Check for updates

Received: 3 January 2023 | Revised: 28 April 2023 | Accepted: 4 May 2023

DOI: 10.1002/eng2.12697

REVIEW

Engineering Reports Open Access WILEY

DDoS attacks and machine-learning-based detection methods: A survey and taxonomy

Mohammad Najafimehr | **Sajjad Zarifzadeh** | **Seyedakbar Mostafavi**

Department of Computer Engineering, Yazd University, Yazd, Iran

Correspondence
Mohammad Najafimehr, Department of Computer Engineering, Yazd University, University Blvd, Safayieh, Yazd, Iran.
Email: mohammad.najafimehr@stu.yazd.ac.ir

Abstract

Distributed denial of service (DDoS) attacks represent a significant cybersecurity challenge, posing a critical risk to computer networks. Developing an effective defense mechanism against these attacks is crucial but challenging, given their diverse attack types, network and computing platform heterogeneity, and complex communication protocols. Moreover, the emergence of innovative DDoS attack methods presents a formidable threat to existing countermeasures. Various machine learning techniques have shown promise in detecting DDoS attacks with low false-positive rates and high detection rates. This survey paper offers a comprehensive taxonomy of machine learning-based methods for detecting DDoS attacks, reviewing supervised, unsupervised, hybrid approaches, and analyzing the related challenges. Further, we explore relevant datasets, highlighting their strengths and limitations, and propose future research directions to address the current gaps in this domain. This paper aims to provide a profound understanding of DDoS attack detection mechanisms, aiding researchers, and practitioners in developing effective cybersecurity approaches against such attacks. This research is essential because DDoS attacks are diverse and pose a formidable threat to computer networks, and various machine learning techniques have shown promise in detecting them. Its implications include providing insights that can inform the development of robust defense mechanisms against DDoS attacks.

KEYWORDS

DDoS attacks, DDoS detection, DDoS survey, machine learning, network security

1 | INTRODUCTION

Despite persistent efforts to prevent, detect, and mitigate Distributed Denial of Service (DDoS) attacks on computer networks, these destructive attacks remain prevalent.[1-3] As such, finding solutions to this problem continues to be a critical challenge in the field of network security. DDoS attacks involve leveraging a network of obedient devices, commonly referred to as zombies or bots. The aim of the attacker is to disrupt network infrastructure services by sending attack traffic to the target through the botnet.[4,5] This action effectively denies legitimate users access to network services.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2023 The Authors. *Engineering Reports* published by John Wiley & Sons Ltd.

Despite a wide array of countermeasures, detecting these attacks remains a challenge. This is largely due to the fact that the botnet, designed for malicious purposes, is dispersed across the internet, and its traffic is often indistinguishable from legitimate traffic.[6-8] Furthermore, bots typically present a forged IP address, making it even more challenging to detect and counter these attacks.[9-11]

DDoS attacks are a significant threat to internet services and can have devastating consequences on website and web application availability, often leading to shutdowns. The financial implications of such attacks can be dire for businesses that rely on internet-based operations. The disruption of communication channels, including access to critical emergency and financial systems, further underscores the significance of DDoS attacks. A noteworthy example of a DDoS attack occurred in February 2020, when Amazon suffered a CLDAP (connection-less lightweight directory access protocol) reflection-amplification attack, with a tremendous rate of 2.3 Tbps, making it the most extensive attack recorded to date, as reported by ZDNet.[12,13] Similarly, GitHub experienced such an attack in February 2018, which caused a temporary service disruption.[14] In yet another instance, Dyn's managed DNS infrastructure was targeted in 2016 in an attack that lasted approximately 3 h and affected several high-profile web services, including Twitter and PayPal, causing severe disruption.[15,16] Overall, DDoS attacks pose a significant challenge and require a comprehensive approach to mitigate and manage the associated risks.

Numerous techniques exist to prevent, detect, and mitigate DDoS attacks. In terms of detection methods, there are two primary approaches: signature-based and anomaly detection.[17] Signature-based methods can only detect known attacks for which the signature is already known, and are not effective against novel or zero-day attacks.[18] On the other hand, the anomaly detection approach can detect new and unknown attacks by identifying anomalous circumstances caused by the attack.[19] Statistical methods such as entropy analysis[20] and machine learning (ML) methods[21,22] are typically utilized in the anomaly detection approach.

The categorization of DDoS detection methods based on network topology entails three distinct groups–the source, destination, and network-based methods.[11,23] Source-based methods locate and operate from the attack's point of origination close to the attacker, while destination-based methods are implemented within the attack's destination network in proximity to the target. On the other hand, network-based methods function within the Internet infrastructure, positioned between the attacker and the victim.

Currently, a significant gap in research on countering DDoS attacks exists whereby although defense mechanisms are increasingly effective, attack methods have become increasingly sophisticated. Consequently, novel forms of DDoS attacks could arise, which existing detection methods may not be able to mitigate effectively.[24] For instance, Cambiasoa et al.[25] introduced the SlowDrop attack in 2019, where the attack imitates the behavior of a legitimate user with a weak and unreliable connection to the server. Another example is the Portmap DDoS attack;[26] a reflection and amplification DDoS attack was initially detected in 2015 and targeted the Lumen company.[27] A vital research question is the effectiveness of ML-based detection methods for real-world DDoS attacks. Although these methods are significantly accurate in simulated testbeds and prepared datasets, Bakker et al.[28] indicate that the discrepancy between the lab testbed conditions and real-world circumstances could hinder their efficacy.

Despite the numerous studies conducted on DDoS attacks and their detection methods, there are still several limitations in the existing literature:

1. Many studies have focused on detecting DDoS attacks in specific fields, which constrains their scope and effectiveness.
2. Some studies have overlooked the importance of introducing relevant datasets and their pertinent features that can be used for cross-comparison of detection methods.
3. with the increase in the popularity of ML methods in the DDoS detection field, some studies have not focused on this modern approach.
4. There is a lack of a systematic classification of ML-based detection methods, which hinders researchers' ability to compare and evaluate different approaches.
5. Some studies have failed to illustrate the most common types of DDoS attacks, which makes it difficult for readers to understand the methods employed by attackers in these attacks.

To address these limitations, this paper aims to comprehensively explore DDoS attacks and detection methods, with a particular focus on ML-based approaches. We provide a detailed taxonomy of such methods, which will enable researchers to systematically classify and evaluate different approaches. Additionally, we introduce significant datasets and their key characteristics, which will facilitate the cross-comparison of detection methods. We also depict the most prevalent types of DDoS attacks, which will help readers understand the methods employed by attackers in these attacks. Our study's

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

theoretical contribution is significant because it summarizes ML-based DDoS detection methods in a single paper, which will assist researchers in grasping the current state of the field. We also provide tables for an effective comparison of the results obtained from different ML methods utilized in DDoS detection. By discussing the proposed methods' shortcomings and the ongoing challenges in DDoS detection, we help researchers understand the limitations of the existing approaches. Finally, we offer various suggestions for conducting further research in this area to address the gaps and limitations found in the existing literature on DDoS detection.

This paper is organized as follows. Section 2 provides a comprehensive review of DDoS attacks, encompassing their diverse variations and categories. Section 3 compares the current survey against previous surveys. In Section 4, the classification of machine learning-based methods employed in detecting DDoS attacks is outlined along with recently suggested techniques. To conclude the paper, Section 6 offers its final remarks.

2 | DDOS ATTACKS: CONCEPTS AND CATEGORIES

Various techniques and methods have been employed by attackers to execute DDoS attacks. As detection and mitigation methods have progressed, new forms of attacks have arisen. There are different ways to categorize these attacks, such as the rate of attacks and their mechanisms. With regards to the attack's rate, low-rate, and high-rate attacks can be considered, which are described as follows.

Low-rate attacks: A low-rate DDoS attack involves sending malicious traffic at a slow pace to the target. This attack exploits the vulnerability of TCP's congestion control mechanism. The malicious traffic is sent repeatedly over short periods as in a "pulsing attack", or at a steady, low rate termed a "constant attack".[6] A DDoS attack is considered low-rate if its rate is below 1000 bps or it accounts for 10% to 20% of the target's background network traffic.[29] As an example of the low-rate attack, Pascoal et al.[30] in 2020, proposed a novel type of low-rate DDoS, namely slow ternary content-addressable memory (slow-TCAM) attack, and demonstrated that it is disruptive even with the rate of four packets per second compared to 1000 packets per second rate of existing similar attacks. This type of attack works by sending distinctive packets to software defined network (SDN) switches and results in exhaustion of the switches' memory by generating new fake entries in the flow table of the switches.[31] In comparison to traditional volumetric attacks, low-rate attacks have a relatively minimal impact on bandwidth consumption, resulting in a reduced average number of attack packets. This, in turn, renders them challenging to detect, given that their generated traffic is challenging to distinguish from legitimate traffic. Low-rate attacks frequently target thread-based web servers through the slow transmission of requests.[32] As a result, the attack rate remains low, but every thread is tied up and cannot fulfill legitimate requests. This is achieved by transmitting data slowly but still quickly enough to avoid the server from timing out on the established connection.

High-rate attacks: Conversely, high-rate attacks involve a voluminous quantity of packets sent by the attacker to the victim to compromise its service availability. These attacks are often referred to as volumetric or flooding attacks due to the sheer volume of malicious traffic generated, which includes SYN flood,[33] HTTP flood,[11,34] UDP flood,[35] and ICMP flood.[36] Generally, high-rate DDoS attacks can be accomplished through two distinct methods:[37] directly or through reflectors. In direct attacks, an attacker typically employs a botnet to launch the attack. In contrast, reflection attacks involve the use of botnets, along with other devices (called reflectors), to target victims. These attacks will be subsequently discussed in this section.

Protocol exploitation attacks: This category of attack involves exploiting network protocol vulnerabilities to exhaust a server's resources.[23] There are some examples of these types of attacks:

1. *SYN flood*: The TCP-SYN flood attack, commonly referred to as SYN flood, is a well-known type of DDoS attack.[38] The attack is perpetrated against a host that uses a service over the TCP. TCP is a transfer-layer protocol used by various application-layer services such as HTTP, FTP, SMTP, Telnet, SSH, and IMAP, making them vulnerable to this attack. The attacker sends numerous TCP-SYN requests with the SYN flag set but does not respond to the ACK response to complete the TCP three-way handshake. This results in a large number of half-open TCP connections on the target, leading to resource exhaustion and prevention of opening new TCP connections, thereby rendering the target inaccessible. The SYN flood attack was first documented in 1996.[39]
2. *UDP flood*: Another DDoS attack is the UDP flood attack, a volumetric attack whereby the attacker sends too many UDP datagrams to different ports on the server . If no application is listening to the specific port at the intended destination, the server returns an ICMP packet to the sender informing it that the destination is unreachable.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The volumetric nature of this attack causes server responses to degrade or become unresponsive, effectively rendering the targeted service unavailable.

3. *Lag switch cheating*: The concept of "Lag switch cheating" is a form of exploitation that is commonly seen in online gaming environments. Specifically, it involves an attacker who intentionally creates a temporary delay or lag in their game routine to gain an unfair advantage. This deceitful act permits the attacker to continue playing while others are oblivious, and the attacker suddenly materializes in a different position within the virtual ecosystem.[40] To carry out this attack, the attacker connects the game console to a specially designed network switch called a "lag switch" that has a button used to activate the lag for a few seconds. The function of the switch is to buffer the packets that travel through it while the attacker plays the game on their console. Since online gaming usually leverages the UDP transfer protocol, this tactic is frequently referred to as "UDP-lag." Although this malicious activity poses a risk to the integrity of online games, the execution of the attack closely resembles that of slower denial-of-service attacks.

Reflection-amplification attacks: Reflection-based DDoS attacks involve malicious traffic where the attacker's source IP is substituted with the victim's IP. The malicious packets are directed towards other nodes in the network which then send their responses to the victim. This allows the attacker to remain anonymous.[41] In addition, there is another concept called "amplification",[42] where short-length requests generate longer responses. Reflection attacks exploit this amplification, thus they are called reflection-amplification DDoS attacks. Figure 1 illustrates the process of the attack, in which reflectors direct heavier traffic to the victim compared to the traffic sent from the attacker to the reflector.

