, struggles with detecting accuracy under large-scale DDoS attacks. Additionally, this method frequently has significant false negative rates. The assumptions of independence used in Chi-square methods rarely apply to packet field values, even under normal conditions. As a result, it is possible that the Chi-square method does not accurately estimate how far a current traffic profile deviates from the baseline. Additionally, this method does not match real-life traffic patterns over an internet-enabled network. It is well known that the frequency of each symbol, which provides packet header information such as the source IP and packet length, exhibits power law behaviour [206]. The chi-square equation obeys the (B-1) degrees of freedom of the χ^2 distribution. As a result, directly adapting the χ^2 method for incoming packets is difficult. In the chaos-based DDoS attack detection studies, it is observed that an estimate of the Lyapunov exponent in the network traffic orbit is used to determine the attack. The exponent is calculated during a time window. As a result, the detection accuracy is severely influenced by the window size selection. A study published by [89] revealed that a small window size is sufficient to identify an amplification attack, but at the expense of computational resources. Although the authors in [207] investigated a detection algorithm for fixing the detection window size, the detection accuracy is relatively low, with some missed cases. Even though computational power has increased, the algorithm cannot reliably detect attacks in an anonymous and encrypted network environment. Thus, an adaptive window size may improve the accuracy of the approach. This means that the trade-off between window size and computing power cost requires additional research.

Entropy-based DDoS attack detection methods only require a minimal amount of network packet header information to construct patterns of legitimate traffic, as noticed in several studies [61–67]. Thus, they require fewer resources and enable quick and relatively more accurate detection. However, these methods strongly depend on the use of thresholds to achieve the desired detection results. It can be difficult to choose the right detection threshold in various attack environments due to the dynamic nature of network traffic patterns and rising attack intensities. Therefore, this approach requires a self-adaptive threshold, which is currently challenging to establish. Entropy values are also proven to be unable to distinguish between distinct traffic feature distributions with the same degree of uncertainty. As a result, anomalies that are unrelated to chance are missed. Similarly, the heuristic-based detection system [122–125] also requires a detection threshold to achieve a result. Though the threshold decision is optimized, as noted in [122], the DDoS defence of HBDS is based on an adjustable threshold. This means each approach might need to determine its own threshold to assess the currently observed traffic. In addition, HBDS also consumes computational resources such as CPUs and memory. The findings also reveal that most of these approaches could not effectively analyse traffic features, which affected their detection accuracy. To improve on this, machine learning techniques are currently being used because they can learn the characteristics of traffic and create a very precise model for identifying anomalous traffic features. Despite the relatively higher detection accuracy that machine learning algorithms have achieved, traditional machine learning models, such as SVM [127], k-NN [134], DT [135], NB [149], RF [158], etc., are shallow learners with a high error rate due to false alarms and changes in network traffic. This might restrict how many of these models can be used in real-world scenarios. Additionally, the features chosen from the datasets have a big impact on the detection accuracy. Results presented in Figure 15 show that the level of accuracy recorded by these models varies. Although this strongly depends on the quality of the dataset used and the application environment, it is evident that, even when using the same network dataset, different feature selections can produce very diverse results. The size of network traffic is rapidly expanding due to the established effectiveness of IoT technology with big data. Thus, conventional machine learning models may struggle to work with massive amounts of data due to their limited ability to learn features.

Additionally, as data volumes increase, the likelihood of false alarms rises dramatically. The structure of deep learning models has overcome the feature engineering problem found in the shallow machine learning models. This permits them to manage enormous datasets and learn more complex patterns within those datasets [163]. However, because there are so many layers being used, a long processing time is expected. DDoS attacks should be detected as quickly as possible, which has been a significant problem for deep learning models. Furthermore, they increase the processing overhead, making them difficult to use in IoT devices.

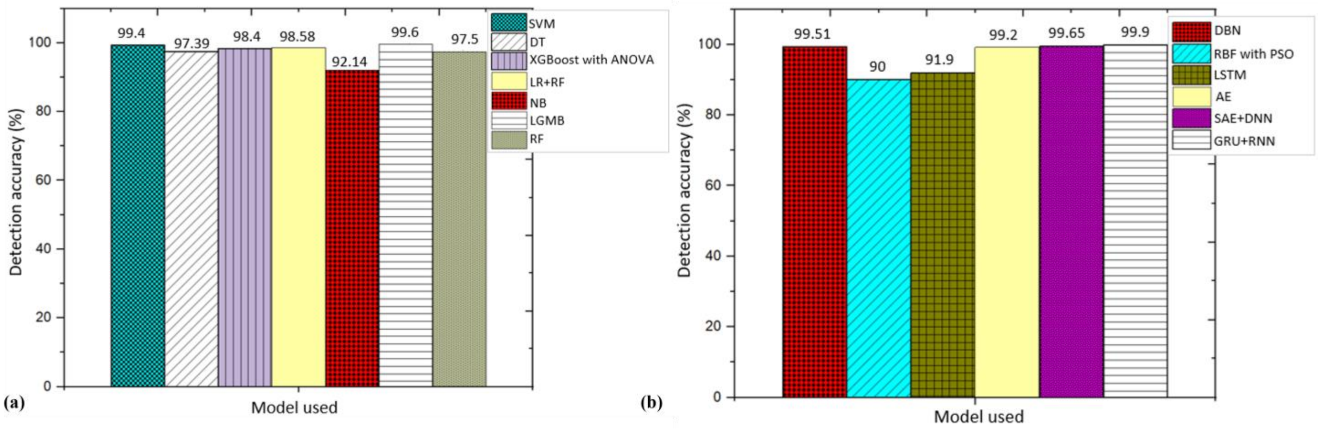

Figure 15. Detection accuracies observed from different research studies: (**a**) conventional machine learning models such as SVM [127], DT [135], XGBoost with ANOVA [140], LR with RF [141], NB [149], LGMB [150], and RF [158], (**b**) deep learning models such as DBN [11], RBF with PSO [154], LSTM [166], AE [170], SAE with DNN [176], and GRU with RNN [178].

6. Challenges and Future Research Directions

This section summarizes the current challenges identified from the research studies on DDoS attack detection methods. It can be noticed that several research studies are available in the literature that present the application of various methodologies for attack detection and demonstrate the significance of this research field. According to the literature, each methodology and research study has limitations and applicability. Six research challenges that could be the focus of future research studies were identified.

6.1. Detection Speed and Computational Overhead

Some detection methods employing entropy and statistical methods have fast detection speeds and low computational overhead. However, in most cases, the accuracy result is not sufficient for practical applications. In addition, the high false alarm rates associated with these methods limit their application in real-world scenarios. Nevertheless, machine learning and deep learning have improved detection accuracy but suffer from longer processing times. Deep learning uses more layers to achieve better performance, thus increasing the processing time. Balancing the trade-off between accuracy and detection speed remains a future research problem.

6.2. Real-Time Realization under Resource Constraints for IoT Devices

Most of the DDoS detection presented ignores real-time scenarios in an IoT environment to achieve the desired results. To solve real-time problems, modern smart devices have data processing abilities that may permit the use of a deep learning model to learn the processed data. Deep learning models demand that the machines be provided with adequate processing capacity, such as graphic processing units (GPUs), which consume a lot of power. IoT devices are well known for having limited power and size. In addition, the UAVs used in the IoFT and IoD networks have very limited computational resources. As a result, the potential of deep learning cannot be fully leveraged, due to the long processing time and computational overhead that could deplete the power of IoT devices and other UAVs used in these networks. Even reducing the training time for GPU acceleration is difficult. Thus, maintaining security under these conditions is a major concern.

