electronics

Review

DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions

Amal A. Alahmadi [1], Malak Aljabri [2], Fahd Alhaidari [1], Danyah J. Alharthi [1], Ghadi E. Rayani [1], Leena A. Marghalani [1], Ohoud B. Alotaibi [1,*] and Shurooq A. Bajandouh [1]

[1] SAUDI ARAMCO Cybersecurity Chair, Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia; aaalahmadi@iau.edu.sa (A.A.A.); faalhaidari@iau.edu.sa (F.A.); 2190004427@iau.edu.sa (D.J.A.); 2190000354@iau.edu.sa (G.E.R.); 2190004310@iau.edu.sa (L.A.M.); 2190004908@iau.edu.sa (S.A.B.)

[2] Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia; mssjabri@uqu.edu.sa

* Correspondence: 2190000825@iau.edu.sa

Abstract: With the emergence of technology, the usage of IoT (Internet of Things) devices is said to be increasing in people's lives. Such devices can benefit the average individual, who does not necessarily have to have technical knowledge. The IoT can be found in home security and alarm systems, smart fridges, smart televisions, and more. Although small Internet-connected devices have numerous benefits and can help enhance people's efficiency, they also can pose a security threat. Malicious actors often attempt to find new ways to exploit and utilize certain resources, and IoT devices are a perfect candidate for such exploitation due to the huge volume of active devices. This is particularly true for Distributed Denial of Service (DDoS) attacks, which involve the exploitation of a massive number of devices, such as IoT devices, to act as bots and send fraudulent requests to services, thus obstructing them. To identify and detect whether such attacks have occurred or not in a network, there must be a reliable mechanism of detection based on adequate techniques. The most common technique for this purpose is artificial intelligence, which involves the use of Machine Learning (ML) and Deep Learning (DL) to help identify cyberattacks. ML models involve algorithms that use structured data to learn from, predict outcomes from, and identify patterns. The goal of this paper is to review selected studies and publications relevant to the topic of DDoS detection in IoT-based networks using machine-learning-relevant publications. It offers a wealth of references for academics looking to define or expand the scope of their research in this area.

Keywords: IoT; machine learning; DDoS attack; dataset; cyberattacks; attack detection; artificial intelligence; cybersecurity

check for updates

Citation: Alahmadi, A.A.; Aljabri, M.; Alhaidari, F.; Alharthi, D.J.; Rayani, G.E.; Marghalani, L.A.; Alotaibi, O.B.; Bajandouh, S.A. DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. *Electronics* **2023**, *12*, 3103. https://doi.org/10.3390/electronics12143103

Academic Editors: Ayush Goyal, Avdesh Mishra, Mais Nijim and David Hicks

Received: 15 May 2023
Revised: 30 June 2023
Accepted: 7 July 2023
Published: 17 July 2023