Due to this exploitation, reflectors are also referred to as amplifiers. Several types of reflection-amplification DDoS attacks are described here based on the protocol they exploit, including:

1. *MSSQL*: The Microsoft SQL Server (MSSQL) can be exploited by attackers to launch DDoS attacks, by misusing the Microsoft SQL Server Resolution Protocol (MC-SQLR). Clients use the MC-SQLR protocol to identify database instances in a cluster of servers on a network. During an attack, a client sends a request to a server, which responds with a list of current database instances. Attackers forge the IP address of the target and send requests to a large number of servers on the internet, causing the bombardment of the target with MC-SQLR responses.[43]
2. *SSDP*: Simple service discovery protocol (SSDP) attack is a malicious network activity that exploits the universal plug and play (UPnP) networking protocols. The aim of the attacker is to flood the victim's infrastructure with a high-volume of amplified traffic, leading to the saturation of resources and ultimately causing inaccessibility of those resources.[44,45] In a typical network setup, the SSDP is utilized to enable UPnP devices to advertise their presence and services to other devices on the network. For example, when a UPnP printer connected to the network obtains an IP address, it can send a notification message to the SSDP service provider, which in turn multicasts a message to all the devices on the network about the new printer. When a computer on the network receives the discovery message, it sends a request message to the printer, requesting a complete description of its services. The printer then sends a response message containing a comprehensive list of provided services to the requesting device. However, attackers can exploit this final request for services to

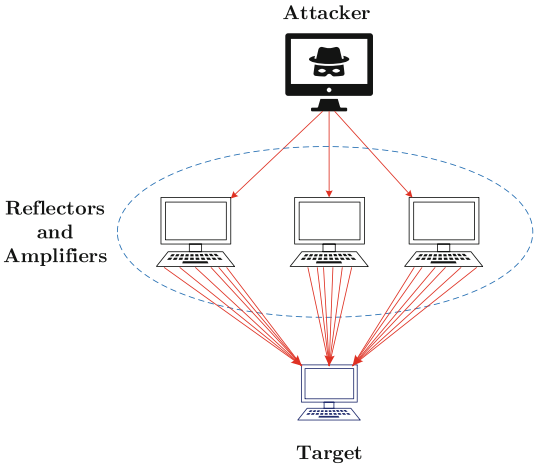

FIGURE 1 A reflection-amplification Distributed denial of service (DDoS) attack diagram.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

(a) An initial scan is performed by the attacker to search for plug-and-play devices to use for amplification purposes.
(b) Once acquainted with the available devices on the network, a list of responding devices is generated.
(c) A UDP packet is then generated with a forged IP address belonging to the intended victim.
(d) Through the use of its botnet and certain flags, such as *ssdp:rootdevice* and *ssdp:all*, the attacker sends a forged discovery packet towards all UPnP devices, requesting as much data as possible.
(e) As a result, each device will respond to the targeted victim with a much larger message than the one sent by the attacker.
(f) The victim receives a substantial volume of traffic from all devices, potentially overwhelming it and rendering it unable to respond to legitimate users.

3. *CharGEN*: The Character GENerator Protocol (CharGEN)[46] is a network protocol that has been specifically designed for testing, debugging, and measurement purposes. It enables clients to connect to a server offering the CharGEN Protocol on either TCP or UDP port 19. Once a TCP connection is established, the server will begin sending random characters to the client until the connection is terminated. In the case of the UDP implementation, the server will send a UDP datagram consisting of a random number of characters following the receipt of each connecting client datagram. However, despite its intended use, the CharGEN protocol is susceptible to abuse. CharGEN attacks, including CharGEN amplification attacks, can be carried out by sending small, spoofed IP packets to network devices that have the CharGEN service enabled. These fake requests are then utilized to send UDP flood packets to the victim as responses from these devices. Since this protocol is enabled by default on most Internet-enabled devices, such as printers or copiers, it can be leveraged to launch a CharGEN attack. This type of attack overwhelms the victim's port 19 with a large number of UDP packets. Consequently, the victim's resources are quickly depleted, and it can no longer respond to requests as usual.
4. *LDAP*: Lightweight directory access protocol (LDAP) is a client-server protocol used to access and manage directory information. LDAP servers provide directory service by storing information about network resources such as users, groups, and devices. However, LDAP can also be exploited by attackers to carry out DDoS attacks. In such attacks, they send a large number of LDAP requests with a forged IP address of the target as the source IP address. Consequently, the servers respond to the target with massive traffic, overwhelming it.[47]
5. *NetBIOS*: Network basic input/output system (NetBIOS) is a protocol that enables computers on a local network to share files and resources.[48] Computers running Microsoft Windows with the same workgroup can communicate with each other using NetBIOS, and each computer is identified by its NetBIOS name. The NetBIOS name server (NBNS) is used to map NetBIOS names to network addresses. However, like other reflection-amplification attacks, NetBIOS can be used by attackers to launch DDoS attacks. By broadcasting a large number of queries to NBNS reflectors over the Internet, an attacker can force them to respond with massive traffic to the target.

The impact of the reflection-amplification attacks is measured by bandwidth amplification factor (BAF),[49] which is described as follows:

$$BAF = \frac{len(S)}{len(R)} \tag{1}$$

in which, $len(S)$ and $len(R)$ are the length of the packets (in Bytes) sent from the amplifier to the target and from the attacker to the amplifier, respectively. The greater the BAF value, the more the attack is amplified. Table 1 shows the BAF for the various prevalent reflection-amplification attacks.[50]

TABLE 1 Multiple reflection-amplification attacks and their BAF.

Protocol	BAF
SSDP	30.8
LDAP	46 to 55
CharGEN	358.8
NetBIOS	3.8
NTP	556.9
Portmap	28 to 57

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 The comparison of this survey and the related review papers.

Paper	Date	Area	Dataset list	Method taxonomy	Attack types
Alrehan and Alhaidari[51]	2018 and older	Only VANET	No	No	No
Loukas and Gülay[52]	2009 and older	SDN and IoT not covered	No	No	Partially
Bhardwaj et al.[53]	2015 and older	SDN and IoT not covered	No	Partially	Partially
Kamboj et al.[54]	2015 and older	SDN and IoT not covered	No	No	Partially
This paper	2023 and older	Web servers, IoT, SDN, Cloud	Yes	Yes	Partially

3 | RELATED WORDK

The preventions, detections, and mitigations against DDoS attacks have been subjects of numerous surveys, ranging from comprehensive narrative papers that cover different attack types, challenges, and countermeasure methods to papers that focus on specific aspects of the phenomenon. In this section, we provide a literature review of such papers, while appraising and contrasting their contributions with those of the present survey paper. To aid in this evaluation, we provide Table 2 that highlights the distinctive features of our study in relation to the previously noted surveys.

Alrehan and Alhaidari[51] reviewed the various ML-based methods for detecting DDoS attacks in the setting of Vehicular Ad-hoc Networks (VANET), which represents a subset of the Internet of Things (IoT) phenomenon. The authors delved into the weaknesses and vulnerabilities of VANET, presenting an analysis of the DDoS attack approaches and proposed detection methods that were documented up until 2018. However, a comprehensive taxonomy of detection methods and an extensive list of datasets were not provided, although some datasets were briefly introduced.

Loukas and Gülay[52] conducted a review of the prevention, detection, and mitigation strategies employed against DDoS attacks. The authors presented historical background information on DDoS attacks and surveyed a range of the methods proposed to combat them until 2009. However, a taxonomy of the discussed techniques was not established, and there was no list of the datasets available for use with ML algorithms. Additionally, their survey did not examine evaluated methods in the context of SDN or IoT environments.

Bhardwaj et al.[53] focused their survey on DDoS attacks that occur in cloud computing environments and the associated detection strategies. The reviewed papers were published up until 2015, and the study proposed a classification system for the attacks. Although a taxonomy for detection methods was proposed, it was not entirely comprehensive and did not focus specifically on ML approaches, thus leaving out detection strategies that can be employed in SDN and IoT environments.

Kamboj et al.[54] conducted a general overview of the structure, functionality, and existing solutions for DDoS attacks. The authors provided a survey of the detection methods available until 2015, but there was no established taxonomy of these methods or detailed descriptions of the techniques.

This paper aims to improve on the shortcomings of mentioned surveys. In particular, we provide a more extensive and up-to-date literature review and a comprehensive taxonomy of detection methods, including some of those that can be also employed in emerging SDN and IoT networks. Additionally, we offer a broader set of surveyed techniques for detecting DDoS attacks and a detailed description of the techniques. Our paper also introduce a set of datasets for training and testing machine learning algorithms, which were not provided in some of the previous surveys.

4 | ML-BASED DDOS DETECTION METHODS

Utilizing machine learning as an anomaly detection mechanism to differentiate between benign and attack traffic is a contemporary research topic that presents promising results. One approach involves utilizing a physical network as a testbed, wherein both the attacking and victim machines are present, and multiple attacks are conducted in a controlled manner. The resulting traffic logs can be used to train supervised learning algorithms to distinguish between attack and benign traffic. Alternatively, unsupervised learning algorithms can be used to cluster incoming traffic in real-time, separating normal traffic from the attack based on their behavioral and feature characteristics. In both approaches, the traffic packets or flows are represented using key features such as packet size, protocol, and interval between packets.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Machine Learning (ML)-based DDoS detection methods can be categorized into three primary groups, namely supervised, unsupervised, and hybrid, each with multiple subcategories. A comprehensive taxonomy of ML-based DDoS detection methods is presented in Figure 2. In the ensuing section, this paper will expound on primary concepts and notations and discuss each of the mentioned categories of ML-based DDoS detection methods, including recent research endeavors. Additionally, Table 3 tabulates a summary of all the proposed ML-based DDoS detection approaches reviewed.

To conduct a comprehensive comparison of the existing detection methods, we have synthesized the best proposed ML methods based on their reported evaluation metrics in Table 4. Moreover, we have also discussed the inadequacies of each method in Table 5. Overall, our analysis suggests that random forest (RF) and support vector machine (SVM) among the ML models, and convolutional neural network (CNN) and long short-term memory (LSTM) models among the deep learning methods, are more efficient in detecting attacks. Nonetheless, other methods have also shown promising results in specific cases. Moving ahead, several areas warrant investigation with respect to the shortcomings of current methods, including:

1. In a majority of instances, various evaluation metrics remain unreported. As an example, it is posited that recall and FPR are of greater significance than accuracy and precision when it comes to detecting DDoS attacks. The reason being that it is imperative for the model to detect as many attack flows as possible, which outweighs the importance of detecting normal flows. Additionally, categorizing benign traffic as malicious–that is, falsely detecting attack flows, can be as detrimental as failing to detect them.
2. In typical machine learning practice, the model evaluation is conducted on a distinct subset of the same dataset used in training, albeit employing different instances. It is our belief that the model assessment must be performed using a completely new dataset to ensure reliable evaluation. This approach is necessary because in real-world scenarios, various factors, such as botnets deployed by attackers, attack vectors and types, as well as network parameters, including noise

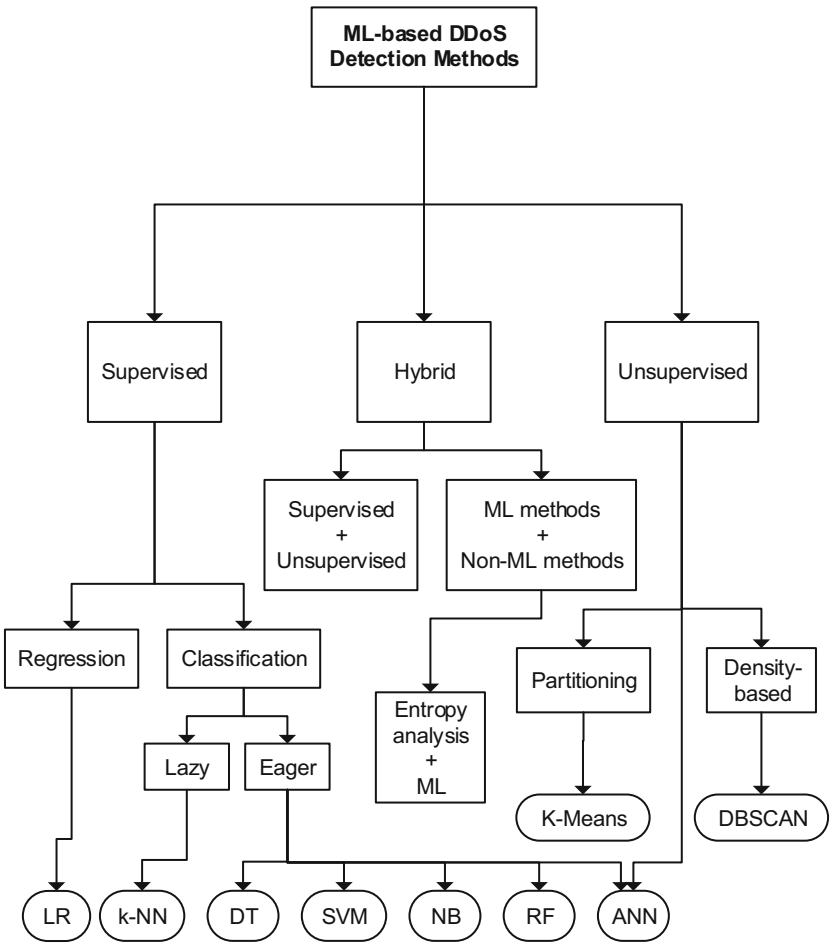

FIGURE 2 Taxonomy of machine learning (ML)-based distributed denial of service (DDoS) detection methods.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 Machine learning (ML)-based distributed denial of service (DDoS) detection methods classification.