6.3. Adaptive Threshold and Feature Selection

It became evident that different threshold and feature selection methods can produce diverse results, even with the same attack detection dataset. The performance of entropy- and statistical-based methods strongly depends on the choice of threshold and feature selection. Since traffic features are rapidly changing due to new attack structures and growing attack intensities, the threshold should be self-adaptive, based on the new features. However, it is expected that more computational overhead will be added due to the adaptive threshold and feature selection algorithms integrated, so it is crucial to strike a balance between adaptive threshold and computational overhead while maintaining reasonable detection speed and accuracy. This is currently lacking in the literature and would be a good focus for future research.

6.4. Self-Learning and Adaptation

In recent times, the pattern of attack has changed, and more attack vectors are now being employed to launch DDoS attacks. As a result, new attack scenarios with different features are evolving daily. It is possible that the features identified to detect one category of attack vector may not be sufficient to successfully detect another category of attack. In addition, the frequency and pattern of DDoS attacks are expected to increase with the development of 5G networks, which have made it possible for more devices to be connected. Hence, there is a need to further develop a DDoS attack detection framework that can swiftly learn and adapt to changing attack patterns, while transiting from the current-day network to 5G and beyond, with reduced computational time and increased accuracy. Further investigation in this area is essential.

6.5. Data Quality Issues

As noticed from the survey work, the current approaches for DDoS attack detection employ machine and deep learning models due to their ability to learn traffic patterns quickly with good detection accuracy. These models, however, heavily rely on the dataset. The quality as well as the size of the datasets used to train the algorithms are crucial for better performance. The quality of the training datasets plays an essential role in developing an efficient DDoS attack detection method. However, a fundamental issue with this approach is the lack of readily accessible, high-quality datasets. In some cases, there is a data class imbalance that occasionally reduces the detection accuracy of these models. Although the data class imbalance is the current focus in some studies [208–211], ensuring a sufficient high-quality dataset would improve the accuracy of DDoS attack detection methods and should be a focus for future studies. Further investigation in this direction should be sustained, which is a future research direction. In addition, most of the datasets are outdated, have low traffic diversity, and are unreliable for modern attack detection methods.

6.6. Lack of Real-Time Datasets

This research area suffers from a lack of real-time datasets for attack scenarios. It is unfortunate that access to real-time data is constrained by a number of ethical issues. The only real-world datasets still in existence are Kyoto, UNIBS, Kent2016, and ISCX2012, which are outdated. Existing datasets are obviously too obsolete to capture the latest attack features, as novel attack types are constantly appearing. The commonly used datasets for DDoS attack detection research, such as CICIDS2017 and CISDDOS2019, are generated in a simulated environment or synthesized. Although the simulated data are appropriate for research purposes, in most cases, a simplified simulation of an internet-enabled environment is typically utilized, which does not accurately depict the heterogeneous nature of contemporary internet-enabled smart devices. Additionally, they might not account for the evolution of new DDoS attack types on real networks. Furthermore, the issue of attack detection in the internet of flying things and drone networks is a recent one. As a result, the dataset in this field is very sparse. Some research studies [14,146,147,176,179] have adopted the regularly used datasets in other internet-enabled networks; unfortunately, this may not accurately represent genuine attack scenarios in drone networks. This hinders the advancement of research in this field. In addition, real-time datasets to be used for machine and deep learning for attack detection in RPL and NDN networks are unavailable. Research studies in this domain make use of simulated data, which do not accurately reflect the dynamism of current internet traffic. There is a dire need for studies to be conducted in this area. It is necessary to create realistic and traffic-dynamic datasets which are capable of simulating the heterogeneous traffic of a DDoS attack in an internet-enabled environment. Moreover, the datasets should be balanced, with reduced noise and redundancy.

7. Conclusions and Limitations

This paper aims to offer an overview of the current DDoS attack detection methods and a survey of research perspectives in this domain. DDoS attack detection methods are classified based on how they are used. Different research studies under each classification are presented. DDoS attack detection studies, which range from a conventional entropy-based approach to more sophisticated deep learning models, are extensively covered. These approaches have all been used, with varying levels of effectiveness and restrictions. Although information-entropy-based DDoS attack detection approaches use fewer resources, they all require selecting the right threshold to achieve the desired detection results. It can be challenging to choose the right detection threshold in various attack environments, since different networks have different traffic patterns. The features of network traffic can be understood by machine learning and deep learning algorithms, which can then be trained to create a very precise model for detecting these features. However, they have more computational overhead, which limits their applicability to resource-constrained IoT devices. The relevance of research advancements in this field is demonstrated by the sizeable amount of research now being carried out. It is clear from this review that the approaches currently in use can identify DDoS attacks with varying degrees of accuracy. However, every approach has one or more limitations to overcome. Thus, more research studies are still required in this domain. With the advent of 5G and beyond, which will enable the connectivity of more devices, more attacks with varying attack patterns are anticipated, making improved attack detection more important. This paper has identified six research gaps that could influence future research on improving attack detection performance. Even though this survey concentrates primarily on DDoS attack types and detection studies in the IoT, studies focusing on attack detection in SDN, RPL, NDN, and vehicular (the internet of flying things) are also researched and presented. However, detailed discussions on the attack types in RPL, NDN, and the internet of flying things are part of the limitations of this study. In addition, this study is limited to a few carefully chosen datasets, even though they are thought to be the most pertinent and related to the research topic of this study. However, other surveys may also include some additional datasets pertinent to this subject.

Author Contributions: K.B.A. conceived the original idea of the paper and was in charge of the manuscript draft, while A.M.A.-M. and A.M.K. helped with some improvements to the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledged the resources provided by the Tshwane University of Technology, Pretoria, and CSIR, Pretoria, South Africa.

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

References

1. Kiran, S.; Sriramoju, S.B. A study on the applications of IoT. *Indian J. Public Health Res. Dev.* **2018**, *9*, 1173–1175. [CrossRef]
2. Khan, M.A. Challenges facing the application of IoT in medicine and healthcare. *Int. J. Comput. Inf. Manuf.* **2021**, *1*, 39–55. [CrossRef]
3. Banafshehvaragh, S.T.; Rahmani, A.M. Intrusion, anomaly, and attack detection in smart vehicles. *Microprocess. Microsyst.* **2023**, *96*, 104726. [CrossRef]