1. Introduction

The Internet of Things (IoT) refers to the network of interconnected physical devices that are embedded with sensors, software, and other technologies, enabling them to collect and exchange data. These devices span various domains, including the home automation, healthcare, transportation, and industrial sectors. For instance, smart thermostats, wearable fitness trackers, autonomous vehicles, and industrial sensors are all examples of IoT devices. Recent years have witnessed a runaway rise in IoT devices, which have expanded to everyday appliances and systems. It is estimated that IoT devices will reach a staggering number of 24.6 billion connected devices by 2025 [1]. With this astonishing number of IoT devices, there is a greater threat of cyberattacks, which can be induced by IoT devices, whether intentionally or by accident. Among the many attacks associated with IoT devices is a type of attack called DDoS attacks. DDoS attacks are growing at a disturbing speed and becoming more sophisticated. IoT devices are an integral part of many DDoS attacks, as they enable and improve the capability of DDoS attacks due to the connectivity of such devices to the Internet and the absence of firewalls and other security components in these devices. Cyberattacks can result in devastating incidents, such as power cuts, military equipment failures, and the leaking of confidential information. It is also possible that these attacks interrupt phone and computer networks, rendering data inaccessible or paralyzing systems. Moreover, IoT devices are more susceptible to being hijacked, as they lack numerous computational resources for adequate security. They can be exploited to conduct large-scale attacks without the knowledge of the device's owner, potentially leading to the creation of botnets. The connectivity among an increasing number of IoT devices corroborates the necessity for security measures to protect IoT infrastructure; this is becoming evident with the growing number of connected IoT devices, and this number increases daily. Therefore, addressing the severity of DDoS attacks in IoT-based networks is imperative to safeguard the integrity, reliability, and security of these interconnected systems, ensuring the smooth operation of essential services and protecting the interests of individuals and organizations alike. Thus, there has been a focus on increasing the level of security and protection against such attacks. A few approaches associated with increased cybersecurity include integrating ML techniques in identifying DDoS attacks over IoT devices. ML or DL models are essential for analyzing various datasets, particularly IoT, to enhance the ability to detect cyberattacks. The limited resources, such as the limited computational ability, related to IoT devices pose numerous limitations; thus, implementing other technologies to resolve these limitations is often necessary. Cloud computing, or fog computing, is the most convenient solution for tackling the restrictions of IoT devices. This paper investigates this aspect by reviewing the related work on IoT-based DDoS.

The following subsections explain each of the essential concepts discussed in this paper: each relevant element that constitutes an IoT device; the relationship between cloud computing and fog computing; why fog computing is taking over; and essential information regarding DDoS attacks and machine learning in cybersecurity. Subsequently, the research contribution section presents the relevant studies that have been thoroughly examined. Following this, the discussion section addresses our gap analysis based on our exhaustive readings. Finally, the conclusion section presents a summary and possible directions for future research.

1.1. Internet of Things and Fog Layer

As a concept, the IoT is the addition of intelligence to physical objects and things to generate, process, and exchange data. The goal underlying the IoT is to bring efficiency and provide complete control over our lives. Although objects are sufficiently smart to work without human intervention, they are controlled by their owners. IoT devices use sensors to collect massive amounts of data and reduce human data entry efforts. On the other hand, IoT devices are known for their limited computation abilities in terms of processing power and storage, thereby drawing attention to the cloud. The intersection of IoT devices and cloud computing is called the Cloud of Things (CoT), which helps overcome all the above-mentioned limitations. The cloud provides the services and infrastructure needed to power IoT devices. According to IoT analytics [2], there were approximately 12.3 billion active endpoints globally in 2021, a 9% increase in connected IoT devices. There will most certainly be approximately 27 billion IoT connections by 2025 [1]. Certain IoT devices hold time-sensitive applications, thereby implying that the data generated must be processed and analyzed on time. Cloud-centralized architecture suffers from latency, thereby affecting the traffic passing back and forth between IoT devices and the cloud. For this reason and more, as depicted in Figure 1, the fog layer is introduced to help resolve issues related to cloud architecture. As addressed by Cisco [3], fog computing can be considered an extended portion of the paradigm of cloud computing that brings it nearer to the network's edge. Fog computing has been defined and outlined by numerous organizations, researchers, and network experts from several distinct perspectives.