Ref.	Major cat.	Minor cat.	Learning algorithm	Testbed	Year
55	Sup.	Eager	NB, DT	CAIDA backscatter dataset	2014
56	Uns.	Partitioning	K-Means	DNS AMPL DDOS DEC2015 dataset	2017
57	Uns.	Density-based	DBSCAN	DARPA 2000 and CAIDA datasets	2017
58	Uns.	Density-based	DBSCAN	Physical network	2018
59	Sup.	Eager, Lazy	NB, DT, k-NN, RF, ANN, SVM	Physical IoT	2018
28	Sup.	Eager, Lazy	SVM, k-NN, RF	Physical SDN, ICSX dataset	2018
60	Sup.	Eager	NB, LR, DT	Physical network	2018
61	Sup.	Eager	RF, AdaBoost, DT, SVM	Physical network, CICIDS2017 dataset	2018
62	Hyb.	Sup. + Uns. + Entropy analysis	Extra trees, co-clustering	NSL-KDD, ISCXIDS2012, UNSW-NB15 datasets	2018
63	Hyb.	Sup. + Uns.	SVM, SOM	Simulated SDN	2018
64	Sup.	Eager	NB	Simulated SDN, NSL-KDD dataset	2018
65	Sup.	Eager, Lazy	ANN, RF, SVM, DT, k-NN	CICIDS2017, KDDCup99 datasets	2019
66	Sup.	Eager, Lazy	k-NN, SVM, ANN	KDDCup99, NSL-KDDdatasets	2019
67	Sup.	Eager	NB, SVM, RF	Physical cloud	2019
68	Sup.	Eager, Lazy	DT, RF, SVM, k-NN	Simulated SDN	2019
69	Sup.	Eager	ANN, SVM, RF, Bayes	CICIDS2017 dataset	2019
70	Sup.	Eager	NB, DT, RF, ANN	a dataset[71]	2020
72	Sup.	Eager	DT, RF, SVM, GBT, LR	a public dataset[73]	2020
74	Hyb.	ML + Entropy analysis	LSTM	1999 DARPA, 2000 DARPA, CICDDoS2019, simulated SDN	2020
75	Sup.	Eager	Boosting-based LMT	Physical IoT testbed	2021
76	Sup.	Eager	ANN	KDDCup99	2021
77	Hyb.	Sup. + Uns.	DBSCAN, DT, RF, NB, SVM	CICIDS2017, CICDDoS2019	2022
78	Sup.	Eager	CNN, LSTM	CICDDoS2019	2022
79	Hyb.	ML + Entropy analysis	Alternating DT, Simple LR, MLP	MTS-IoT dataset	2023
80	Sup.	Eager	RF, SVM, LR, DT, MLP	IoT-CIDDS dataset	2023
81	Sup.	Eager	WFL	CAIDA dataset	2023

and bandwidth during an attack, may deviate significantly from the dataset employed during the model training. As a result, these factors present unknown variables for the model that must be accounted for in the development phase to enhance the model's generalizability. Therefore, the focus must be on developing a model that overcomes these unknown factors in real-world situations.

3. As various environmental factors, including jitter, bandwidth, noise and traffic load, may negatively impact the detection capabilities, it is advisable to evaluate the detection methods in an IDS deployed in a real-time network facing the attack. This approach can provide a more accurate assessment of the detection capability that accounts for the factors mentioned above. Accordingly, the validity of the detection method evaluation can be improved through replicating real-world conditions in the IDS test environment.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 The ability of machine learning (ML)-based methods in detecting distributed denial of service (DDoS) attacks.

Ref.	Best method	Acc.	Rec.	Prec.	F_1-score	FPR	AUC
55	DT	0.99	1	0.96	0.97	0.01	N/A
56	K-means	N/A	N/A	N/A	N/A	N/A	N/A
57	DBSCAN	0.98	0.52	0.41	0.45	0.0068	N/A
58	DBSCAN	N/A	N/A	N/A	N/A	N/A	N/A
59	RF	0.999	0.999	0.999	0.999	N/A	N/A
28	SVM	0.93	0.14	N/A	N/A	0.0020	N/A
60	DT	N/A	0.98	N/A	N/A	0.0026	N/A
61	RF.	0.986	N/A	N/A	N/A	0.024	N/A
62	Proposed method	0.86	≈ 0.16	≈0.30	≈0.20	0.36	N/A
63	Proposed method	0.98	0.97	N/A	N/A	0.027	N/A
64	NB	N/A	0.77	0.81	0.78	N/A	N/A
65	Multi-channel CNN	0.98	N/A	N/A	N/A	N/A	N/A
66	KNN	0.99	N/A	N/A	N/A	N/A	N/A
67	SVM	0.997	0.998	0.998	0.998	0.002	N/A
68	SVM	1	1	1	1	0	N/A
69	CNN+LSTM	0.97	0.99	0.97	0.97	N/A	N/A
70	DT	0.98	0.87	0.99	0.92	0.0002	N/A
72	RF	1	N/A	N/A	N/A	N/A	N/A
74	proposed method	N/A	1	N/A	N/A	0.03	N/A
75	LMT	0.99	0.99	0.99	0.99	0.001	N/A
76	ANN	0.99	N/A	N/A	N/A	N/A	N/A
77	RF-based proposed method	0.14	0.14	0.99	0.24	0.04	N/A
78	CNN+LSTM	0.99	0.99	0.99	0.99	N/A	0.99625
79	proposed method	N/A	N/A	N/A	0.963	N/A	0.939
80	RF	0.98	0.98	0.96	0.96	0.01	0.9806
81	WFL	0.98	0.99	0.99	0.99	0.0221	N/A

4. In some instances, a method may be proposed for a specific environment, yet a general dataset is utilized to train and/or test the model. As an example, a DDoS detection method may be proposed for IoT networks, but a dataset collected from non-IoT devices is used. This practice may result in model overfitting or reduced performance, as the proposed method may not be optimized for the unique characteristics of the specific environment. Thus, in order to ensure efficient and accurate detection methods, it is advisable to procure datasets that are tailored to the particular environment, and the method should be trained and tested on these datasets.

4.1 | Background and basic concepts

This section introduces and depicts a background, some basic concepts, and notations used in the next sections. In the attack detection procedure, regarding the result of the algorithm, there are some notations described as follows:

1. **True positives (TP):** Number of samples (packets or flows) correctly recognized as DDoS.
2. **True negatives (TN):** Number of samples (packets or flows) correctly recognized as benign.
3. **False positives (FP):** Number of samples (packets or flows) incorrectly recognized as DDoS.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 The shortcomings of the machine learning (ML)-based methods in detecting distributed denial of service (DDoS) attacks.

Ref.	Shortcomings
55	Obsolete dataset
56	Lack of reported numeric and comparative evaluation metrics
57	Obsolete dataset, detected only about half of the attack traffic
58	Not reported many essential evaluation metrics, not covered modern types of the attack
59	Not covered modern types of the attack
28	Lack of a solution for the real network environment
60	Not reported all essential evaluation metrics, not covered modern types of the attack
61	Not reported all essential evaluation metrics, not covered modern types of the attack
62	Not covered modern types of the attack, lower detection rate for attack than normal
63	Not reported all essential evaluation metrics, not specified the attack type
64	Obsolete dataset, not reported all essential evaluation metrics
65	Not reported all essential evaluation metrics, not covered modern types of the attack
66	Obsolete dataset, not reported all essential evaluation metrics
67	Not specified the attack type, not reported the size of the dataset
68	Not covered modern types of the attack, not clarified the reason for the ideal evaluation results achieved
69	Not evaluated on an IoT-specific dataset, not covered modern types of the attack
70	Low size of the dataset than common, not evaluated on a different dataset
72	Not reported many essential evaluation metrics, not clarified the reason for the ideal evaluation results achieved
74	Not reported all essential evaluation metrics, lack of comparison with existing approaches
75	Not covered modern types of the attack, lack of comparison with existing approaches
76	Obsolete dataset, not reported many essential evaluation metrics
77	Roughly weaker results than ideal, not evaluated on a real testbed or emulation
78	Not evaluated on an IoT-specific dataset, omited many attack types and samples
79	Not reported many essential evaluation metrics, not evaluated on a real testbed or emulation
80	Not evaluated on a real testbed or emulation
81	Out-of-date dataset, not evaluated on a real testbed or emulation

4. **False negatives (FN):** Number of samples (packets or flows) incorrectly recognized as benign.
5. **Precision:** This measure indicates the proportion of samples (packets or flows) correctly recognized by the model as DDoS.

$$precision = \frac{TP}{TP + FP}$$

6. **Detection rate (recall, sensitivity, true positive rate):** This measure indicates the proportion of actual DDoS samples (packets or flows) that the model successfully recognizes as DDoS.

$$detection\ rate = \frac{TP}{TP + FN}$$

7. **Specificity (true negative rate):** This measure indicates the proportion of actual benign samples (packets or flows) that the model successfully recognizes as Benign.

$$specificity = \frac{TN}{TN + FP}$$

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

8. **False negative rate (FNR):** This measure indicates the proportion of actual DDoS samples (packets or flows) that the model falsely recognizes as benign.

$$FNR = 1 - detection\ rate = \frac{FN}{FN + TP}$$

9. **False positive rate (FPR):** This measure indicates the proportion of actual benign samples (packets or flows) that the model falsely recognizes as DDoS.

$$FPR = 1 - specificity = \frac{FP}{FP + TN}$$

In addition, a glossary of abbreviation and terms related to DDoS attacks and the ML-based detection methods are provided in Table 6.

4.2 | Significance of machine learning techniques in DDoS attack detection

The use of ML methods in cybersecurity has gained significant attention due to their potential to enable effective decision-making and efficient automatic operation.[82] ML techniques have been successfully utilized in various cybersecurity domains, including spam detection, malware identification, user authentication, software vulnerability detection, and DDoS attack detection.[83] These techniques have demonstrated promising results, achieving high accuracy and recall while maintaining low false positive rates.

For instance, Banitalebi Dehkordi et al. proposed a classification algorithm to calculate entropy for SDN that achieved about 99% accuracy and a low false positive rate of 0.1.[84] Another approach proposed by Pande et al. utilized the RF classifier and reported a recall and precision of approximately 99% and a false positive rate of 0.002, as tested on the NSL-KDD dataset.[85] Almiani et al. proposed a deep neural network for detecting DDoS attacks in fifth-generation Internet of Things (5G IoT) networks, which achieved a 91% recall and a low false alarm rate of 0.09.[86] Othman et al. evaluated the SVM classifier using Apache Spark for big data analysis, which demonstrated an Area Under the Receiver Operating Characteristic curve (AUROC) of approximately 99% and an Area Under the Precision-Recall curve (AUPR) of around 96%.[87]

4.3 | Supervised methods

Supervised learning in the context of DDoS attack detection involves utilizing an algorithm to learn the function $f(x) = y$, which maps the input variable x to the output variable y. Learning the mapping function is achieved based on a dataset that contains significant information about network traffic, enabling the development of predictive models that can distinguish between normal and malicious traffic. In essence, a model is trained using the dataset, and when presented with new input data x, it calculates the corresponding output value y.

Supervised learning algorithms are typically categorized into two main categories: classification and regression. The output variable of classification models is discrete, while in the regression models, it is continuous. The dataset used in supervised learning is sourced from network traffic logs and captured traffic, with each row of the dataset representing a flow or packet and each column representing a feature. While both packet-based and flow-based datasets are available, the latter may be preferred for DDoS detection due to its ability to capture a large number of packets that belong to the same flow. This approach enhances the efficiency of the model by consolidating packets of the same flow as one entity in the dataset and considering features such as "number of packets" and "time interval between each packet" to preserve relevant information.

There is a single column in the dataset, known as the "class label," which denotes the class to which each row belongs. The term "class" refers to the type of flow or packet and is typically categorized into two broad classes: normal or benign and attack, resulting in a binary classification scheme. However, it is possible to include different types of attacks within the class label, such as [Normal, TCP-flood, CharGEN, …].

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 6 Glossary of terms for distributed denial of service (DDoS) attacks and machine learning-based detection methods.

Term	Description
ANN	Artificial neural network; a type of machine learning algorithm that is inspired by the structure and function of the human brain
AUC	Area under the ROC curve; the area under the ROC curve
BAF	Bandwidth amplification factor; a measure of the effectiveness of a DDoS attack in terms of the ratio of the attack traffic to the actual traffic sent by the attacker
Botnet	A network of compromised devices controlled by an attacker to launch DDoS attacks
CNN	Convolutional neural network; a type of deep learning architecture that is commonly used for image recognition tasks, but can also be applied to DDoS detection by treating network traffic data as a 2D image
DBSCAN	Density-based spatial clustering of applications with noise; a type of clustering algorithm that works by grouping together similar data points based on their proximity to each other
DDoS	Distributed denial of service; a type of attack that aims to disrupt the availability of a network or service
Deep learning	A type of machine learning that involves training neural networks with multiple layers to make predictions based on complex patterns in data
Decision tree	A type of machine learning algorithm that works by creating a hierarchical model of rules based on the features of the data
Feature extraction	The process of selecting relevant features from network traffic data for machine learning
FPR	False positive rate
GBT	Gradient boosting tree; a type of ensemble learning algorithm that combines multiple weak learners to create a more accurate and robust model
IIoT	Industrial internet of things; a subset of IoT that focuses on the integration of sensors, software, and connectivity in industrial and manufacturing settings
IoT	Internet of things; a network of physical devices, vehicles, home appliances, and other objects that are embedded with sensors, software, and connectivity to exchange data and interact with each other
k-NN	k-nearest neighbors; a type of machine learning algorithm that identifies the k-nearest data points in the feature space and assigns a class to a new data point based on the most common class among its k-nearest neighbors
LDAP	Lightweight directory access protocol; an application protocol for accessing and maintaining distributed directory information services over an IP network
LMT	Logistic model trees; a hybrid machine learning algorithm that combines decision trees and logistic regression to create interpretable models for binary classification tasks
LOIC	Low orbit ion cannon; a type of DDoS attack tool that can be used to flood a targeted network or service with a large volume of traffic from multiple sources
LR+	Likelihood ratio positive; the ratio of the true positive rate to the false positive rate
LSTM	Long short-term memory; a type of neural network architecture used for sequence prediction tasks
ML	Machine learning; a type of artificial intelligence that involves training a model to make predictions based on patterns and relationships in data
MLP	Multi-layer perceptron; a type of neural network architecture consisting of multiple layers of interconnected nodes
NB	Naïve Bayes; a type of machine learning algorithm based on Bayes' theorem that assumes the features are conditionally independent given the class
NIDS	Network intrusion detection system; a system that monitors network traffic for signs of unauthorized access, malware, or other malicious activity
Normalization	The process of scaling the values of features in network traffic data to a common range, typically between 0 and 1
PCA	Principal component analysis; a technique for reducing the dimensionality of data by identifying the most important components that capture the majority of the variation in the data