4. Svaigen, A.R.; Boukerche, A.; Ruiz, L.B.; Loureiro, A.A. Trajectory Matters: Impact of jamming attacks over the drone path planning on the internet of drones. *Ad Hoc Netw.* **2023**, *146*, 103179. [CrossRef]
5. Rahman, K.; Aziz, M.A.; Usman, N.; Kiren, T.; Cheema, T.A.; Shoukat, H.; Bhatia, T.K.; Abdollahi, A.; Sajid, A. Cognitive lightweight logistic regression-based IDS for IoT-enabled FANET to detect cyberattacks. *Mob. Inf. Syst.* **2023**, *2023*, 7690322. [CrossRef]
6. Almasoud, A. Jamming-aware optimization for UAV trajectory design and internet of things devices clustering. *Complex Intell. Syst.* **2023**, 1–20. [CrossRef]
7. Srivastava, A.; Prakash, J. Internet of low-altitude UAVs (IoLoUA): A methodical modelling on integration of internet of "things" with "UAV" possibilities and tests. *Artif. Intell. Rev.* **2023**, *56*, 2279–2324. [CrossRef]
8. Mykytyn, P.; Brzozowski, M.; Dyka, Z.; Langendoerfer, P. GPS-spoofing attack detection mechanism for UAV swarms. *arXiv* **2023**, arXiv:2301.12766.
9. Mekdad, Y.; Aris, A.; Babun, L.; El Fergougui, A.; Conti, M.; Lazzeretti, R.; Uluagac, A.S. A survey on security and privacy issues of UAVs. *Comput. Netw.* **2023**, *224*, 109626. [CrossRef]
10. Wu, S.; Li, Y.; Wang, Z.; Tan, Z.; Pan, Q. A highly interpretable framework for generic low-cost UAV attack detection. *IEEE Sens. J.* **2023**, *23*, 7288–7300. [CrossRef]
11. Xie, Z.; Li, Z.; Gui, J.; Liu, A.; Xiong, N.N.; Zhang, S. UWPEE: Using UAV and wavelet packet energy entropy to predict traffic-based attacks under limited communication, computing and caching for 6G wireless systems. *Future Gener. Comput. Syst.* **2023**, *140*, 238–252. [CrossRef]
12. Mohsan, S.A.H.; Othman, N.Q.H.; Li, Y.; Alsharif, M.H.; Khan, M.A. Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends. *Intell. Serv. Robot.* **2023**, *2023*, 109–137. [CrossRef] [PubMed]
13. Nayfeh, M.; Li, Y.; Al Shamaileh, K.; Devabhaktuni, V.; Kaabouch, N. Machine learning modelling of GPS features with applications to UAV location spoofing detection and classification. *Comput. Secur.* **2023**, *126*, 103085. [CrossRef]
14. Escorcia-Gutierrez, J.; Gamarra, M.; Leal, E.; Madera, N.; Soto, C.; Mansour, R.F.; Alharbi, M.; Alkhayyat, A.; Gupta, D. Sea turtle foraging algorithm with hybrid deep learning-based intrusion detection for the internet of drones environment. *Comput. Electr. Eng.* **2023**, *108*, 108704. [CrossRef]
15. Altaweel, A.; Mukkath, H.; Kamel, I. GPS Spoofing attacks in FANETs: A systematic literature review. *IEEE Access* **2023**, *11*, 55233–55280. [CrossRef]
16. Wei, X.; Aman, M.N.; Sikdar, B. A Light-Weight Technique to Detect GPS Spoofing Using Attenuated Signal Envelopes. *IEEE Open J. Comput. Soc.* **2023**, *4*, 158–170. [CrossRef]
17. Tong, F.; Zhang, Z.; Zhu, Z.; Zhang, Y.; Chen, C. A novel scheme based on coarse-grained localization and fine-grained isolation for defending against Sybil attack in low power and lossy networks. *Asian J. Control* **2023**, *2023*, 1–12. [CrossRef]
18. Bang, A.; Rao, U.P. Performance evaluation of RPL protocol under decreased and increased rank attacks: A focus on smart home use-case. *SN Comput. Sci.* **2023**, *4*, 329. [CrossRef]
19. Babu, V.J.; Jose, M.V. Dynamic forest of random subsets-based one-time signature-based capability enhancing security architecture for named data networking. *Int. J. Inf. Technol.* **2023**, *15*, 773–788. [CrossRef]
20. F5. DDoS Architecture Diagram and White Paper. 2020. Available online: https://www.f5.com/services/resources/white-papers/the-f5-ddos-protection-reference-architecture (accessed on 15 November 2022).
21. Gil, T.M.; Poletto, M. MULTOPS: A data-structure for bandwidth attack detection. In Proceedings of the 10th USENIX Security Symposium, Washington, DC, USA, 13–17 August 2001.
22. Waizumi, Y.; Sato, T.; Nemoto, Y. A new traffic pattern matching for DDoS traceback using independent component analysis. *World Acad. Sci. Eng. Technol.* **2011**, *60*, 760–766.
23. Zargar, S.T.; Joshi, J.; Tipper, D. A survey of defence mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2046–2069. [CrossRef]
24. Sonar, K.; Upadhyay, H. A survey: DDOS attack on internet of things. *Int. J. Eng. Res. Dev.* **2014**, *10*, 58–63.
25. Kaur, P.; Kumar, M.; Bhandari, A. A review of detection approaches for distributed denial of service attacks. *Syst. Sci. Control Eng.* **2017**, *5*, 301–320. [CrossRef]
26. Kamboj, P.; Trivedi, M.C.; Yadav, V.K.; Singh, V.K. Detection techniques of DDoS attacks: A survey. In Proceedings of the 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics, Mathura, India, 26–28 October 2017; pp. 675–679.
27. Abdul-Ghani, H.A.; Konstantas, D.; Mahyoub, M. A comprehensive IoT attacks survey based on a building-blocked reference model. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 355–373.
28. Alhajri, R.; Zagrouba, R.; Al-Haidari, F. Survey for anomaly detection of IoT botnets using machine learning auto-encoders. *Int. J. Appl. Eng. Res.* **2019**, *14*, 2417–2421.
29. Khalaf, B.A.; Mostafa, S.A.; Mustapha, A.; Mohammed, M.A.; Abduallah, M.W. Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. *IEEE Access* **2019**, *7*, 51691–51713. [CrossRef]
30. 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]