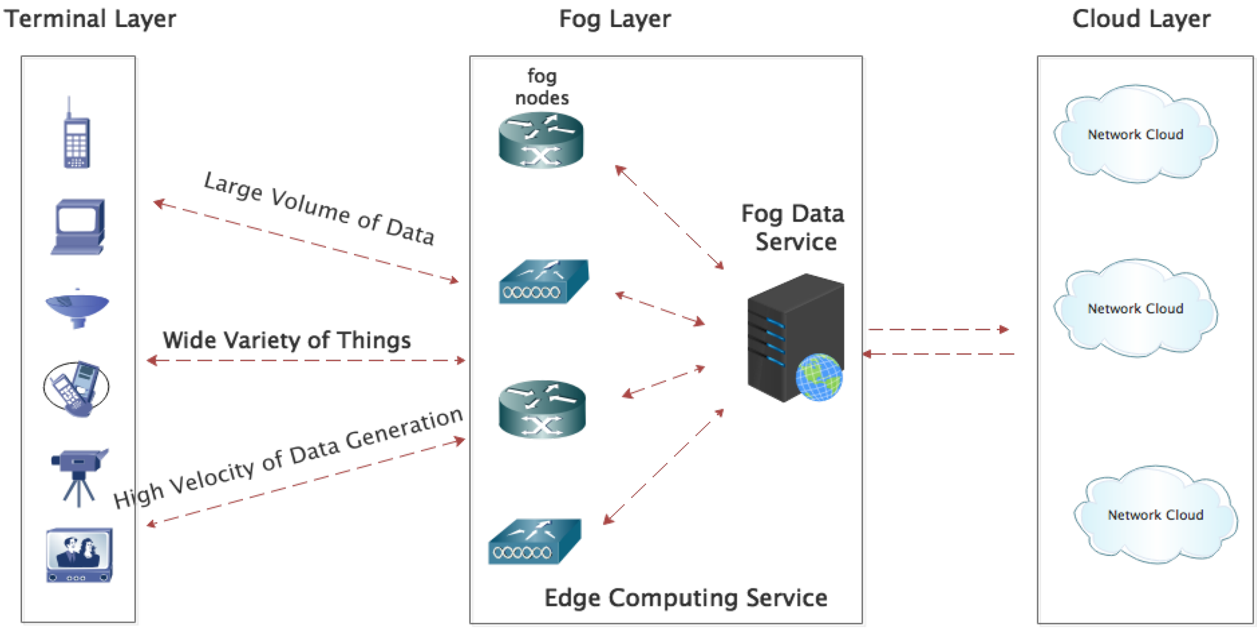

Figure 1. Fog Computing Architecture [4].

1.2. Distributed Denial of Service Attacks

DDoS attacks are one of the major risks to the security of IoT networks. In this attack, the attacker uses numerous compromised nodes to overwhelm the target by producing significant network traffic that consumes the target's resources. This eventually destroys the infrastructure, interrupts services, and prevents authorized users from accessing associated services. DDoS attacks employ two diverse types of techniques: reflection and amplification techniques. The attacker uses the reflection technique to send packets to several destinations while using the target's IP address as the packets' originating address. On the other hand, the attacker uses the amplification technique to send a large number of packets to the target's system [5]. DDoS attacks can be carried out in numerous ways, as summarized in Figure 2.

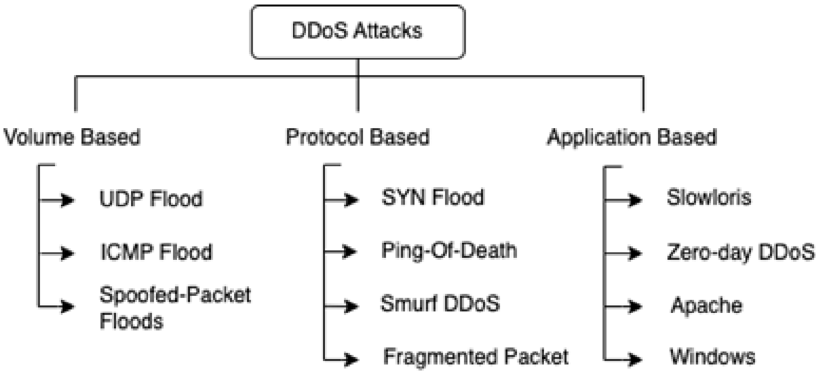

Figure 2. Taxonomy of DDoS Attacks.