(Continues)

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 6 (Continued)

Term	Description
Precision	The fraction of true positive predictions among all positive predictions
RNN	Recurrent neural network; a type of neural network architecture that is designed to process sequences of data
ROC	Receiver operating characteristic; a graphical plot of the true positive rate against the false positive rate for a binary classifier as the discrimination threshold is varied
SVM	Support vector machine; a type of machine learning algorithm that works by finding the hyperplane that maximally separates the data into different classes
TFT	Time-frequency transform; a method for analyzing the spectral content of a signal as it changes over time
TPOT	Tree-based pipeline optimization tool; an open-source software package for automating the machine learning pipeline
True positive rate	The fraction of true positive predictions among all positive instances in the dataset
UDP	User datagram protocol; a transport layer protocol that is used to send datagrams over an IP network without establishing a dedicated connection
VPN	Virtual private network; a technology that allows users to securely access a private network over a public network, such as the Internet

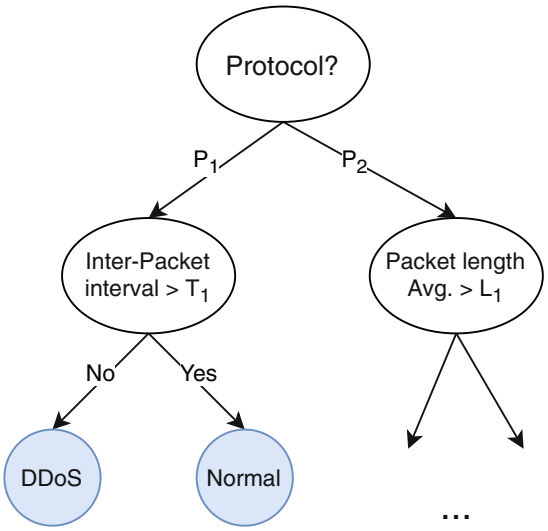

FIGURE 3 A sample of decision tree (DT) functionality for distributed denial of service (DDoS) detection.

4.3.1 | Eager algorithms

The current discourse expounds on the process of constructing a classification model in eager learners for DDoS detection. Initially, the model is trained with a training dataset to classify subsequently received data. The construction of the classification model in eager learners is exclusively based on the given training data. This feature enables the learning process to commence without having to wait for the arrival of new data that requires prediction. Notably, the learners require prolonged time to construct the model and brief time to predict.

Decision tree (DT): One of the most commonly used eager classification algorithms for DDoS detection is the Decision Tree, which is a supervised learning hierarchical model. In this model, every node is a decision node, and it implements a test function $f_m(x)$ with discrete outputs that generates individual branches in the tree. The DT algorithm requires a recently received input (packet or flow) to traverse the nodes before eventually being assigned a label in a leaf. Figure 3 provides a schematic illustration of the workings of the DT algorithm, which can be utilized for DDoS detection, although the figure only serves as a visual aid for understanding its functionality and does not exhibit a practical decision-making process for DDoS detection.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Naive Bayes (NB): The Naive Bayes (NB) classifier is an eager classification algorithm that predicts the probability of a given sample's membership in any existing classes using Bayes' theorem:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

where $P(C_i|X)$ is the posterior probability and represents the probability that the sample X belongs to class C_i, such as the probability of the flow being DDoS. $P(C_i)$ is the prior probability indicating the probability of class C_i occurring in the given dataset, which is calculated by dividing the number of class C_i's records by all the records in the dataset. $P(X|C_i)$ is the posterior probability of X given C_i, and $P(X)$ is the probability of X in the dataset where sample X is a vector with n features in the form $X = (x_1, x_2, \ldots, x_n)$.

The Naive Bayes classifier assumes that features' values are conditionally independent of each other; hence, $P(X|C_i)$ can be calculated as follows:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

Typically, it is assumed that feature k has a Gaussian distribution with mean μ and standard deviation σ, defined by the following equation:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Therefore:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

4.3.2 | Lazy algorithms

A lazy classifier, in contrast to an eager classifier, defers constructing the model and making predictions until new data arrives. This approach eliminates the need for model construction until encountering new data, resulting in less model construction time. However, the time to make predictions can be significant. The most widely used lazy classifier is k-Nearest Neighbors (k-NN), which is described as follows:

k-nearest neighbors (k-NN): This classifier calculates the distance between the new received sample (e.g., flow or packet) and the samples in the dataset. The Euclidean distance, commonly used to calculate the distance of two samples, is defined as follows:[88]

$$distance(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

where x_{1i} and x_{2i} are the values of the i-th feature for samples x_1 and x_2, respectively, and n is the number of features. Then, the top k nearest samples to the received sample are selected, and the following equation[89] is computed:

$$p(C_i, x) = \frac{k_i}{k}$$

where C_i represents the class labels considered (e.g., [DDoS, normal]), $p(C_i, x)$ is the probability that sample x belongs to class C_i, and k_i is the number of samples in the k nearest samples that are members of class C_i.

The remainder of this section provides a summary of recent research on DDoS detection using the aforementioned supervised machine learning algorithms.

Doshi et al.[59] conducted a study on detecting attacks in an IoT network by implementing multiple supervised learning algorithms. They evaluated their approach using a physical testbed, as Figure 4 depicts. The testbed consisted of a local

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

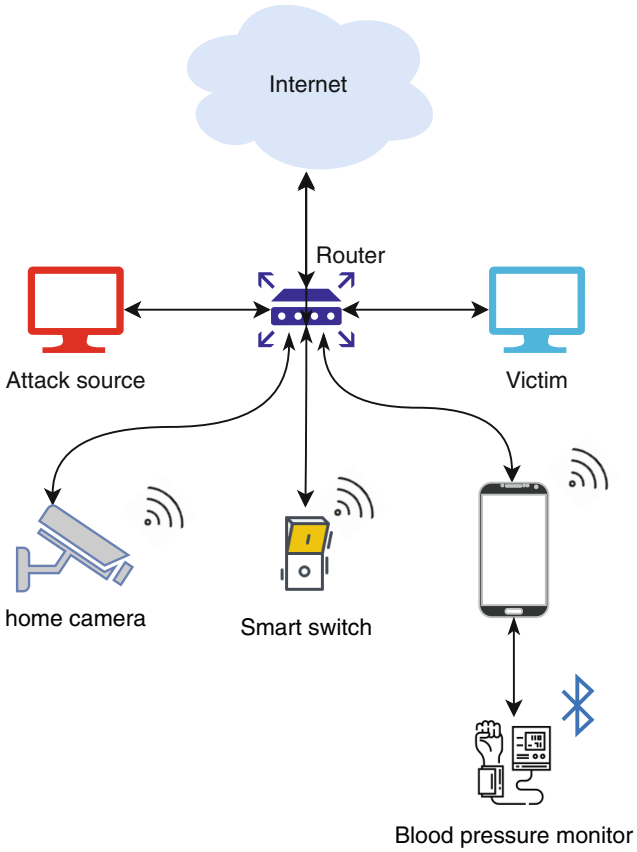

FIGURE 4 The physical testbed used for extracting the dataset in Reference 59.

network connected to an Internet router, with the local network comprising several IoT devices, a virtual machine running Kali Linux as the attacker (DoS source), a Raspberry Pi device executing an Apache web server as the DoS target, and some IoT devices engaged in standard network communication.

The study captured network traffic over a 10-min period, including approximately 1.5 min of attack traffic generated in random order. The attack traffic, consisting of HTTP Get Flood, TCP SYN Flood, and UDP Flood, was not produced by Mirai malware but was simulated to emulate common attacks conducted by Mirai. The DoS source sent the attack traffic to the DoS target with the IP and MAC addresses of all IoT devices on the network forged, causing the target to assume receiving a combination of normal and attack traffic from the same devices. The resulting dataset contained 491,855 packets, including 459,565 attack packets and 32,290 normal ones.

To apply ML algorithms, the study utilized two sets of features. The first set, stateless features, included packet size, inter-packet interval, and protocol. The second set, stateful features, included bandwidth, the number of distinct destination IP addresses, and the variation of destination IP addresses among time windows.

The study implemented several supervised learning algorithms, including DT by using Gini score, RF by using Gini score, SVM with the linear kernel (LSVM), k-NN, and ANN, 4-layer, feed-forward, 11 neurons in each layer. All the algorithms were implemented in Python using Keras library for ANN and Scikit-learn for the others. The train set consisted of 85% of the dataset, and the remaining 15% was used for the test set.

According to the results, RF had the most optimal detection rate of 0.999, precision and accuracy, and FPR of 0.002. LSVM had the lowest efficacy for detecting DDoS attacks with a detection rate of 0.999, precision and accuracy of 0.991, and FPR of 0.130.

Balkanli et al.[55] compared DT and NB classification algorithms and two IDSs using backscatter traffic for detecting DDoS attacks. They used the Gini score for DT and e1071 package for NB. Both classifiers were applied on two feature sets, where DT outperformed NB in terms of accuracy, and smaller datasets can lead to a higher detection rate. Although NB has a shorter training time than DT, its testing time is significantly higher. The effectiveness of the feature set is crucial, and the source IP, source port, and destination port have less impact on the result.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Roempluk and Surinta[66] conducted an investigation on DDoS detection utilizing machine learning algorithms by using the two most common datasets: "KDDCup99"[90] and "NSL-KDD".[91] In their study, the redundant records of the datasets were initially removed, and the non-numerical features were transformed into numerical ones. Then, they extracted three individual datasets from each of the sources, which consisted of series 1, series 2, and series 3. The datasets were derived based on the following criteria:

1. *Series 1*: The class label was determined as two values: [*DDoS*, *normal*].
2. *Series 2*: Only the attack records were extracted, and the class label was considered as six values (attack types): [*e, pod, smurf, teardrop, land, back*].
3. *Series 3*: This consisted of the Series 2 dataset combined with the normal records; hence, there were seven values for the class label: [*neptune, pod, smurf, teardrop, land, back, normal*].

The researchers used three supervised Machine Learning algorithms on the six aforementioned datasets:

1. *k-NN*: Multiple values were set for k ($k = 1, 3, 5, 7, 9$).
2. *SVM*: This was performed with two different kernels; a linear and a non-linear Kernel, specifically the Radial Basis Function (RBF).
3. *ANN*: More specifically, a Multi-Layer Perceptron (MLP) classifier was employed with various hidden layer counts (10, 50, 100, 150, 200, 500, and 1000).

Finally, the evaluation criteria of the classifiers was based on accuracy alone. The results showed that the k-NN classifier, with an accuracy of 99.99%, outperformed the other classifiers, including the MLP and the SVM.

Bakker et al.[28] conducted an evaluation of RF, k-NN, and SVM algorithms. Initially, the researchers tested the algorithms on a prevalent publicly-published dataset called "ISCX".[92] The F1-scores reported for RF, k-NN, and SVM during the evaluation were approximately 95%, 94%, and 93%, respectively. However, the result on a physical SDN testbed was disappointing. Although the accuracy and FPR (False Positive Rate) were acceptable, the detection rate had significantly declined. The detection rate values reported for SVM, k-NN, and RF on the SDN were 14%, 0.02%, and 0.005%, respectively. This indicates that the classifiers accurately identified the normal traffic, but most attack traffic was incorrectly classified as normal. The researchers attributed the observed degradation of the classifiers' detection rate on the physical network testbed to packet loss.

Rahman et al.[68] simulated an SDN testbed using Mininet[93] to evaluate DT, RF, SVM, and k-NN classifiers. The testbed included an attacker, a controller, an OpenFlow switch, some PCs communicating normally, and web servers as targets. The attacker launched ICMP and TCP flood attacks on the targets using the *hping3* tool, and normal traffic was subsequently sent. The captured traffic was used to extract a dataset, which underwent a pre-processing phase before the training procedure began using WEKA.[94] The results of the evaluation indicate that all classifiers had an F1-score, detection rate, accuracy, and precision of 1. The training time, that is, the learning model's construction time, and the testing time of the algorithms were also recorded. The average time of training and testing showed that DT was the fastest, while k-NN was the slowest, as demonstrated in Table 7.

Wani et al.[67] conducted an investigation on ML-based DDoS detection methods in a cloud environment. A computer with Kali Linux as the attacker machine and a botnet were employed to launch an attack on the ownCloud platform. SNORT was used in the cloud to detect the attack and extract a dataset from the traffic. The class label of the dataset, created by SNORT, included two values: [normal, suspicious]. Three classification algorithms, including SVM, RF, and NB, were deployed for attack detection. SVM had the highest detection rate of 99.8%, followed by RF with

TABLE 7 Training and testing time of the algorithms used in Reference 68.

Algorithm	Training time (s)	Testing time (s)	Average
DT	17.43	3.03	10.23
RF	171.11	5.19	88.15
SVM	168.59	1.97	85.28
k-NN	0.13	15957.7	7978.915

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

99.3%, and NB with 86.0%. Therefore, it can be inferred that SVM had the most optimum detection rate among the three algorithms.