31. Tayyab, M.; Belaton, B.; Anbar, M. ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review. *IEEE Access* **2020**, *8*, 170529–170547. [CrossRef]
32. Nooribakhsh, M.; Mollamotalebi, M. A review on statistical approaches for anomaly detection in DDoS attacks. *Inf. Secur. J. A Glob. Perspect.* **2020**, *29*, 118–133. [CrossRef]
33. Haji, S.H.; Ameen, S.Y. Attack and anomaly detection in IoT networks using machine learning techniques: A review. *Asian J. Res. Comput. Sci.* **2021**, *9*, 30–46. [CrossRef]
34. Huang, K.; Yang, L.Y.; Yang, X.; Xiang, Y.; Tang, Y.Y. A low-cost distributed denial-of-service attack architecture. *IEEE Access* **2020**, *8*, 42111–42119. [CrossRef]
35. De Donno, M.; Giaretta, A.; Dragoni, N.; Spognardi, A. A taxonomy of distributed denial of service attacks. In Proceedings of the IEEE International Conference on Information Society, Dublin, Ireland, 17–19 July 2017; pp. 100–107.
36. Shorey, T.; Subbaiah, D.; Goyal, A.; Sakxena, A.; Mishra, A.K. Performance comparison and analysis of slowloris, goldeneye and xerxes DDoS attack tools. In Proceedings of the IEEE International Conference on Advances in Computing, Communications and Informatics, Bangalore, India, 19–22 September 2018; pp. 318–322.
37. Douligeris, C.; Mitrokotsa, A. DDoS attacks and defense mechanisms; classification and state-of-the-art. *Compt. Netw.* **2004**, *44*, 643–666. [CrossRef]
38. Singh, K.J.; De, T. Mathematical modelling of DDoS attack and detection using correlation. *J. Cyber Secur. Technol.* **2017**, *1*, 175–186. [CrossRef]
39. Luo, J.; Yang, X.; Wang, J.; Xu, J.; Sun, J.; Long, K. On a mathematical model for low-rate shrew DDoS. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 1069–1083. [CrossRef]
40. Akamai. Threat Advisory: Internet of Things and the Rise of 300 Gbps DDoS Attacks. Available online: https://www.akamai.com/us/en/multimedia/documents/social/q4-state-of-the-internet-security-spotlight-iot-rise-of-300-gbp-ddos-attacks.pdf (accessed on 16 December 2022).
41. Ibrahim, R.F.; Abu Al-Haija, Q.; Ahmad, A. DDoS attack prevention for internet of thing devices using ethereum blockchain technology. *Sensors* **2022**, *22*, 6806. [CrossRef]
42. Shroff, J.; Walambe, R.; Singh, S.K.; Kotecha, K. Enhanced security against volumetric DDoS attacks using adversarial machine learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5757164. [CrossRef]
43. Salim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. *J. Supercomput.* **2020**, *76*, 5320–5363. [CrossRef]
44. Erhan, D.; Anarim, E. Hybrid DDoS detection framework using matching pursuit algorithm. *IEEE Access* **2020**, *8*, 118912–118923. [CrossRef]
45. Praseed, A.; Thilagam, P.S. DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 661–685. [CrossRef]
46. F5 Labs. DDoS Attack Trends for 2020. Available online: https://www.f5.com/labs/articles/threat-intelligence/ddos-attack-trends-for-2020 (accessed on 19 November 2022).
47. Nexus Guard. Threat Report Distributed Denial of Service. 2018. Available online: https://www.nexusguard.com/hubfs/Threat%20Report%20Q2%202018/Nexusguard_DDoS_Threat_Report_Q2_2018_EN.pdf (accessed on 31 December 2022).
48. Sekar, V.; Duffield, N.G.; Spatscheck, O.; van der Merwe, J.E.; Zhang, H. LADS: Large-scale automated DDoS detection system. In Proceedings of the USENIX Annual Technical Conference, Boston, MA, USA, 30 May–3 June 2006; pp. 171–184.
49. Shafiq, M.Z.; Ji, L.; Liu, A.X.; Pang, J.; Wang, J. Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM Trans. Netw.* **2013**, *21*, 1960–1973. [CrossRef]
50. Moore, A.W.; Zuev, D. Internet traffic classification using Bayesian analysis techniques. In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, Banff, AB, Canada, 6–10 June 2005; pp. 50–60.
51. Lima Filho, F.S.D.; Silveira, F.A.; de Medeiros Brito Junior, A.; Vargas-Solar, G.; Silveira, L.F. Smart detection An online approach for DoS/DDoS attack detection using machine learning. *Secur. Commun. Netw.* **2019**, *2019*, 1574749. [CrossRef]
52. Shafiq, M.; Yu, X.; Bashir, A.K.; Chaudhry, H.N.; Wang, D. A machine learning approach for feature selection traffic classification using security analysis. *J. Supercomput.* **2018**, *74*, 4867–4892. [CrossRef]
53. Wu, Y.C.; Tseng, H.R.; Yang, W.; Jan, R.H. DDoS detection and traceback with decision tree and grey relational analysis. *Int. J. Ad Hoc Ubiquitous Comput.* **2011**, *7*, 121–136. [CrossRef]
54. Krasnov, A.E.; Nikol'Skii, D.N.; Repin, D.S.; Galyaev, V.S.; Zykova, E.A. Detecting DDoS attacks using the analysis of network traffic as dynamical system. In Proceedings of the IEEE International Scientific and Technical Conference Modern Computer Network Technologies, Moscow, Russia, 27–28 October 2018; pp. 1–7.
55. Guo, F.; Chen, J.; Chiueh, T.C. Spoof detection for preventing dos attacks against DNS servers. In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, Lisboa, Portugal, 4–7 July 2006; p. 37.
56. Wang, Z.; Wang, X. DDoS attack detection algorithm based on the correlation of IP address analysis. In Proceedings of the IEEE International Conference on Electrical and Control Engineering, Yichang, China, 16–18 September 2011; pp. 2951–2954.
57. ren Cheng, J.; ping Yin, J. Distributed denial of service attack detection method based on address correlation. *Comput. Res. Dev.* **2009**, *46*, 1334–1340.

58. Xiao, P.; Qu, W.; Qi, H.; Li, Z. Detecting DDoS attacks against data center with correlation analysis. *Comput. Commun.* **2015**, *67*, 66–74. [CrossRef]
59. Rastegari, S.; Saripan, M.I.; Rasid, M.F.A. Detection of denial-of-service attacks against domain name system using neural networks. *Int. J. Comput. Sci. Issues* **2009**, *6*, 23–27.
60. Saied, A.; Overill, R.E.; Radzik, T. Detection of known and unknown DDoS attacks using artificial neural networks. *Neurocomputing* **2016**, *172*, 385–393. [CrossRef]
61. Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Comput. Netw.* **2014**, *62*, 122–136. [CrossRef]
62. Rahmani, H.; Sahli, N.; Kammoun, F. Joint entropy analysis of DDoS attack detection. In Proceedings of the 5th IEEE International Conference on Information Assurance and Security, Washington, DC, USA, 18–20 August 2009; pp. 267–271.
63. Gaurav, A.; Gupta, B.B.; Hsu, C.H.; Yamaguchi, S.; Chui, K.T. Fog layer-based DDoS attack detection approach for internet-of-things (IoTs) devices. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 10–12 January 2021; pp. 1–5.
64. Gaurav, A.; Gupta, B.B.; Hsu, C.H.; Peraković, D.; Peñalvo, F.J.G. Filtering of distributed denial of services (DDoS) attacks in cloud computing environment. In Proceedings of the IEEE International Conference on Communications Workshops, Montreal, QC, Canada, 14–18 June 2021; pp. 1–6.
65. Lakhina, A.; Crovella, M.; Diot, C. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Comput. Commun. Rev.* **2005**, *35*, 217–228. [CrossRef]
66. Li, J.; Liu, M.; Xue, Z.; Fan, X.; He, X. RTVD: A real-time volumetric detection scheme for DDoS in the internet of things. *IEEE Access* **2020**, *8*, 36191–36201. [CrossRef]
67. David, J.; Thomas, C. DDoS attack detection using fast entropy approach on flow-based network traffic. *Procedia Comput. Sci.* **2015**, *50*, 30–36. [CrossRef]
68. David, J.; Thomas, C. Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic. *Comput. Secur.* **2019**, *82*, 284–295. [CrossRef]
69. Winter, P.; Lampesberger, H.; Zeilinger, M.; Hermann, E. On detecting abrupt changes in network entropy time series. In Proceedings of the IFIP International Conference on Communications and Multimedia Security, Ghent, Belgium, 19–21 October 2011; pp. 194–205.
70. Qin, X.; Xu, T.; Wang, C. DDoS attack detection using flow entropy and clustering technique. In Proceedings of the 11th IEEE International Conference on Computational Intelligence and Security, Shenzhen, China, 19–20 December 2015; pp. 412–415.
71. Koay, A.; Chen, A.; Welch, I.; Seah, W.K. A new multi classifier system using entropy-based features in DDoS attack detection. In Proceedings of the IEEE International Conference on Information Networking, Chiang Mai, Thailand, 10–12 January 2018; pp. 162–167.
72. Nychis, G.; Sekar, V.; Andersen, D.G.; Kim, H.; Zhang, H. An empirical evaluation of entropy-based traffic anomaly detection. In Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, Vouliagmeni, Greece, 20–22 October 2008; pp. 151–156.
73. Bhalodiya, S.; Vaghela, K. Enhanced detection and recovery from flooding attack in MANETs using AODV routing protocol. *Int. J. Comput. Appl.* **2015**, *125*, 10–15. [CrossRef]
74. Singh, N.; Ghrera, S.P.; Chaudhuri, P. Denial of service attack: Analysis of network traffic anomaly using queuing theory. *J. Comput. Sci. Eng.* **2010**, *1*, 48–51.
75. Little, J.D.C.; Graves, S.C. Little's law. In *Building Intuition*; Chhajed, D., Lowe, T.J., Eds.; Springer: New York, NY, USA, 2008; pp. 81–100.
76. Syed, N.F.; Baig, Z.; Ibrahim, A.; Valli, C. Denial of service attack detection through machine learning for the IoT. *J. Inf. Telecommun.* **2020**, *4*, 482–503. [CrossRef]
77. Ramanauskaitė, S.; Čenys, A.; Goranin, N.; Janulevicius, J. Modelling of two-tier DDoS by combining different type of DDoS models. In Proceedings of the IEEE Open Conference of Electrical, Electronic and Information Sciences, Vilnius, Lithuania, 27 April 2017; pp. 1–4.
78. Rastogi, S.; Zaheer, H. Comparative analysis of queuing mechanisms: Droptail, RED and NLRED. *Soc. Netw. Anal. Min.* **2016**, *6*, 70. [CrossRef]
79. Serrano, J.B.; Wang, S.; Chavez, K.M.G.; Hourani, A.; Sithamparanathan, K. A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks. *Eng. Sci. Technol. Int. J.* **2022**, *31*, 101065.
80. Hao, S.; Song, H.; Jiang, W.; Dai, Y. A queue model to detect DDos attacks. In Proceedings of the IEEE International Symposium on Collaborative Technologies and Systems, Saint Louis, MO, USA, 15–20 May 2005; pp. 106–112.
81. Khan, S.; Traore, I. Queue-based analysis of DoS attacks. In Proceedings of the 6th Annual IEEE SMC Information Assurance Workshop, West Point, NY, USA, 15–17 June 2005; pp. 266–273.
82. Jeong, S.; Kim, H.; Kim, S. An effective DDoS attack detection and packet-filtering scheme. *IEICE Trans. Commun.* **2006**, *89*, 2033–2042. [CrossRef]
83. Lin, H.Y.; Chiang, T.C. Intrusion detection mechanisms based on queuing theory in remote distribution sensor networks. *Adv. Mater. Res.* **2010**, *121*, 58–63. [CrossRef]