1.2.1. Volume-Based Distributed Denial of Service Attacks

Volume-based attacks seek to render a system inaccessible by overloading the communication lines used to access the victim; therefore, compared to application layer assaults, resource exhaustion attacks, which are comprised of protocol exploitation or volume-based attacks, are far more expressive regarding the amount of traffic created during their execution [6]. The most frequent volume-based attacks take advantage of any unnecessary growth in packet size when using the UDP protocol. Amplification attacks, which ask

Internet servers to change the source address field to include the victim's IP address, are well-known instances of volume-based attacks. In essence, this results in the servers amplifying the responses, using up all of the target's bandwidth. Due to the high rates of response amplification, NTP and DNS servers are mostly utilized as enablers [7,8].

1.2.2. Protocol-Based Distributed Denial of Services Attacks

Protocol-based attacks, in contrast to volume-based attacks, target server resources rather than bandwidth. They also go after "intermediate communication equipment", which is another way of referring to firewalls and load balancers that stand between the server and the website. By sending bogus protocol requests to eat up the available resources, hackers overtax websites and these servers. The potency of these attacks is gauged in packets per second (pps). By taking advantage of flaws in protocols that are typically implemented at the network layer, attacks in this category attempt to exhaust hardware resources such as memory, CPU, and storage, thereby rendering servers inaccessible. Consequently, depletion assaults depend both on the combination of individual messages and the amount of traffic utilized [9]. Exploiting features of the TCP communication protocol is a key component of resource depletion attacks. The TCP SYN Flood is a generic form of this attack, which creates a three-way connection using the TCP protocol to use up all available capacity for managing connections (backlog). TCP SYN Flood involves forcing the target to constantly establish a fresh connection for the malicious client by delivering SYN signals to the victim while utilizing spoof source addresses [10]. The target server then awaits the client's approval to complete the step that involves establishing the connection, which never happens. Finally, it reduces the backlog and, consequently, prevents the opening of new connections.

1.2.3. Application-Based Distributed Denial of Service Attacks

Attacks on the application layer attempt to take advantage of flaws in a service or application that can lead to instability and make it impossible for authorized users to access the system. Since it takes only a small amount of malicious traffic to mimic the behavior of real consumers, these attacks are frequently misunderstood for technical errors. Consequently, most traditional detection systems fail to notice these threats [11]. The Slowloris is a frequent attack on the application layer [12]. In this scenario, partial requests are sent to a web server at regular intervals to maintain numerous connections for as long as possible and eventually cause the server to reach its connection limit. This indicates that the server stops accepting new connections shortly after the attack begins, at which point it stops being accessible to authorized users. The Slowloris attack uses the least amount of bandwidth possible to ensure its effectiveness, thereby making it undetectable by monitoring systems triggered by abnormalities in network traffic [13].

1.3. Machine Learning Models and Techniques

ML is a technique used to give computers the ability to automatically learn from previous data and make decisions like humans. ML involves training a model on training data and processing additional data to make a classification, detection, or prediction. This intelligent approach has begun to be implemented in network security, as it addresses the shortcomings of non-intelligent techniques. Moreover, ML or DL algorithms can be trained on network data to distinguish between benign and malicious network traffic, thereby protecting the networks from intruders. Additionally, if the network traffic is malicious, the algorithms can be trained to identify the type of attack and take the necessary actions to prevent the attack. These implementations of intelligent techniques, which are the concern of this study, have the potential to be beneficial in several cases. Moreover, there are two types of ML—supervised learning and unsupervised learning.

- Supervised learning is the commonly used type of ML in which a model is developed using a training dataset with predefined outputs. Once the model has been trained, it can predict the decision when new data are provided.

- Unsupervised learning utilizes the entered data solely without having an output corresponding to it. The objective of unsupervised learning is to identify possible hidden patterns or clusters in the data from unlabeled data.

Machine Learning Classifiers

In this subsection, a few supervised ML algorithms will be discussed. Decision tree, random forest, K-nearest neighbors, XGBoost, artificial neural networks, support vector machines, and adaptive boosting are discussed here, each with an informative brief. All these algorithms are considered within the top and most common ML model.