Chen et al.[65] implemented multiple deep learning techniques, such as CNN, MC-CNN (Multi-Channel CNN), LSTM, in addition to several traditional machine learning methods, including RF, SVM, k-NN, and DT. KDDCup99 and CICIDS2017[95] were the two prevalent datasets used. They also split the features into subsets to feed each of them into channels in CNN. Table 8 indicates the results of the evaluation on CICIDS2017, where MC-CNN and DT had the highest and lowest accuracy, respectively.

Hou et al.[61] investigated three distinct types of features in their study. The first category consisted of flow-based features, which included variables such as the quantity of uploaded and downloaded packets. The second category consisted of pattern-based features that identified inbound and outbound packet and byte patterns. Lastly, the third category of features encompassed other variables such as the flow arrival interval and rate. They utilized many tools such as *LOIC*[*], *RUDY*[†], and *hping3* to generate attack traffic at their own research lab's network; hence, we consider the attack is comprised of both high-rate and low-rate types. They adopted two distinct approaches to creating their datasets: (1) using a 1:1000 sampling rate on the network traffic, and (2) without sampling. Four classifiers were utilized in the study, namely, RF, Adaboost, DT, and SVM. The RF algorithm demonstrated the highest efficiency, yielding an accuracy rate of 0.986 and an FPR of 0.024, according to the findings. This study's results indicate that RF, Adaboost, DT, and SVM were the most effective classifiers, exhibiting the highest accuracy and the lowest FPR. Several important observations can be drawn from this investigation:

1. The utilization of all three types of features identified in the study was found to produce the most optimal outcome.
2. The adopted sampling procedure can result in higher efficacy, despite excluding numerous samples from the original dataset. However, it should be noted that aggregating the features may elevate the model's accuracy in the sampled dataset, but conversely causes more noise in the non-sampled dataset.
3. Creating distinct class labels for each attack type or balancing can lead to superior results in the classification models.

Zhou et al.[60] proposed a real-time model for the detection and analysis of DDoS attack traffic. In this method, the traffic is transmitted from an edge switch to a machine executing Apache Kafka, which is a messaging tool for receiving stream data from multiple data sources and transmitting it to one or more destination applications. The target application in this model is a machine processing Apache Spark-streaming, which is a stream processing tool utilized to execute machine learning algorithms on traffic flow in real-time. The research team considered various attributes such as flow length and entropy of protocols when developing the model. To determine the most effective set of features, they employed a metric, as follows:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

where k denotes the number of features within the selected feature set, $\overline{r_{cf}}$ represents the average correlation between the selected features and the class label, and $\overline{r_{ff}}$ represents the average correlation among the selected features. The optimal feature set identified in the study comprises Incoming and Outgoing Ratio (Rio), Ratio of ICMP Protocol (Ri), Average

TABLE 8 Accuracy of the classifiers on CICIDS2017 dataset used in Reference 65.

Classifier	Accuracy
MC-CNN	98.87%
CNN	98.11%
LSTM	97.86%
RF	95.63%
SVM	95.46%
k-NN	93.54%
DT	92.60%

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Length of IP Flow (L_{ave_flow}), Source IP Address Number, and Destination IP Address Number Ratio (Rsd). Four datasets with varying normal and attack packets ratios were subsequently generated, consisting of purely normal traffic, light attacks, medium attacks, and heavy attacks. The performed attack types include TCP, UDP, and ICMP flood. Evaluation results demonstrate that DT, Logistic Regression (LR), and NB classifiers achieved the best performance in detecting DDoS attacks, based on the detection rate and FPR metrics.

Roopak et al.[69] conducted a comparative analysis of various machine learning and deep learning models for detecting DDoS attacks in IoT networks. The ML algorithms included SVM, RF, and NB, while the deep learning models consisted of MLP, LSTM, CNN, and a combination of CNN and LSTM. The results of their evaluation utilizing the CICIDS2017 dataset revealed that CNN+LSTM had the highest accuracy (97.16%) and detection rate (99.1%), while MLP demonstrated the lowest rate. Furthermore, the detection rate of SVM was reported to be 99.12%, which is approximately the same as that of CNN+LSTM.

Mohammed et al.[64] proposed a supervised machine learning approach tested on both a dataset and a physical network. The NB algorithm was initially trained using the NSL-KDD dataset and then tested on it, resulting in a low attack class precision rate of 0.02 and a high detection rate of 1.00. The trained model was subsequently tested on the authors' collected dataset from a SYN flood attack executed on a physical SDN. While the average precision and detection rate of the results were not impressive, the elimination of some redundant features boosted these rates by 62% and 54%, respectively.

Saini et al.[70] evaluated multiple supervised learning models such as NB, DT, RF, and ANN using a prepared dataset comprising 6527 records and 27 features. The dataset utilized for evaluation[71] consisted of four distinct types of DDoS attacks: Smurf, HTTP flood, UDP flood, and SQL injection DDoS. The results revealed that DT was the most optimal classifier based on precision and recall.

Morfino and Rampone[72] proposed a near-real-time intrusion detection system for IoT devices utilizing several supervised learning algorithms such as DT, RF, SVM, LR, and Gradient Boosting Tree (GBT). They evaluated the algorithms' performance in detecting the SYN flood attack using the Apache Spark framework with a public dataset.[73] Based on the findings, RF exhibited the highest accuracy rate of 1 for detecting the SYN flood attack.

Cvitić et al.[75] proposed a supervised approach to detect the increasing threat of DDoS attacks on IoT systems. The authors point out that traditional detection methods are often ineffective for IoT systems due to the unique characteristics of IoT traffic, which includes a diverse range of device types and communication protocols. To address this issue, a boosting-based detection model is proposed that uses Logistic Model Trees (LMTs) to classify network traffic from different IoT device classes in real-time. To evaluate the effectiveness of the proposed model, certain experiments are conducted using a dataset of network traffic from a simulated smart home environment. The authors categorized devices into four different classes based on the level of traffic predictability, and generated a separate LMT model for each device class. The evaluation results showed that the proposed approach achieved high accuracy for each of the four device classes, between 99.92% and 99.99%. The performance of the proposed model is also compared with other popular machine learning algorithms, including RF and SVM, and showed that the proposed model outperformed these algorithms in terms of accuracy and false positive rate.

Gupta et al.[76] proposed an approach for DDoS attacks in cloud computing environments using big data and deep learning techniques. The paper highlights the growing threat of DDoS attacks in cloud computing environments and the need for effective detection methods. The proposed method consists of three phases: data pre-processing, feature extraction, and deep learning-based classification. In the first phase, the raw network traffic data is collected and pre-processed using various techniques such as data filtering, normalization, and aggregation with the aim of Apache Spark. In the second phase, relevant features are extracted from the pre-processed data using statistical and machine learning-based methods. In the final phase, a deep learning-based classification model is trained and deployed for DDoS detection using TensorFlow. The authors used KDDCup99 for training and testing which showed a 99.73% of accuracy.

Kamaldeep et al.[80] proposed a feature engineering and machine learning framework for detecting DDoS attacks in standardized IoT networks using a novel dataset called "IoT-CIDDS," which contains 21 features and a single labelling attribute. The framework has two phases: in the first phase, the algorithms are developed for dataset enrichment and advanced feature engineering, including statistical analysis of the dataset with probability distribution and correlation among features. Specifically, RF is used for feature selection because it is able to strike balance of unbiased and noisy model with low variance. Moreover, RF is suitable to address multi-scale data with huge training samples. In the second phase, a machine learning model is used which are LR, SVM, DT, MLP, and RF. The results show that RF with recall 0.987 and FPR 0.01 is the best compared to the others in detecting DDoS attacks in the IoT network.

Zainudin et al.[78] proposed a deep learning method that combines a CNN and a LSTM network for detecting and classifying DDoS attacks in Software-Defined Industrial Internet of Things (IIoT) networks. The architecture consists of

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

three main components: data preprocessing, feature extraction, and classification. In the data preprocessing stage, the network traffic data is preprocessed to remove redundant and irrelevant features. In the feature extraction stage, the CNN model is used to extract the spatial features from the preprocessed data. In the classification stage, the LSTM model is used to classify the temporal features extracted from the CNN. The LSTM network is used to capture the sequential dependencies in the data and identify the patterns of DDoS attacks. The method is evaluated using a real-world dataset of network traffic collected from a Software-Defined IIoT testbed. The dataset includes various types of DDoS attacks, such as TCP SYN flood, UDP flood, and ICMP flood. The proposed approach achieved an accuracy of 98.9%, a precision of 98.2%, and a recall of 99.6% for detecting and classifying DDoS attacks. The FPR and FNR were also low, at 0.04% and 0.02%, respectively. The authors also compared the performance of their approach with several state-of-the-art methods, including K-means, decision tree, SVM, and deep learning-based methods. The proposed approach outperformed all other methods in terms of accuracy, precision, and recall.

Ismail et al.[96] proposed a method for detecting, classification and prediction of DDoS attacks using machine learning. The authors use the UNSW-NP 15 dataset to develop a framework for DDoS attack prediction, using the RF and XGBoost classification algorithms. The results show that both algorithms achieve high precision and recall, with an average accuracy of around 89% for RF and 90% for XGBoost. The comparison of the work to existing methods shows a significant improve of accuracy in detecting the attack.

4.4 | Unsupervised methods

Unsupervised learning is another type of learning where the data is not labeled. Clustering is the most commonly used unsupervised learning method. Clustering aims to group the data into distinct groups (clusters) based on their similarities. In the context of attack detection, clustering can be used to separate normal and attack traffic into individual clusters. There are two common categories of clustering: density-based and partitioning methods. Two popular clustering algorithms include:

1. **K-Means:** This algorithm takes the number of desired clusters, denoted as k, and the dataset as inputs. Initially, it selects k arbitrary records as the center points (centroids) of the clusters. K-Means assigns each record to the cluster that has the least distance from its centroid and then computes the new centroids. This iterative assigning process continues until the centroids no longer change.
2. **DBSCAN:** The DBSCAN algorithm operates on a dataset comprising a collection of records that are regarded as points in n-dimensional space. The algorithm relies on identifying dense regions as clusters, whereby two parameters, ϵ and min_points, are employed. Specifically, the parameter ϵ represents a neighborhood radius for each point p, and if the number of points in the ϵ neighborhood of p, including p itself, is not less than min_{points}, p is considered a core point. Another point q is characterized as a directly reachable point if it lies within a distance of ϵ from a core point p. It follows that a point q is density-reachable from a point p_i if there exists a path $p_1, \ldots, p_n$ such that $p_1 = p$ and $p_n = q$, where each point p_{i+1} is directly reachable from p_i. The algorithm traverses all points in the dataset to identify dense regions (clusters), and a noise point is a point that is not reachable by any other point. In broad terms, the DBSCAN algorithm comprises three steps:
   1. Identifying core points by determining the points in the ϵ neighborhood of each point
   2. Identifying non-core points adjacent to core points on the neighbor graph
   3. Assigning each edge point to a nearby cluster if possible

Within this section, we explore recent research in unsupervised ML techniques for detecting DDoS attacks.

Villalobos et al.[56] proposed a two-step unsupervised approach for DDoS detection. In the first step, a lightweight process of statistical methods is utilized for identifying flows that may be suspicious of an attack. In the subsequent step, suspicious flows are passed to an exhaustive ML algorithm to make the final determination. This second step involves the use of the K-means algorithm for clustering network traffic into normal and attack clusters. The testbed's architecture consists of three components: core nodes, edge nodes, and external agents, each carrying out specific responsibilities. External agents, which could be routers, for example, pass online NetFlow traffic to edge nodes for further analysis. Edge nodes, in turn, continuously receive NetFlow packets from external agents and pass them onto core nodes. Furthermore, edge nodes are responsible for transferring control commands to the external agents. Core nodes, on the other hand, are tasked with performing two decision-making operations, including executing the K-means algorithm and aggregating

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

received data. All processes, such as the K-means algorithm and aggregation of received data, occur on core nodes. Core nodes also generate control commands for the edge nodes.

To model the network, the researchers employed an in-memory distributed and directed graph as a data structure, in which each node can be stored and processed in a separate thread in one or multiple core nodes. Characteristics of the graph, such as the indegree and outdegree of the nodes, are considered features for the ML algorithm.

To evaluate the approach, the researchers utilized the Apache Storm for online processing on a cluster of computers and the Apache Kafka as the messaging framework to transfer the NetFlow traffic to the Apache Storm in real-time. They utilized a real-world DNS DDoS dataset,[97] which had a size of approximately 33 GB and implemented one core node, one edge node, and multiple external agents. The reported results show that the attack and benign traffic are successfully clustered in two separate clusters.

Dinçalp et al.[58] leveraged the DBSCAN algorithm to detect DDoS attacks. To evaluate their proposed approach, they conducted a TCP Flood attack on a physical testbed using a web server as the attack target. They collected two datasets: D1, which contains only normal traffic, and D2, which includes both normal and attack traffic. They determined that $\varepsilon = 0.03$ for D1, $\varepsilon = 0.08$ for D2, and $min_{points} = 15\%\ of\ dataset$ for both datasets resulted in optimal DBSCAN clustering. Finally, they reported that DBSCAN successfully separated attack and normal traffic and identified noise.

Al-mamory and Algelal[57] proposed an unsupervised approach for detecting DDoS attacks consisting of two primary phases: training and testing. In the training phase, the DBSCAN clustering algorithm is applied to two-thirds of the dataset, and centroids for each cluster are calculated. The cluster that contains the most points is deemed benign, whereas the other clusters are considered anomalous (DDoS). In the testing phase, the Euclidean distance of each sample from the remaining one-third of the data to the cluster centroids is computed, and the nearest centroid's label is assigned to the testing sample. The researchers evaluated their method on two datasets: "DARPA 2000"[98] and the "CAIDA DDoS attack 2007"[‡]. Based on their results, the detection rate, FPR, and accuracy for the DARPA dataset were approximately 52%, 0.68%, and 99%, respectively. Additionally, they compared DBSCAN with other clustering algorithms, but DBSCAN was deemed the most optimal.