84. Hussain, S.M.; Beigh, G.R. Impact of DDoS attack (UDP Flooding) on queuing models. In Proceedings of the 4th IEEE International Conference on Computer and Communication Technology, Tiruchengode, India, 4–6 July 2013; pp. 210–216.
85. Wei, W.; Song, H.; Wang, H.; Fan, X. Research and simulation of queue management algorithms in ad hoc networks under DDoS attack. *IEEE Access* **2017**, *5*, 27810–27817. [CrossRef]
86. Feinstein, L.; Schnackenberg, D.; Balupari, R.; Kindred, D. Statistical approaches to DDoS attack detection and response. In Proceedings of the DARPA Information Survivability Conference and Exposition, Washington, DC, USA, 22–24 April 2003; pp. 303–314.
87. Abouzakhar, N.; Bakar, A. A Chi-square testing-based intrusion detection model. In Proceedings of the 4th International Conference on Cybercrime Forensics Education & Training, Canterbury, UK, 2–3 September 2010.
88. Leu, F.Y.; Lin, L.L. A DoS/DDoS attack detection system using chi-square statistic approach. *J. Syst. Cybern. Inform.* **2010**, *8*, 41–51.
89. Ye, N.; Chen, Q. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Qual. Reliab. Eng. Int.* **2001**, *17*, 105–112. [CrossRef]
90. Siris, V.A.; Papagalou, F. Application of anomaly detection algorithms for detecting SYN flooding attacks. *Comput. Commun.* **2006**, *29*, 1433–1442. [CrossRef]
91. Machaka, P.; Bagula, A.; Nelwamondo, F. Using exponentially weighted moving average algorithm to defend against DDoS attacks. In Proceedings of the IEEE Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, Stellenbosch, South Africa, 30 November–2 December 2016; pp. 1–6.
92. Zhan, S.; Tang, D.; Man, J.; Dai, R.; Wang, X. Low-rate dos attacks detection based on MAF-ADM. *Sensors* **2020**, *20*, 189. [CrossRef] [PubMed]
93. Shinde, P.; Guntupalli, S. Early DoS attack detection using smoothened time-series and wavelet analysis. In Proceedings of the IEEE the 3rd International Symposium on Information Assurance and Security, Manchester, UK, 29–31 August 2007; pp. 215–220.
94. De Moura, A.S. Anomaly detection using Holt-Winters forecast model. In Proceedings of the IADIS International Conference WWW/Internet, Rio De Janeiro, Brazil, 5–8 November 2011; pp. 349–356.
95. Zhang, G.; Jiang, S.; Wei, G.; Guan, Q. A prediction-based detection algorithm against distributed denial-of-service attacks. In Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, Leipzig, Germany, 21–24 June 2009; pp. 106–110.
96. Yaacob, A.H.; Tan, I.K.T.; Chien, S.F.; Tan, H.K. ARIMA based network anomaly detection. In Proceedings of the IEEE 2nd International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 205–209.
97. Nezhad, S.M.T.; Nazari, M.; Gharavol, E.A. A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks. *IEEE Commun. Lett.* **2016**, *20*, 700–703. [CrossRef]
98. Barbhuiya, S.; Kilpatrick, P.S.; Nikolopoulos, D. Linear regression-based DDoS attack detection. In Proceedings of the 13th International Conference on Machine Learning and Computing, Shenzhen, China, 26 February–1 March 2021; pp. 568–574.
99. Fachkha, C.; Bou-Harb, E.; Debbabi, M. Towards a forecasting model for distributed denial of service activities. In Proceedings of the IEEE 12th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 22–24 August 2013; pp. 110–117.
100. Khan, M.S.; Ferens, K.; Kinsner, W. A chaotic measure for cognitive machine classification of distributed denial of service attacks. In Proceedings of the IEEE 13th International Conference on Cognitive Informatics and Cognitive Computing, London, UK, 18–20 August 2014; pp. 100–108.
101. Chen, C.L. A new detection method for distributed denial-of-service attack traffic based on statistical test. *J. Univ. Comput. Sci.* **2009**, *15*, 488–504.
102. Machaka, P.; McDonald, A.; Nelwamondo, F.; Bagula, A. Using the cumulative sum algorithm against distributed denial of service attacks in internet of things. In Proceedings of the 4th EAI International Conference on Context-Aware Systems and Applications, Ho Chi Minh City, Vietnam, 26–27 November 2015; pp. 62–72.
103. Zhang, T. Cumulative sum algorithm for detecting SYN flooding attacks. *arXiv* **2012**, arXiv:1212.5129.
104. Özcelik, I.; Brooks, R.R. Cusum-entropy: An efficient method for DDoS attack detection. In Proceedings of the 4th IEEE International Istanbul Smart Grid Congress and Fair, Istanbul, Turkey, 20–21 April 2016; pp. 1–5.
105. Udhayan, J.; Hamsapriya, T. Statistical segregation method to minimize the false detections during DDoS attacks. *Int. J. Netw. Secur.* **2011**, *13*, 152–160.
106. Tan, Z.; Jamdagni, A.; He, X.; Nanda, P.; Liu, R.P. A system for denial-of-service attack detection based on multivariate correlation analysis. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 447–456.
107. Jin, S.; Yeung, D.S. A covariance analysis model for DDoS attack detection. In Proceedings of the IEEE International Conference on Communications, Paris, France, 20–24 June 2004; pp. 1882–1886.
108. Fortunati, S.; Gini, F.; Greco, M.S.; Farina, A.; Graziano, A.; Giompapa, S. An improvement of the state-of-the-art covariance-based methods for statistical anomaly detection algorithms. *Signal Image Video Process.* **2016**, *10*, 687–694. [CrossRef]
109. Peng, T.; Leckie, C.; Ramamohanarao, K. Detecting distributed denial of service attacks by sharing distributed beliefs. In *Information Security and Privacy*; Lecture Notes in Computer Science; Safavi-Naini, R., Seberry, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2727.
110. Hoque, H.; Bhattacharyya, D.K.; Kalita, J.K. FFSc: A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. *Secur. Commun. Netw.* **2016**, *9*, 2032–2041. [CrossRef]