Decision Tree (DT)

DT is a hierarchical structure supervised algorithm in which the internal nodes represent the features of the dataset, the branches represent the decision rules, and the leaf nodes represent the outcome. The structure is illustrated in Figure 3. It classifies the instances beginning with the root downward up to the leaf nodes of the tree. DT deals with data inconsistency since all entities in a class have equivalent conditional probability values and fewer data cleansing requirements than other methods. The logic of DT can be easily understood and can mimic human thinking in decision-making [14].

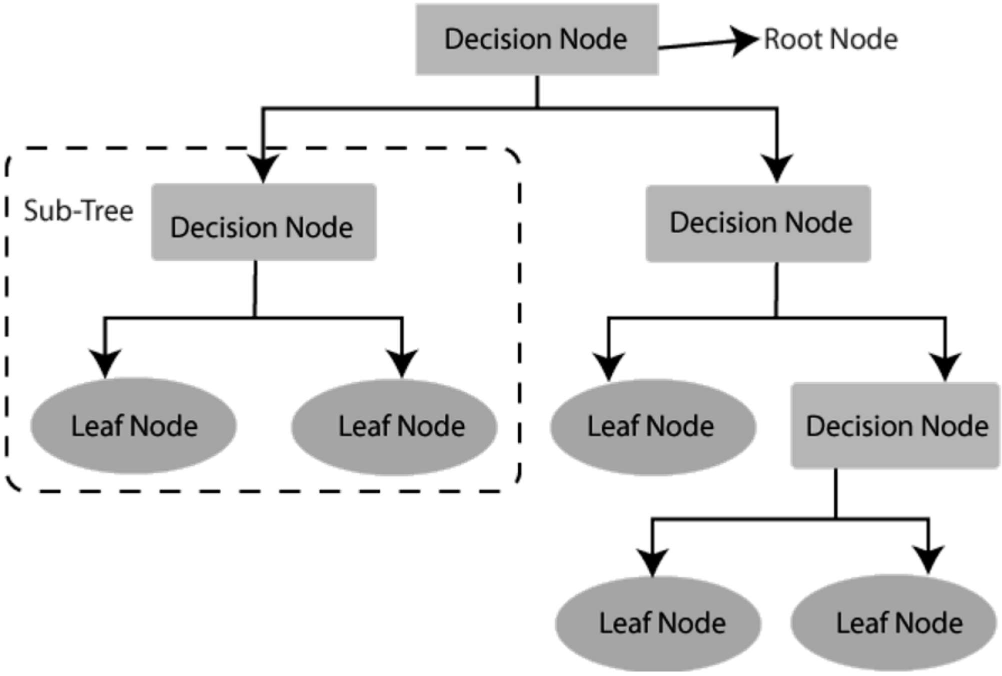

Figure 3. The Structure of the Decision Tree Algorithm [14].

Random Forest (RF)

RF is an ensemble-based supervised learning technique that merges multiple classifiers to tackle a challenging problem and improves the performance of models. RF takes less training time and maintains a high prediction accuracy even for large datasets and large missing proportions of the data [15,16]. Figure 4 illustrates the breakdown of RF that contains multiple decision trees for each subset of a dataset. To improve its predictive accuracy, RF aggregates the prediction outcomes of each tree to predict the outcome based on the most votes. Furthermore, to predict an accurate result, there must be some actual values in the dataset's feature variable, as well as a greater number of trees.