4.5 | Hybrid methods

Both supervised and unsupervised methods for DDoS detection have their advantages and drawbacks. As a result, several methods combine both approaches to overcome their limitations while gaining their benefits. Additionally, some researchers have combined non-ML methods with ML techniques to enhance detection accuracy, such as using an entropy analysis approach paired with a classification algorithm. This section covers such hybrid methods in the field of DDoS detection.

Idhammad et al.[62] proposed a hybrid learning approach for DDoS detection consisting of three steps: entropy computation, co-clustering, and classification. First, the average entropy of four features, including *Source packet count*, *Destination packet count*, *Source byte count*, and *Destination byte count*, is computed for an online traffic time window. If the entropy value falls outside the specified range, the traffic is deemed suspicious of a DDoS attack. The entropy is calculated using Shannon's Entropy metric,

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

where $H(X)$ represents the entropy of feature X, n is the number of records in the current time window, and $p(x)$ denotes the probability of record x in the current time window.

In the second step, the traffic is divided into three co-clusters, and the information gain is calculated for each cluster. The cluster with the minimum gain is deemed normal, while the others are suspicious of being an attack. The formula for computing the information gain is as follows:

$$Gain(C) = avgH(W) - \frac{|C|}{|W|} \cdot avgH(C)$$

where C represents the given cluster, $avgH$ indicates the average entropy, and W denotes the entire time window.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Finally, in the third step, the Extra-Trees classification algorithm, which is an ensemble classifier similar to RF, identifies DDoS traffic. The approach's effectiveness is assessed on three datasets: "NSL-KDD",[91] "ISCXIDS2012",[92] and "UNSW-NB15".[99] The experimental results are summarized in Table 9.

Deepa et al.[63] proposed a detection method for SDN networks that combined two ML algorithms: (1) a supervised method using SVM; and (2) an unsupervised method using Self Organizing Map (SOM), a type of Artificial Neural Network (ANN) used for dimension reduction. Their method blocked any connections recognized as an attack, while passing other connections on to the SOM for further analysis. The authors evaluated the proposed method using a simulated SDN testbed with Scapy, an open-source network packet generator, to generate attack traffic. The results showed that combining the SVM and SOM algorithms resulted in approximately 5% higher detection rate and 50% lower FPR than using each algorithm individually.

Li et al.[74] proposed a real-time entropy-analysis method using an ANN algorithm to detect high-rate DDoS attacks. Their method analyzed traffic flow in real-time using a sliding window and computed the entropy of both source and destination IPs. To account for factors such as the target numbers and other policies, they introduced a joint entropy metric. To eliminate the effect of noise and jitter, they used LSTM, a type of recurrent neural network, to predict the value of entropy and subtract it from the real calculated value. The authors named their approach "Quintile Deviation Check," which detected the DDoS attack by analyzing changes in the entropy of traffic flows through the sliding window. They evaluated their method on three public datasets: "1999 DARPA",[100] "2009 DARPA",[101] and "CICDDoS2019," in addition to a dataset generated by the authors from a simulation of an SDN testbed.

Ali et al.[79] proposed a dual-stack machine learning framework for securing IoT-based maritime transportation systems. The authors explain that the rise of the IoT has led to an increased need for security in transportation systems, especially in maritime transportation, where cyber-attacks can have severe consequences. In the proposed framework named "Dual Stack Machine Learning (S2ML)," first, 10 features are extracted from the .pcap files of the traffic and then then entropy of them is calculated. Subsequently, the entropy-based features are passed to the ML framework, which comprises of an Alternating DT and a Simple LR models. Using majority voting, these models classify the data into two classes of benign and DDoS. Finally, the classified data are reclassified using an MLP neural network that uses entropy-based feature selection to select relevant features from the data generated by the IoT sensors. The evaluation is done using real-world data obtained from a shipping company's IoT-based maritime transportation system called "MST-IoT". The results show that the dual-stack machine learning framework is 1.5% more effective in detecting the attack, in terms of F1-score.

Najafimehr et al.[77] proposed a novel method combining supervised and unsupervised approaches for detecting unprecedented DDoS attacks. Their method involved separating DDoS flows from other flows into different clusters, partitioning and analyzing the points (flows) in each cluster based on their distances to one another, and calculating several statistical measures of distance values. Finally, a classifier determined the abnormality of each cluster. The authors evaluated their method by training the models on DDoS attacks in the CICIDS2017 dataset and testing on the CICDDoS2019 dataset. The results showed that using RF as the classifier and 20 partitions per cluster had the best efficacy, achieving 198% more Positive Likelihood Ratio (the ratio of recall to False Positive Rate) compared to using a conventional RF classifier alone for detecting attacks.

Nadeem Ali et al.[81] proposed a Weighted Federated Learning (WFL) model for detecting and mitigating the low-rate DDoS attack in the SDN control plane for IoT. The proposed model is based on local training of data using ANN to extract the weights of the trained model, which are then shared with the federated server for aggregation. Federated learning is a machine learning technique that allows multiple devices or parties to collaboratively train a model without sharing their data directly with each other, by aggregating the local models' parameters or weights instead of the raw data. The WFL model shows high prediction accuracy, sensitivity, and F1-Score, while maintaining a very low misclassification rate. The federated server assigns a unique preference to each locally trained model and aggregates all the local models

TABLE 9 Experimental result of the hybrid approach proposed in Reference 62.

Dataset	Accuracy (%)	FPR (%)
NSL-KDD	82.73	0.33
ISCXIDS2012	61.22	0.50
UNSW-NB15	86.57	0.36

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

to form a global model, which is shared back to the end-user device or local network for further attack detection and mitigation. The WFL model also brings pertinent benefits in terms of a smaller number of elements to transmit over the network, storage, and process, by only sharing the weights of the model, not the local device or network data, which also guarantees the privacy of the end-user data. According to the result, the prediction time per record is 0.019 ms. The WFL model has shown accurate compared with the existing approaches by 98.85% accuracy and 99.27% recall (according to our calculation on the confusion matrix).

5 | EXISTING DATASETS

In this section, we provide a discussion of some of the available and widely-used datasets commonly used for various machine learning-based network security applications. Table 10 summarizes the characteristics of these datasets. In order to select a dataset for detecting DDoS attacks, we think that there are a few criteria to consider:

1. **Size:** The dataset should be large enough to provide a comprehensive representation of the DDoS attacks. A larger dataset may also help in building more accurate and robust models. Especially, deep learning models require more data due to the complexity and number of parameters in deep neural networks. An important consideration when dealing with large and potentially complex data sets is resource limitations. This involves not only the ability to load such data into memory but also the computational resources necessary to process and analyze it effectively. In response, several bulk data analysis methods and distributed computing frameworks have emerged, including Apache Flink,[102] Apache Spark,[103] and TensorFlow.[104] These frameworks provide distributed processing capabilities that enable scalable, efficient, and flexible data analysis, even in the presence of resource limitations. Specifically, they offer tools to perform distributed processing on big data, effectively managing resource allocation and minimizing computing overhead. In this way, they help to address some of the major challenges of big data analysis and support the development of powerful data-driven applications.
2. **Diversity:** When building models to detect and mitigate DDoS attacks, it is crucial to ensure that the datasets used for training and evaluation are comprehensive and diverse. This diversity includes different types of attacks, various attack vectors, and varying levels of attack intensity. By incorporating varieties of DDoS attacks into the training data,

TABLE 10 The comparison of existing datasets.

				Records						
Dataset*	**Type**	**PCAP†**	**Year**	**All**	**DDoS**	**Benign**	**Other**	**Size‡**	**Features**	**Attack types**
KDDCup99	Flow-based	×	1999	4,898,431	3,883,370	972,781	42,280	743	42	Smurf, Neptune, back, pod, land, teardrop
NSL-KDD	Flow-based	×	2009	125,973	45,921	67,343	12,709	18	42	Smurf, Neptune, back, pod, land, teardrop
UNSW-NB15	Flow-based	✓	2015	2,540,044	16,353	2,218,761	304,930	559	49	Not mentioned
CICIDS2017 (DDoS subset)	Flow-based	✓	2017	225,745	128,027	97,718	—	92	80	TCP, UDP and HTTP flood
CICDDoS2019	Flow-based	✓	2019	≈ 46,000,000	≈ 45,939,000	≈ 61,000	—	≈ 22,000	80	Variety of reflection and flooding attacks
Edge-IIoTset	Packet-based	✓	2022	20,952,648	8,389,984	11,223,940	1,338,724	6,824	61	HTTP, TCP SYN, UDP and ICMP flooding attacks

*All the information in this table are about the training subset of the datasets.
†Are the PCAP files of the dataset available?
‡Size of the uncompressed CSV file in MegaBytes.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

the generated models gain the ability to recognize and respond to a wider range of attack types. This can significantly improve their accuracy and generalization capability, thereby increasing their resilience to different types of attacks. It is also important that the dataset includes modern and sophisticated types of attacks as attackers continue to create novel and unprecedented DDoS attacks. Therefore, the datasets should be continually updated with new attacks, ensuring that the models remain effective and can identify even the most complex and advanced attack patterns.

3. **Authenticity:** To ensure the effectiveness of machine learning models that detect DDoS attacks, it is important that the dataset used for training and evaluation of these models is representative of real-world scenarios. The dataset must be collected from real-world attacks and should reflect the techniques, mechanisms, and goals that attackers utilize in actual attacks. Therefore, the dataset should include data from actual scenarios where DDoS attacks have occurred, and the attacks should have been carried out by real attackers. This can provide a more accurate representation of the types of attacks that can be launched in the real world, as well as the nature and patterns of these attacks. Ultimately, such a dataset could lead to the development of more effective and reliable machine learning models that can better detect and mitigate DDoS attacks in the future.
4. **Labeling:** To ensure the reproducibility and reliability of research results, it is essential to have access to high-quality datasets that provide accurate and detailed labels for cyber attacks. In particular, these labels should include relevant information that is critical for the analysis and classification of the attacks, such as the type of attack, the attack vector, and the attack intensity. Having such detailed labels can enable researchers to compare and validate the results of different studies, and can also help improve the understanding of the mechanisms and strategies used by attackers. Therefore, it is important to ensure that datasets used in cyber security research meet these criteria and are carefully curated to provide the maximum utility for the research community.
5. **Real-world testing:** Evaluating the effectiveness of ML-based attack detection models through real-world testing is crucial for validating their practical utility and demonstrating their capability to detect attacks in diverse and complex scenarios. Such tests involve running the ML models on live or emulated network traffic, allowing the model's performance to be evaluated in a realistic environment where multiple factors affect the detection accuracy. To accomplish this, it is necessary to have access to network traffic files, such as PCAP files, that contain a wide range of representative attack patterns that can be used to evaluate the model's ability to detect threats accurately. Therefore, researchers working in the field of cyber security should consider using real network testbeds or emulation to evaluate their ML detection models comprehensively and accurately.

5.1 | KDD Cup 99

The dataset referred to as the "KDD Cup 99" dataset[90,105] was developed by the University of California, Irvine in 1999, using data from the 1998 DARPA program of the MIT Lincoln Labs. The dataset is publicly available and represents a simulation of a military LAN environment. It contains both a training dataset, which spans a period of 7 weeks, and a testing dataset, which spans a period of 2 weeks. Each record in the dataset represents a flow of packets.

A noteworthy aspect of this dataset is its distribution of records. Specifically, approximately 79% of the records in the training subset contain DDoS attacks, 20% contain benign activity, and 1% contain other malicious activities. The records in the dataset contain 41 features, including the protocol, provided service, duration, number of bytes, and the flags set in the connection.

It is important to note that the probability distribution of records in the training set is not representative of the distribution in the testing set. Additionally, the training set contains 14 more types of attacks, including non-DDoS attacks, that do not exist in the testing set. The attack types represented in the training subset of KDDCup99 dataset are as follows:

1. *Netptune*: The Neptune attack, also known as the SYN flood attack, is a type of denial of service attack that floods a target server with a high volume of TCP SYN packets.[106,107]
2. *Smurf*: The Smurf attack is a form of DDoS attack, where an attacker sends a high volume of ICMP Echo packets to target by spoofing the victim's IP address.[108,109]
3. *Back*: The Back attack is an application-layer denial of service attack that targets Apache web servers by sending requests with a significant number of "\" characters, which can cause the server to crash.[110]
4. *PoD*: The Ping of Death (PoD) attack takes advantage of the vulnerability in some systems' maximum packet fragmentation size to send large fragmented ICMP packets that can overflow the target system's buffer and cause a crash.[110]

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5. *Land*: The Local Area Network Denial (LAND) attack is a type of denial of service attack that exploits an old TCP/IP implementation vulnerability by sending TCP-SYN packets with the same source and destination IP addresses, causing the target to respond to itself in an endless loop.[106]
6. *Teardrop*: The Teardrop attack is a denial of service attack that exploits a vulnerability in the reassembly of overlapping fragmented IP datagrams, causing the system to crash or become unresponsive.[111,112]

However, Tavallaee et al.[91] have identified certain limitations in the dataset under consideration. Notably, over 70% of the records exhibit redundancies. This, in turn, may lead classifiers to exhibit a bias towards repetitive records, posing potential challenges to the generalizability of the results.