111. Grimit, E.P.; Gneiting, T.; Berrocal, V.J.; Johnson, N.A. The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification. *Q. J. R. Meteorol. Soc. A J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr.* **2006**, *132*, 2925–2942.
112. Bouyeddou, B.; Kadri, B.; Harrou, F.; Sun, Y. DDOS-attacks detection using an efficient measurement-based statistical mechanism. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 870–878. [CrossRef]
113. Harrou, F.; Sun, Y.; Madakyaru, M.; Bouyedou, B. An improved multivariate chart using partial least squares with continuous ranked probability score. *IEEE Sens. J.* **2018**, *18*, 6715–6726. [CrossRef]
114. Sharma, D.K.; Dhankhar, T.; Agrawal, G.; Singh, S.K.; Gupta, D.; Nebhen, J.; Razzak, I. Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Netw.* **2021**, *121*, 102603. [CrossRef]
115. Wolf, A.; Swift, J.B.; Swinney, H.L.; Vastano, J.A. Determining lyapunov exponents from a time series. *Phys. D Nonlinear Phenom.* **1992**, *16*, 285–317. [CrossRef]
116. Chonka, A.; Singh, J.; Zhou, W. Chaos theory-based detection against network mimicking DDoS attacks. *IEEE Commun. Lett.* **2009**, *13*, 717–719. [CrossRef]
117. Iyengar, N.C.S.N.; Ganapathy, G. Chaotic theory based defensive mechanism against distributed denial of service attack in cloud computing environment. *Int. J. Secur. Its Appl.* **2015**, *9*, 197–212. [CrossRef]
118. Chen, Y.; Ma, X.; Wu, X. DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Commun. Lett.* **2013**, *17*, 1052–1054. [CrossRef]
119. Ma, X.; Chen, Y. DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Commun. Lett.* **2013**, *18*, 114–117. [CrossRef]
120. Wu, X.; Chen, Y. Validation of chaos hypothesis in NADA and improved DDoS detection algorithm. *IEEE Commun. Lett.* **2013**, *17*, 2396–2399. [CrossRef]
121. Procopiou, A.; Komninos, N.; Douligeris, C. ForChaos: Real time application DDoS detection using forecasting and chaos theory in smart home IoT network. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 8469410. [CrossRef]
122. Kumar, P.A.R.; Selvakumar, S. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Comput. Commun.* **2013**, *36*, 303–319. [CrossRef]
123. Roopak, M.; Tian, G.Y.; Chambers, J. An intrusion detection system against DDoS attacks in IoT networks. In Proceedings of the 10th IEEE Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 6–8 January 2020; pp. 0562–0567.
124. Roopak, M.; Tian, G.Y.; Chambers, J. Multi-objective-based feature selection for DDoS attack detection in IoT networks. *IET Netw.* **2020**, *9*, 120–127. [CrossRef]
125. Yin, J.; Tao, T.; Xu, J. A multi-label feature selection algorithm based on multi-objective optimization. In Proceedings of the IEEE International Joint Conference on Neural Networks, Killarney, Ireland, 12–17 July 2015; pp. 1–7.
126. Saeed, A.A.; Jameel, N.G.M. Intelligent feature selection using particle swarm optimization algorithm with a decision tree for DDoS attack detection. *Int. J. Adv. Intell. Inform.* **2021**, *7*, 37–48. [CrossRef]
127. Velliangiri, S.; Karthikeyan, P.; Vinoth Kumar, V. Detection of distributed denial of service attack in cloud computing using the optimization-based deep networks. *J. Exp. Theor. Artif. Intell.* **2021**, *33*, 405–424. [CrossRef]
128. Varghese, M.; Victor Jose, M. An optimized radial bias function neural network for intrusion detection of distributed denial of service attack in the cloud. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7321. [CrossRef]
129. Sokkalingam, S.; Ramakrishnan, R. An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm-based approach. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7334. [CrossRef]
130. Amma, N.G.; Selvakumar, S. Optimization of vector convolutional deep neural network using binary real cumulative incarnation for detection of distributed denial of service attacks. *Neural Comput. Appl.* **2022**, *34*, 2869–2882. [CrossRef]
131. Alshamrani, A.; Chowdhary, A.; Pisharody, S.; Lu, D.; Huang, D. A defense system for defeating DDoS attacks in SDN based networks. In Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, Miami, FL, USA, 21–25 November 2017; pp. 83–92.
132. Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS attack detection method based on SVM in software defined network. *Secur. Commun. Netw.* **2018**, *2018*, 9804061. [CrossRef]
133. Khuphiran, P.; Leelaprute, P.; Uthayopas, P.; Ichikawa, K.; Watanakeesuntorn, W. Performance comparison of machine learning models for DDoS attacks detection. In Proceedings of the 22nd IEEE International Computer Science and Engineering Conference, Chiang Mai, Thailand, 21–24 November 2018; pp. 1–4.
134. Rahman, O.; Quraishi, M.A.G.; Lung, C.H. DDoS attacks detection and mitigation in SDN using machine learning. In Proceedings of the IEEE World Congress on Services, Milan, Italy, 8–13 July 2019; pp. 184–189.
135. Khashab, F.; Moubarak, J.; Feghali, A.; Bassil, C. DDoS attack detection and mitigation in SDN using machine learning. In Proceedings of the IEEE 7th International Conference on Network Softwarization, Tokyo, Japan, 28 June–2 July 2021; pp. 395–401.
136. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
137. Gopalan, S.S. Towards Effective Detection of Botnet Attacks Using BoT-IoT Dataset. Master's Thesis, Department of Computer Science, Rochester Institute of Technology, Rochester, NY, USA, 2021.

138. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and application layer DDos attacks detection to IoT devices by using machine learning and deep learning model. *Sensors* **2022**, *22*, 3367. [CrossRef] [PubMed]
139. Chen, Y.W.; Sheu, J.P.; Kuo, Y.C.; Van Cuong, V. Design and implementation of IoT DDoS attacks detection system based on machine learning. In Proceedings of the IEEE European Conference on Networks and Communications, Dubrovnik, Croatia, 15–18 June 2020; pp. 122–127.
140. Mihoub, A.; Fredj, O.B.; Cheikhrouhou, O.; Derhab, A.; Krichen, M. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Comput. Electr. Eng.* **2022**, *98*, 107716. [CrossRef]
141. Alzahrani, R.J.; Alzahrani, A. Security analysis of DDoS attacks using machine learning algorithms in networks traffic. *Electronics* **2021**, *10*, 2919. [CrossRef]
142. Santos, R.; Souza, D.; Santo, W.; Ribeiro, A.; Moreno, E. Machine learning algorithms to detect DDoS attacks in SDN. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5402. [CrossRef]
143. Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.A.; Jilani, S.F. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT. *Sensors* **2022**, *22*, 2697. [CrossRef] [PubMed]
144. Gaur, V.; Kumar, R. Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices. *Arab. J. Sci. Eng.* **2022**, *47*, 1353–1374. [CrossRef]
145. Aldaej, A.; Ahanger, T.A.; Atiquzzaman, M.; Ullah, I.; Yousufudin, M. Smart cybersecurity framework for IoT-empowered drones: Machine learning perspective. *Sensors* **2022**, *22*, 2630. [CrossRef]
146. Nishanth, N.; Mujeeb, A. Modelling and detection of flooding-based denial-of-service attack in wireless ad hoc network using Bayesian inference. *IEEE Syst. J.* **2021**, *15*, 17–26. [CrossRef]
147. Ouiazzane, S.; Addou, M.; Barramou, F. A multiagent and machine learning based denial of service intrusion detection system for drone networks. In *Geospatial Intelligence. Advances in Science, Technology & Innovation*; Barramou, F., El Briichi, E.H., Mansouri, K., Dehbi, Y., Eds.; Springer: Cham, Switzerland, 2022; pp. 51–65.
148. Musaddiq, A.; Zikria, Y.B.; Kim, S.W. Routing protocol for low-power and lossy networks for heterogeneous traffic network. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 21. [CrossRef]
149. Airehrour, D.; Gutierrez, J.; Ray, S.K. Secure routing for internet of things: A survey. *J. Netw. Comput. Appl.* **2016**, *66*, 198–213. [CrossRef]
150. Mayzaud, A.; Badonnel, R.; Chrisment, I. A taxonomy of attacks in RPL-based internet of things. *Int. J. Netw. Secur.* **2016**, *18*, 459–473.
151. Sharma, G.; Grover, J.; Verma, A. Performance evaluation of mobile RPL-based IoT networks under version number attack. *Comput. Commun.* **2023**, *197*, 12–22. [CrossRef]
152. Al-Amiedy, T.A.; Anbar, M.; Belaton, B.; Kabla, A.H.H.; Hasbullah, I.H.; Alashhab, Z.R. A systematic literature review on machine and deep learning approaches for detecting attacks in RPL-based 6LoWPAN of internet of things. *Sensors* **2022**, *22*, 3400. [CrossRef]
153. Mehbodniya, A.; Webber, J.L.; Shabaz, M.; Mohafez, H.; Yadav, K. Machine learning technique to detect sybil attack on IoT based sensor network. *IETE J. Res.* **2021**, *2021*, 1–9. [CrossRef]
154. Osman, M.; He, J.; Mokbal, F.M.M.; Zhu, N.; Qureshi, S. ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks. *IEEE Access* **2021**, *9*, 83654–83665. [CrossRef]
155. Sharma, S.; Verma, V.K. AIEMLA: Artificial intelligence enabled machine learning approach for routing attacks on internet of things. *J. Supercomput.* **2021**, *77*, 13757–13787. [CrossRef]
156. Verma, A.; Ranga, V. ELNIDS: Ensemble learning based network intrusion detection system for RPL based internet of things. In Proceedings of the 4th IEEE International Conference on Internet of Things: Smart Innovation and Usages, Ghaziabad, India, 18–19 April 2019; pp. 1–6.
157. Sharma, M.; Elmiligi, H.; Gebali, F.; Verma, A. Simulating attacks for RPL and generating multi-class dataset for supervised machine learning. In Proceedings of the IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, Vancouver, BC, Canada, 17–19 October 2019; pp. 20–26.
158. Karami, A.; Guerrero-Zapata, M. A hybrid multi-objective RBF-PSO method for mitigating dos attacks in named data networking. *Neurocomputing* **2015**, *151*, 1262–1282. [CrossRef]
159. Lee, R.; Leau, Y.; Park, Y.J.; Anbar, M. A survey of interest flooding attack in named-data networking: Taxonomy, performance and future research challenges. *IETE Tech. Rev.* **2022**, *39*, 1027–1045. [CrossRef]
160. Kumar, N.; Singh, A.K.; Srivastava, S. Feature selection for interest flooding attack in named data networking. *Int. J. Comput. Appl.* **2021**, *43*, 537–546. [CrossRef]
161. Zhi, T.; Liu, Y.; Wang, J.; Zhang, H. Resist interest flooding attacks via entropy–SVM and Jensen–Shannon divergence in information-centric networking. *IEEE Syst. J.* **2019**, *14*, 1776–1787. [CrossRef]
162. Yue, M.; Zheng, H.; Feng, W.; Wu, Z. A detection method for I-CIFA attack in NDN network. In Proceedings of the 6th International Conference on Smart Computing and Communication, New York, NY, USA, 29–31 December 2021; pp. 364–373.
163. Doriguzzi-Corin, R.; Millar, S.; Scott-Hayward, S.; Martinez-del-Rincon, J.; Siracusa, D. LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 876–889. [CrossRef]

164. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection system: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
165. Hasan, M.Z.; Hasan, K.Z.; Sattar, A. Burst header packet flood detection in optical burst switching network using deep learning model. *Procedia Comput. Sci.* **2018**, *143*, 970–977. [CrossRef]
166. Alzahrani, S.; Hong, L. Detection of distributed denial of service (DDoS) attacks using artificial intelligence on cloud. In Proceedings of the IEEE World Congress on Services, San Francisco, CA, USA, 2–7 July 2018; pp. 35–36.
167. Zhu, M.; Ye, K.; Xu, C.Z. Network anomaly detection and identification based on deep learning methods. In *Cloud Computing—CLOUD 2018*; Lecture Notes in Computer Science; Luo, M., Zhang, L.J., Eds.; Springer: Cham, Switzerland, 2018.
168. Priyadarshini, R.; Barik, R.K. A deep learning based intelligent framework to mitigate DDoS attack in fog environment. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 825–831. [CrossRef]
169. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the IEEE International Conference on Smart Computing, Hong Kong, China, 29–31 May 2017; pp. 1–8.
170. Shurman, M.M.; Khrais, R.M.; Yateem, A.A. DoS and DDoS attack detection using deep learning and IDS. *Int. Arab J. Inf. Technol.* **2020**, *17*, 655–661. [CrossRef] [PubMed]
171. Ge, M.; Syed, N.F.; Fu, X.; Baig, Z.; Robles-Kelly, A. Towards a deep learning-driven intrusion detection approach for Internet of things. *Comput. Netw.* **2021**, *186*, 107784. [CrossRef]
172. Elsayed, M.S.; Le-Khac, N.A.; Dev, S.; Jurcut, A.D. Ddosnet: A deep-learning model for detecting network attacks. In Proceedings of the IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks, Cork, Ireland, 31 August–3 September 2020; pp. 391–396.
173. Roopak, M.; Tian, G.Y.; Chambers, J. Deep learning models for cyber security in IoT networks. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 7–9 January 2019; pp. 0452–0457.
174. Abeshu, A.; Chilamkurti, N. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Commun. Mag.* **2018**, *56*, 169–175. [CrossRef]
175. McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In Proceedings of the IEEE International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
176. Ramadan, R.A.; Emara, A.H.; Al-Sarem, M.; Elhamahmy, M. Internet of drones intrusion detection using deep learning. *Electronics* **2021**, *10*, 2633. [CrossRef]
177. Abu Al-Haija, Q.; Al Badawi, A. High-performance intrusion detection system for networked UAVs via deep learning. *Neural Comput. Appl.* **2022**, *34*, 10885–10900. [CrossRef]
178. Alissa, K.A.; Alotaibi, S.S.; Alrayes, F.S.; Aljebreen, M.; Alazwari, S.; Alshahrani, H.; Ahmed Elfaki, M.; Othman, M.; Motwakel, A. Crystal structure optimization with deep-autoencoder-based intrusion detection for secure internet of drones environment. *Drones* **2022**, *6*, 297. [CrossRef]
179. Zhang, Z.; Zhang, Y.; Niu, J.; Guo, D. Unknown network attack detection based on open-set recognition and active learning in drone network. *Trans. Emerg. Telecommun. Technol.* **2022**, 33, e4212. [CrossRef]
180. Morales-Molina, C.D.; Hernandez-Suarez, A.; Sanchez-Perez, G.; Toscano-Medina, L.K.; Perez-Meana, H.; Olivares-Mercado, J.; Portillo-Portillo, J.; Sanchez, V.; Garcia-Villalba, L.J. A dense neural network approach for detecting clone ID attacks on the RPL protocol of the IoT. *Sensors* **2021**, *21*, 3173. [CrossRef] [PubMed]
181. Anitha, A.A.; Arockiam, L. ANNIDS: Artificial neural network-based intrusion detection system for internet of things. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 2583–2588. [CrossRef]
182. Cakir, S.; Toklu, S.; Yalcin, N. RPL attack detection and prevention in the internet of things networks using a GRU based deep learning. *IEEE Access* **2020**, *8*, 183678–183689. [CrossRef]
183. Yavuz, F.Y.; Ünal, D.; Gül, E. Deep learning for detection of routing attacks in the internet of things. *Int. J. Comput. Intell. Syst.* **2018**, *12*, 39–58. [CrossRef]
184. Zeng, Y.; Wu, G.; Wang, R.; Obaidat, M.S.; Hsiao, K.F. False-locality attack detection using CNN in named data networking. In Proceedings of the IEEE Global Communications Conference, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
185. Kumar, N.; Singh, A.K.; Srivastava, S. Evaluating machine learning algorithms for detection of interest flooding attack in named data networking. In Proceedings of the 10th International Conference on Security of Information and Networks, Jaipur, India, 13–15 October 2015; pp. 299–302.
186. MIT Lincoln Laboratory. 1998 DARPA Intrusion Detection Evaluation Dataset. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-dataset (accessed on 12 November 2022).
187. Lippmann, R.; Haines, J.W.; Fried, D.J.; Korba, J.; Das, K. The 1999 DARPA off-line intrusion detection evaluation. *Comput. Netw.* **2000**, *34*, 579–595. [CrossRef]
188. KDD CUP. Information and Computer Science University of California, Irvine U.S. *California.* 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 12 November 2022).
189. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
190. Sahingoz, O.K. A clustering approach for intrusion detection with big data processing on parallel computing platform. *Balk. J. Electr. Comput. Eng.* **2019**, *7*, 286–293. [CrossRef]
191. UNB. NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 12 November 2022).

192. Vasudevan, A.; Harshini, E.; Selvakumar, S. SSENet-2011: A network intrusion detection system dataset and its comparison with KDD CUP 99 dataset. In Proceedings of the IEEE 2nd Asian Himalayas International Conference on Internet, Kathmundu, Nepal, 4–6 November 2011; pp. 1–5.
193. Bhattacharya, S.; Selvakumar, S. Ssenet-2014 dataset: A dataset for detection of multiconnection attacks. In Proceedings of the IEEE 3rd International Conference on Eco-friendly Computing and Communication Systems, Mangalore, India, 18–21 December 2014; pp. 121–126.
194. Kent, A.D. *Comprehensive, Multi-Source Cyber-Security Events Dataset*; Los Alamos National Laboratory: Los Alamos, NM, USA, 2015. [CrossRef]
195. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
196. Canadian Institute for Cybersecurity. Datasets. Available online: http:www.unb.ca/cic/datasets/dos-dataset.html (accessed on 14 November 2022).
197. Alkasassbeh, M.; Al-Naymat, G.; Hassanat, A.B.; Almseidin, M. Detecting distributed denial of service attacks using data mining techniques. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 436–445. [CrossRef]
198. Beer, F.; Hofer, T.; Karimi, D.; Bühler, U. A new attack composition for network security. In Proceedings of the 10th DFN-Forum Kommunikationstechnologien, Berlin, Germany, 30–31 May 2017; pp. 1–8.
199. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
200. Canadian Institute for Cybersecurity. Intrusion Detection Evaluation Dataset (CIC-IDS2017). 2017. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 11 November 2022).
201. A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). Available online: https://registry.opendata.aws/cse-cic-ids2018 (accessed on 23 November 2022).
202. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the IEEE International Carnahan Conference on Security Technology, Chennai, India, 1–3 October 2019; pp. 1–8.
203. Canadian Institute for Cybersecurity. DDoS Evaluation Dataset (CIC-DDoS2019). 2019. Available online: https://www.unb.ca/cic/datasets/ddos-2019.html (accessed on 11 November 2022).
204. Ullah, I.; Mahmoud, Q.H. A technique for generating a botnet dataset for anomalous activity detection in IoT networks. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Toronto, ON, Canada, 11–14 October 2020; pp. 134–140.
205. Mbona, I.; Eloff, A. Detecting zero-day intrusion attacks using semi-supervised machine learning approaches. *IEEE Access* **2022**, *10*, 69822–69838. [CrossRef]
206. Faloutsos, M.; Faloutsos, P.; Faloutsos, C. On power-law relationships of the internet topology. *ACM SIGCOMM Comput. Commun. Rev.* **1999**, *29*, 251–262. [CrossRef]
207. Wang, S.; Chen, Y.; Tian, H. An intrusion detection algorithm based on chaos theory for selecting the detection window size. In Proceedings of the 8th IEEE International Conference on Communication Software and Networks, Beijing, China, 4–6 June 2016; pp. 556–560.
208. Ding, H.; Chen, L.; Dong, L.; Fu, Z.; Cui, X. Imbalanced data classification A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* **2022**, *131*, 240–254. [CrossRef]
209. Batchu, R.K.; Seetha, H. On improving the performance of DDoS attack detection system. *Microprocess. Microsyst.* **2022**, *93*, 104571. [CrossRef]
210. Khanam, S.; Ahmedy, I.; Idris, M.Y.I.; Jaward, M.H. Towards an effective intrusion detection model using focal loss variational autoencoder for internet of things (IoT). *Sensors* **2022**, *22*, 5822. [CrossRef]
211. Riddell, L.; Ahmed, M.; Haskell-Dowland, P. Establishment and mapping of heterogeneous anomalies in network intrusion datasets. *Connect. Sci.* **2022**, *34*, 2755–2783. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.