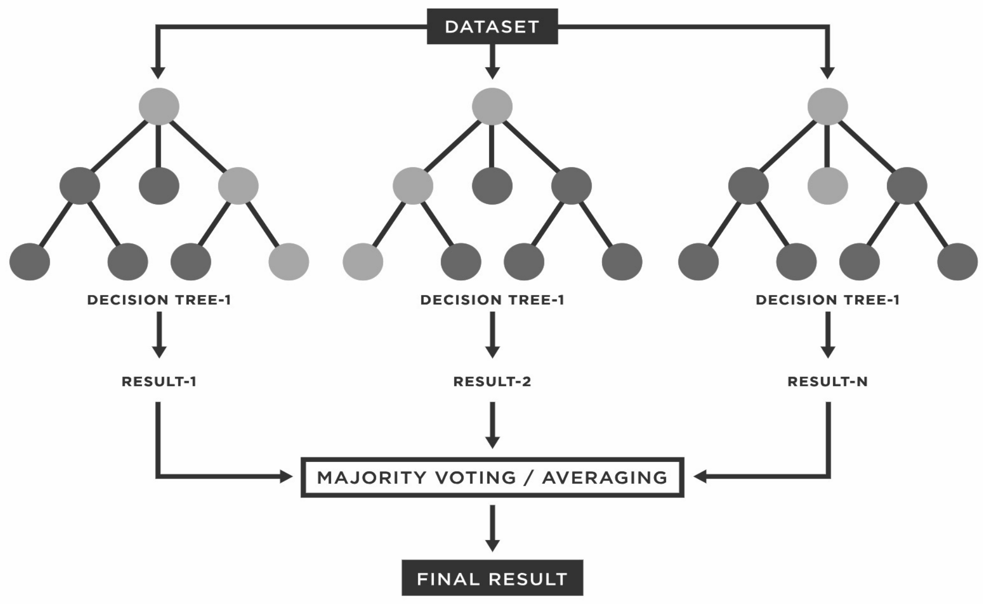

Figure 4. Random Forest Algorithm Structure [17].

K-Nearest Neighbors (KNN)

KNN is a supervised, straightforward machine learning algorithm. KNN does not learn immediately from the training dataset. On the contrary, it works by storing the dataset and assuming the similarity between new and existing cases, then placing the newly identified case in the category that is most similar to the existing ones. KNN is resistant to noisy training data. However, it entails a high computational cost, as it makes its prediction based on a distance calculation using an enhanced distance algorithm [18].

XGBoost (eXtreme Gradient Boosting)

XGBoost is a gradient-boosted tree-structured implementation based on sequential enabling. Gradient descent refers to the underlying objective function; it provides substantial flexibility while delivering the desired results using computational power optimally. XGBoost can handle sparse data, parallel processing, and built-in cross-validation to reduce overfitting [19–21].

Artificial Neural Network (ANN)

ANN, or neural network (NN), architectures mimic the network of neurons and are derived from biological NNs that build the structure of the human brain so that computers may grasp things and make choices in a human-like manner. As depicted in Figure 5, ANNs comprise three layers: input, hidden, and output layers. The hidden layer lies between the input and output layers and can perform all the necessary calculations to find hidden patterns and features. A few advantages of ANNs are that they have a parallel processing capability, work with incomplete knowledge, and have fault tolerance. On the other hand, ANNs have hardware dependence and require assurance that the network is appropriately structured [22].