5.2 | NSL-KDD

In order to address the inadequacies of the widely-used KDDCup99 dataset, Tavallaee et al. introduced the NSL-KDD dataset in 2009.[91] This alternative dataset retains the same features and attack types as its predecessor, but reduces the number of records. Specifically, redundant and similarly challenging records are removed to enhance the effectiveness of classifiers in detecting attacks.

5.3 | UNSW-NB15

The UNSW-NB15[98] is a network intrusion detection dataset that contains a diverse range of network traffic data, including DoS attacks, port scans, and other types of network intrusions. It was created in 2015 by the University of New South Wales in Australia to address some of the limitations of the KDDCup99 dataset. The testbed from which the dataset is collected consists of three servers, two routers, some clients and the IXIA traffic generator for producing attack traffic. The CSV files in this dataset contains about 2.5 million flows, including normal traffic and attack traffics such as DoS, worm and backdoor. However, the type of the DoS attack is not mentioned.

5.4 | CICIDS2017

The dataset under consideration has been introduced by Sharafaldin et al.[95] affiliated with the Canadian Institute for Cybersecurity, University of New Brunswick, in 2017. The dataset, which is publicly available,[113] encompasses various attacks; however, this study focuses solely on the DDoS subset, which is formed by utilizing the LOIC tool for TCP, UDP, and HTTP flooding attacks. In total, the dataset comprises more than 2000 records, including both DDoS and benign traffic samples, with 80 distinct features. The authors' assessments have shown that the standard deviation of the backward packet's length in a flow, the average size of the packets of a flow, the flow duration, and the standard deviation of the flow inter-arrival time are the most salient features for detecting DDoS attacks.

5.5 | CICDDoS2019

The CICDDoS2019 dataset has been proposed by Sharafaldin et al.[114] from the Canadian Institute for Cybersecurity, University of New Brunswick, in 2019, and is publicly available.[115] It shares a similar numeric feature set with CICIDS2017, but some of its records include the *infinity* value, necessitating preprocessing considerations. Unlike CICIDS2017, CICDDoS2019 comprises a greater range of attack types, including NetBIOS, SSDP, CharGen, LDAP reflection attacks, as well as traditional attacks like SYN and UDP flood attacks. The dataset contains approximately 46 million records, with individual CSV files of different attack types' records and some benign-labeled records available, and the authors have also made the PCAP file of the captured traffic accessible.

5.6 | Edge-IIoTset

The IoT and IIoT are rapidly growing phenomenones, and their importance in various fields cannot be overstated. With the increasing number of connected devices and systems, these technologies have significantly improved efficiency, productivity, and safety in various industries. However, these devices and systems are also vulnerable to cyber attacks,

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

including DDoS, which can cause massive disruptions and significant financial losses.[116-118] The need for securing IoT and IIoT systems against such attacks is crucial as the cost of data breaches and cyber attacks is increasing each year. Therefore, it is essential to develop effective security measures to prevent and mitigate the impact of DDoS attacks on IoT and IIoT systems. Hence, we introduce a recently proposed dataset, specially designed for this IoT and IIoT security research in this section.

The Edge-IIoTset dataset[119,120] is a comprehensive and realistic cyber security dataset published in 2022 and designed for IoT and IIoT applications. It can be utilized by machine learning-based intrusion detection systems in either of centralized or federated learning modes. The testbed used for collecting this dataset is organized into seven layers, each featuring new emerging technologies that meet the requirements of IoT and IIoT applications. Various IoT devices are used to generate data, including low-cost digital sensors, ultrasonic sensors, heart rate sensors, and flame sensors. Fourteen attacks related to IoT and IIoT connectivity protocols are identified and analyzed, categorized into five threats, and 61 features with high correlations from 1176 found features are proposed. The DDoS attacks included in this dataset are HTTP flood, TCP Syn Flood, UDP Flood and ICMP Flood. The csv files contain more than 20 million records overall, and each record represents a network packet.

6 | CONCLUSION AND FUTURE WORK

This paper has provided an in-depth analysis of machine learning-based approaches used to identify various types of DDoS attacks. Our investigation reveals that while supervised learning methods are effective, they require pre-labeled datasets and training, which is unfeasible for not-yet-known attacks. In contrast, unsupervised methods can be applied more widely to distinguish DDoS attack traffic from benign traffic under unknown circumstances, albeit with less accuracy and detection ability than supervised methods. Combining both supervised and unsupervised methods, along with non-ML methods, may offer the most effective approach to identify known or unknown attacks.

However, due to emerging novel and unknown types of DDoS attacks, there are noticeable differences between known and lab-based train datasets and the unforeseen factors that occur in real DDoS attacks. Consequently, the recall is low while the false-negative rate is high. We recommend further research on developing resilient and effective methods that accurately detect malicious traffic under real attack scenarios and different test datasets.

Furthermore, the present datasets employed for DDoS research have certain limitations. For instance, the KDDCup99 and NSL-KDD datasets have become outdated and do not encompass the latest innovative and advanced DDoS attacks. Similarly, the CICIDS2017 and Edge-IIoTset lack several novel types of DDoS attacks, rendering them inadequate for such detection purposes. Moreover, the CICDDoS2019 dataset is not suitable for identifying slow and low-rate attacks, and it is also imbalanced with benign-labeled records accounting for less than 1%. These limitations in current datasets underline the need for further and sustained research to provide future-oriented and up-to-date datasets that can assist in the detection and mitigation of DDoS attacks in diverse network environments.

In addition to conducting comprehensive research to address the limitations of existing methods and datasets, we propose that researchers focus on developing novel forms of DDoS attacks to proactively anticipate the malicious techniques that may be employed by attackers. Introducing innovative attack types, such as the SlowDrop attack,[25] may serve as a crucial measure towards preparing for and mitigating future DDoS attacks.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

None.

ENDNOTES

*The LOIC is used for conducting HTTP, TCP, and UDP flood attacks.
†The "R U Dead Yet" (RUDY) is a tool for executing low-rate DDoS attacks.
‡https://www.caida.org/data/passive/ddos-20070804_dataset.xml.

ORCID

Mohammad Najafimehr https://orcid.org/0000-0001-9720-1148
Seyedakbar Mostafavi https://orcid.org/0000-0003-3530-2642

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

REFERENCES

1. Kumari P, Jain AK. A comprehensive study of DDoS attacks over IoT network and their countermeasures. *Comput Secur*. 2023;127:103096. doi:10.1016/j.cose.2023.103096
2. Cai T, Jia T, Adepu S, Li Y, Yang Z. ADAM: an adaptive DDoS attack mitigation scheme in software-defined cyber-physical system. *IEEE Trans Industr Inform*. 2023;1-12. doi:10.1109/TII.2023.3240586
3. CloudFlare, Inc. Cloudflare DDoS threat report for 2022 Q4. 2023. https://blog.cloudflare.com/ddos-threat-report-2022-q4/
4. Tushir B, Dalal Y, Dezfouli B, Liu Y. A quantitative study of DDoS and E-DDoS attacks on WiFi smart home devices. *IEEE Internet Things J*. 2021;8(8):6282-6292. doi:10.1109/JIOT.2020.3026023
5. Khaing MS, Thant YM, Tun T, Htwe CS, Thwin MMS. IoT botnet detection mechanism based on UDP protocol. Paper presented at: 2020 IEEE Conference on Computer Applications (ICCA). 2020:1-7.
6. Zhou L, Liao M, Yuan C, Zhang H. Low-rate DDoS attack detection using expectation of packet size. *Secur Commun Netw*. 2017;2017. doi:10.1155/2017/3691629
7. Zhang B, Zhang T, Yu Z. DDoS detection and prevention based on artificial intelligence techniques. In: *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China; 2017:1276-1280. doi: 10.1109/CompComm.2017.8322748
8. Bhandari A, Kumar K, Sangal AL, Behal S. An anomaly based distributed detection system for DDoS attacks in Tier-2 ISP networks. *J Ambient Intell Humaniz Comput Secur*. 2020;12(1):1387-1406. doi:10.1007/s12652-020-02208-3
9. Lange T, Kettani H. On security threats of botnets to cyber systems. Paper presented at: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE. 2019:176-183.
10. Hoque N, Bhattacharyya DK, Kalita JK. Botnet in DDoS attacks: trends and challenges. *IEEE Commun Surv Tutor*. 2015;17(4):2242-2270.
11. Zargar ST, Joshi J, Tipper D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun Surv Tutor*. 2013;15(4):2046-2069.
12. AWS Shield. Threat Landscape Report–Q1. 2020. https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf
13. ZDNET. AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever. 2021 https://www.zdnet.com/article/aws-said-it-mitigated-a-2-3-tbps-ddos-attack-the-largest-ever/
14. Kottler S. February 28th DDoS incident report. 2020. https://github.blog/2018-03-01-ddos-incident-report/
15. Dyn Blog. Dyn analysis summary of Friday October 21 attack. 2020. https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/
16. Shi L, Li J, Zhang M, Reiher P. On capturing DDoS traffic footprints on the internet. *IEEE Trans Dependable Secure Comput*. 2022;19(4):2755-2770. doi:10.1109/TDSC.2021.3074086
17. Sultana N, Chilamkurti N, Peng W, Alhadad R. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer Peer Netw Appl*. 2019;12(2):493-501.
18. Zoppi T, Ceccarelli A, Salani L, Bondavalli A. On the educated selection of unsupervised algorithms via attacks and anomaly classes. *J Inf Secur Appl*. 2020;52:102474. doi:10.1016/j.jisa.2020.102474
19. Zekri M, El Kafhali S, Aboutabit N, Saadi Y. DDoS attack detection using machine learning techniques in cloud computing environments. Paper presented at: 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), IEEE. 2017: 1-7.
20. Bereziński P, Jasiul B, Szpyrka M. An entropy-based network anomaly detection method. *Entropy*. 2015;17(4):2367-2408. doi:10.3390/e17042367
21. Abdulhammed R, Faezipour M, Abuzneid A, AbuMallouh A. Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. *IEEE Sens Lett*. 2019;3(1):1-4. doi:10.1109/LSENS.2018.2879990
22. Krishnaveni S, Vigneshwar P, Kishore S, Jothi B, Sivamohan S. Anomaly-based intrusion detection system using support vector machine. In: Dash SS, Lakshmi C, Das S, Panigrahi BK, eds. *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer Singapore; 2020:723-731.
23. Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comput Commun Rev*. 2004;34(2):39-53.
24. Rios VM, Inácio PR, Magoni D, Freire MM. Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms. *Comput Netw*. 2021;186:107792. doi:10.1016/j.comnet.2020.107792
25. Cambiasoa E, Chiola G, Aiello M. Introducing the SlowDrop attack. *Comput Netw*. 2019;150:234-249. doi:10.1016/j.comnet.2019.01.007
26. Cybersecurity and Infrastructure Security Agency CISA. Alert (TA14-017A) UDP-based amplification attacks. 2021 https://us-cert.cisa.gov/ncas/alerts/TA14-017A
27. Labs BL. A new DDoS reflection attack: portmapper; an early warning to the industry. 2021. https://blog.lumen.com/a-new-ddos-reflection-attack-portmapper-an-early-warning-to-the-industry/
28. Bakker JN, Ng B, Seah WKG. Can machine learning techniques be effectively used in real networks against DDoS attacks? Paper presented at: 2018 27th International Conference on Computer Communication and Networks (ICCCN). 2018:1-6.
29. Zhijun W, Wenjing L, Liang L, Meng Y. Low-rate DoS attacks, detection, defense, and challenges: a survey. *IEEE Access*. 2020;8:43920-43943.
30. Pascoal TA, Fonseca IE, Nigam V. Slow denial-of-service attacks on software defined networks. *Comput Netw*. 2020;173:107223. doi:10.1016/j.comnet.2020.107223
31. Isyaku B, Mohd Zahid MS, Bte Kamat M, Abu Bakar K, Ghaleb FA. Software defined networking flow table management of OpenFlow switches performance and security challenges: a survey. *Future Internet*. 2020;12(9). doi:10.3390/fi12090147

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

32. Cloudflare, Inc. What is a low and slow attack? Low and slow DDoS attack definition | Cloudflare UK. 2021. https://www.cloudflare.com/en-gb/learning/ddos/ddos-low-and-slow-attack/
33. Lemon J. Resisting SYN flood DoS attacks with a SYN cache. Paper presented at: BSDCon 2002. 2002: 89-97.
34. Ranjana P, Kalai Vani YS. Anomaly detection of DDOS attacks using Hadoop. *Emerging Research in Computing, Information, Communication and Applications*. Springer; 2019:543-552.
35. Kolahi SS, Treseangrat K, Sarrafpour B. Analysis of UDP DDoS flood cyber attack and defense mechanisms on web server with Linux Ubuntu 13. Paper presented at: 2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15). 2015:1-5.
36. Imperva. Ping flood (ICMP flood). 2021 https://www.imperva.com/learn/ddos/ping-icmp-flood/
37. Bhuyan MH, Bhattacharyya D, Kalita JK. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Pattern Recognit Lett*. 2015;51:1-7.
38. Bogdanoski M, Suminoski T, Risteski A. Analysis of the SYN flood DoS attack. *Int J Comput Netw Inf Secur (IJCNIS)*. 2013;5(8):1-11.
39. Eddy W. RFC 4987–TCP SYN flooding attacks and common mitigations. 2021 https://tools.ietf.org/html/rfc4987
40. Tandon V, Devore J. Detecting lag switch cheating in game. 2017 US Patent 9,636,589.
41. Masdari M, Jalali M. A survey and taxonomy of DoS attacks in cloud computing. *Secur Commun Netw*. 2016;9(16):3724-3751. doi:10.1002/sec.1539
42. Krämer L, Krupp J, Makita D, et al. AmpPot: monitoring and defending against amplification DDoS attacks. In: Bos H, Monrose F, Blanc G, eds. *Research in Attacks, Intrusions, and Defenses*. Springer International Publishing; 2015:615-636.
43. Symantec Corp. MS SQL PacketResolution DoS: Attack Signature. 2020 https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=20533
44. Liu X, Zheng L, Cao S, et al. *A Multi-Location Defence Scheme against SSDP Reflection Attacks in the Internet of Things*. Springer; 2019:187-198.
45. CLOUDFLARE. SSDP DDoS Attack. 2021. https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/
46. Postel J. RFC 864–Character generator protocol. 2021. https://tools.ietf.org/html/rfc864
47. F5 Labs. Old Protocols, New Exploits: LDAP unwittingly serves DDoS amplification attacks. 2021. https://www.f5.com/labs/articles/threat-intelligence/old-protocols-new-exploits-ldap-unwittingly-serves-ddos-amplification-attacks-22609
48. Akamai Technologies, Inc. Threat advisory: NetBIOS name server. RPC Portmap and Sentinel Reflection DDoS.
49. Rossow C. Amplification hell: revisiting network protocols for DDoS abuse. Paper presented at: Network and Distributed System Security Symposium. 2014.
50. Cybersecurity and Infrastructure Security Agency CISA. UDP-based amplification attacks. 2021. https://us-cert.cisa.gov/ncas/alerts/TA14-017A
51. Alrehan AM, Alhaidari FA. Machine learning techniques to detect DDoS attacks on VANET system: a survey. Paper presented at: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS). 2019:1-6.
52. Loukas G, Öke G. Protection against denial of service attacks: a survey. *Comput J*. 2010;53(7):1020-1037. doi:10.1093/comjnl/bxp078
53. Bhardwaj A, Subrahmanyam GVB, Avasthi V, Sastry H, Goundar S. DDoS attacks, new DDoS taxonomy and mitigation solutions–A survey. Paper presented at: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). 2016:793-798.
54. Kamboj P, Trivedi MC, Yadav VK, Singh VK. Detection techniques of DDoS attacks: a survey. Paper presented at: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON). 2017:675-679.
55. Balkanli E, Alves J, Zincir-Heywood AN. Supervised learning to detect DDoS attacks. Paper presented at: 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), IEEE. 2014:1-8.
56. Villalobos JJ, Rodero I, Parashar M. An unsupervised approach for online detection and mitigation of high-rate DDoS attacks based on an In-memory distributed graph using streaming data and analytics. Paper presented at: Bdcat'17. Association for Computing Machinery; New York, NY, USA. 2017: 103–112.
57. Al-mamory SO, Algelal ZM. A modified DBSCAN clustering algorithm for proactive detection of DDoS attacks. Paper presented at: 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT). 2017:304-309.
58. Dincalp U, Güzel MS, Sevine O, Bostanci E, Askerzade I. Anomaly based distributed denial of service attack detection and prevention with machine learning. Paper presented at: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). 2018:1-4.
59. Doshi R, Apthorpe N, Feamster N. Machine learning DDoS detection for consumer internet of things devices. Paper presented at: 2018 IEEE Security and Privacy Workshops (SPW). 2018:29-35.
60. Zhou B, Li J, Wu J, Guo S, Gu Y, Li Z. Machine-learning-based online distributed denial-of-service attack detection using spark streaming. Paper presented at: 2018 IEEE International Conference on Communications (ICC). 2018:1-6.
61. Hou J, Fu P, Cao Z, Xu A. Machine learning based DDos detection through NetFlow analysis. Paper presented at: MILCOM 2018–2018 IEEE Military Communications Conference (MILCOM). 2018:1-6.
62. Idhammad M, Afdel K, Belouch M. Semi-supervised machine learning approach for DDoS detection. *Appl Intell*. 2018;48(10):3193-3208.
63. Deepa V, Sudar KM, Deepalakshmi P. Detection of DDoS attack on SDN control plane using hybrid machine learning techniques. Paper presented at: 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE. 2018:299-303.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

64. Mohammed SS, Hussain R, Senko O, et al. A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network. Paper presented at: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE. 2018:1-8.
65. Chen J, Yang YT, Hu KK, Zheng HB, Wang Z. DAD-MCNN: DDoS attack detection via multi-channel CNN. Paper presented at: ICMLC'19. Association for Computing Machinery, New York, NY, USA. 2019:484-488.
66. Roempluk T, Surinta O. A machine learning approach for detecting distributed denial of service attacks. Paper presented at: 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON). 2019:146-149.
67. Wani AR, Rana QP, Saxena U, Pandey N. Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. Paper presented at: 2019 Amity International Conference on Artificial Intelligence (AICAI). 2019: 870-875.
68. Rahman O, Quraishi MAG, Lung C. DDoS attacks detection and mitigation in SDN using machine learning. Paper presented at: 2019 IEEE World Congress on Services (SERVICES). 2019:184-189.
69. Roopak M, Tian GY, Chambers J. Deep learning models for cyber security in IoT networks. Paper presented at: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE. 2019:0452-0457.
70. Saini PS, Behal S, Bhatia S. Detection of DDoS attacks using machine learning algorithms. Paper presented at: 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom). 2020:16-21.
71. Alkasassbeh M, Al-Naymat G, Hassanat A, Almseidin M. Detecting distributed denial of service attacks using data mining techniques. *Int J Adv Comput Sci Appl*. 2016;7(1):436-445.
72. Morfino V, Rampone S. Towards near-real-time intrusion detection for IoT devices using supervised learning and apache spark. *Electronics*. 2020;9(3):444.
73. Mirsky Y, Doitshman T, Elovici Y, Shabtai A. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089. 2018.
74. 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.
75. Cvitić I, Perakovic D, Gupta BB, Choo KKR. Boosting-based DDoS detection in internet of things systems. *IEEE Internet Things J*. 2021;9(3):2109-2123. doi:10.1109/JIOT.2021.3090909
76. Gupta BB, Gaurav A, Peraković D. A big data and deep learning based approach for DDoS detection in cloud computing environment. Paper presented at: 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE). 2021:287-290.
77. Najafimehr M, Zarifzadeh S, Mostafavi S. A hybrid machine learning approach for detecting unprecedented DDoS attacks. *J Supercomput*. 2022;78(6):8106-8136. doi:10.1007/s11227-021-04253-x
78. Zainudin A, Ahakonye LAC, Akter R, Kim D-S, Lee J-M. An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks. *IEEE Internet Things J*. 2023;10(10):8491-8504. doi:10.1109/JIOT.2022.3196942
79. Ali F, Sarwar S, Shafi QM, Iqbal M, Safyan M, Qayyum ZU. Securing IoT based maritime transportation system through entropy-based dual-stack machine learning framework. *IEEE Trans Intell Transp Syst*. 2023;24(2):2482-2491. doi:10.1109/TITS.2022.3177772
80. Kamaldeep MM, Dutta M. Feature engineering and machine learning framework for DDoS attack detection in the standardized internet of things. *IEEE Internet Things J*. 2023;10(10):8658-8669. doi:10.1109/JIOT.2023.3245153
81. Ali MN, Imran M, Salah ud din M, Kim BS. Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. *Appl Sci*. 2023;13(3). doi:10.3390/app13031431
82. Wang M, Lu Y, Qin J. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. *Comput Secur*. 2020;88:101645. doi:10.1016/j.cose.2019.101645
83. Thomas T, Vijayaraghavan A, Emmanuel S. *Machine Learning and Cybersecurity*. Springer Singapore; 2020:37-47.
84. Banitalebi Dehkordi A, Soltanaghaei M, Boroujeni FZ. The DDoS attacks detection through machine learning and statistical methods in SDN. *J Supercomput*. 2021;77(3):2383-2415. doi:10.1007/s11227-020-03323-w
85. Pande S, Khamparia A, Gupta D, Thanh DNH. *DDOS Detection Using Machine Learning Technique*. Springer Singapore; 2021:59-68.
86. Almiani M, AbuGhazleh A, Jararweh Y, Razaque A. DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network. *Int J Mach Learn Cybern*. 2021;12:1-13. doi:10.1007/s13042-021-01323-7
87. Othman SM, Ba-Alwi FM, Alsohybe NT, Al-Hashida AY. Intrusion detection model using machine learning algorithm on big data environment. *J Big Data*. 2018;5(1):34. doi:10.1186/s40537-018-0145-4
88. Han J, Pei J, Kamber M. *Data Mining: Concepts and Techniques*. 3rd ed. Elsevier; 2012.
89. Alpaydin E. *Introduction to Machine Learning*. 2nd ed. The MIT Press; 2010.
90. California UoI. KDD Cup 1999 Data. 2021. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
91. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. Paper presented at: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada. 2009:1-6.
92. Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput Secur*. 2012;31(3):357-374.
93. Mininet. Mininet: An Instant Virtual Network on your Laptop (or other PC). 2020. http://mininet.org/
94. Weka 3. Data Mining with Open Source Machine Learning Software in Java. 2020. https://www.cs.waikato.ac.nz/ml/weka/
95. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Paper presented at: 4th International Conference on Information Systems Security and Privacy. 2018:108-116.

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

96. Ismail MMI, Hussain H, Khan AA, et al. A machine learning-based classification and prediction technique for DDoS attacks. *IEEE Access*. 2022;10:21443-21454. doi:10.1109/ACCESS.2022.3152577
97. Merit Network Inc. DNS AMPL DDOS DEC2015 (07/22/2015 to 07/22/2015). 2016. doi: 10.23721/105/1354086
98. Lincoln Laboratory, Massachusetts Institute of Technology. 2000 DARPA intrusion detection scenario specific datasets. 2020. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets
99. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Paper presented at: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE. 2015:1-6.
100. Lincoln Laboratory, Massachusetts Institute of Technology. 1999 DARPA intrusion detection evaluation dataset. 2021: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset
101. Colorado State University. DARPA. Intrusion Detection Dataset. 2009: http://www.darpa2009.netsec.colostate.edu/
102. The Apache Software Foundation. Apache Flink. 2023. https://flink.apache.org/
103. The Apache Software Foundation. Apache spark™; unified analytics engine for big data. 2023. https://spark.apache.org/
104. Google. TensorFlow. 2023. https://www.tensorflow.org/
105. California UoI. KDD-CUP-99 Task Description. 2021. http://kdd.ics.uci.edu/databases/kddcup99/task.html
106. Gupta BB, Badve OP. Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. *Neural Comput Appl*. 2017;28(12):3655-3682. doi:10.1007/s00521-016-2317-5
107. Haddadi M, Beghdad R. DoS-DDoS: taxonomies of attacks, countermeasures, and well-known defense mechanisms in cloud environment. *EDPACS*. 2018;57(5):1-26. doi:10.1080/07366981.2018.1453101
108. Cloudflare, Inc. Smurf DDoS attack. 2021. https://www.cloudflare.com/learning/ddos/smurf-ddos-attack/
109. Kleiman D, Cardwell K, Clinton T, et al., eds. Computer-assisted attacks and crimes. *The Official CHFI Study Guide (Exam 312-49)*. Syngress; 2007:387-439.
110. Salim MM, Rathore S, Park JH. Distributed denial of service attacks and its defenses in IoT: a survey. *J Supercomput*. 2020;76(7):5320-5363. doi:10.1007/s11227-019-02945-z
111. Marin G. Network security basics. *IEEE Secur Priv*. 2005;3(6):68-72. doi:10.1109/MSP.2005.153
112. Radware Ltd. Teardrop attack. 2021. https://www.radware.com/security/ddos-knowledge-center/ddospedia/teardrop-attack
113. Sharafaldin I, Habibi Lashkari A, Ghorbani AA. Intrusion detection evaluation dataset (CIC-IDS2017). https://www.unb.ca/cic/datasets/ids-2017.html
114. Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. Paper presented at: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE. 2019:1-8.
115. Sharafaldin I, Habibi Lashkari A, Hakak S, Ghorbani AA. DDoS evaluation dataset (CIC-DDoS2019). https://www.unb.ca/cic/datasets/ddos-2019.html
116. Yazdinejad A, Zolfaghari B, Dehghantanha A, Karimipour H, Srivastava G, Parizi RM. Accurate threat hunting in industrial internet of things edge devices. *Digit Commun Netw*. 2022. In Press. doi:10.1016/j.dcan.2022.09.010
117. Amanullah MA, Habeeb RAA, Nasaruddin FH, et al. Deep learning and big data technologies for IoT security. *Comput Commun*. 2020;151:495-517. doi:10.1016/j.comcom.2020.01.016
118. Aljuhani A. Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access*. 2021;9:42236-42264. doi:10.1109/ACCESS.2021.3062909
119. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*. 2022;10:40281-40306. doi:10.1109/ACCESS.2022.3165809
120. Ferrag MA. Edge-IIoTset cyber security dataset of IoT & IIoT. 2023. https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot

How to cite this article: Najafimehr M, Zarifzadeh S, Mostafavi S. DDoS attacks and machine-learning-based detection methods: A survey and taxonomy. *Engineering Reports*. 2023;5(12):e12697. doi: 10.1002/eng2.12697

25778196, 2023, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12697, Wiley Online Library on [28/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License