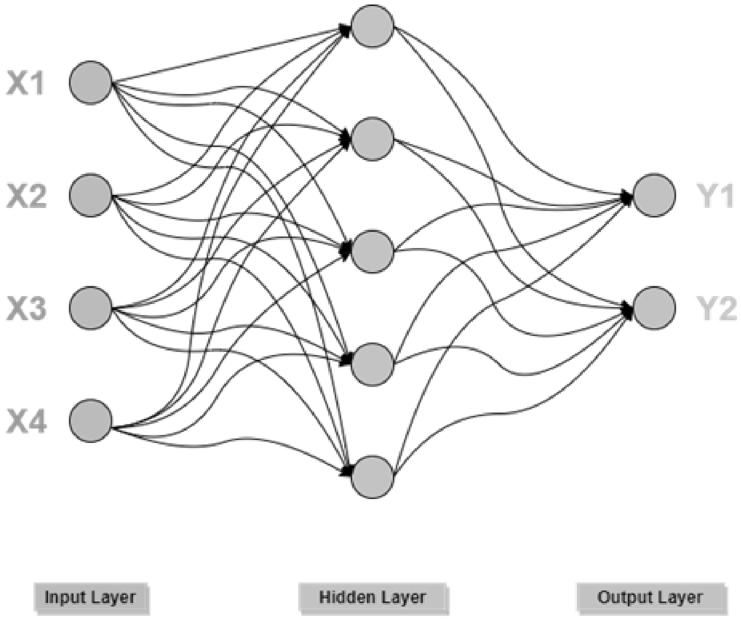

Figure 5. Artificial Neural Network Algorithm Structure [23].

Support Vector Machine (SVM)

A SVM is a supervised learning algorithm that is widely used for classification. The SVM algorithm attempts to locate a decision boundary that differentiates the two classes in the SVM, which is also known as a hyperplane, in an N-dimensional space for each distinct dimension (with N being the number of features). SVM memory is efficient, as it utilizes a subset of the support vectors, which are the training points of the decision function. As depicted in Figure 6, SVMs can be of two types: linear SVMs and non-linear SVMs [24]:

(A) Linear SVM: This is utilized for linearly separable data, which implies that if a dataset using a single straight line can be classified into two classes that are linearly separable, the classifier employed is called a linear SVM.

(B) Non-linear SVM: This is utilized for non-linearly separable data, which implies that if a dataset utilizing one straight line cannot be categorized, it is called non-linear data, while the classifier employed is called a non-linear SVM.

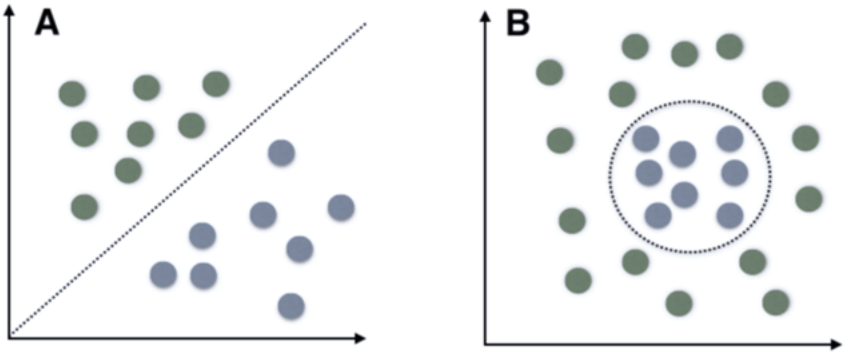

Figure 6. (**A**): Linear SVM; (**B**): Non-linear SVM [25].

Adaptive Boosting (AdaBoost)

AdaBoost is a supervised learning boosting technique based on the ensemble iterative method. It combines multiple low-accuracy classifiers to build a single highly accurate classifier. The objective of AdaBoost is to train the data sample and set the classifier weights in each iteration to produce accurate predictions of anomalous observations. AdaBoost repeatedly adjusts the errors of the weak classifier. However, it is highly affected by noisy data [26–28].

1.4. Machine Learning Performance Evaluation

The following section discusses the commonly employed metrics and criteria for evaluating the performance of machine learning models in DDoS attack detection.

- Accuracy: Accuracy is a widely used metric that measures the overall correctness of the model's predictions. It represents the proportion of correctly classified instances (i.e., both true positives and true negatives) to the total number of instances. A high accuracy indicates a reliable model, but it may not provide a comprehensive view of the model's performance due to class imbalance or other factors.
- Precision: Precision is a metric that quantifies the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (both true positives and false positives). It measures the model's ability to avoid false positives. A higher precision indicates a lower rate of false alarms.
- Recall: Recall, also known as sensitivity or the true-positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives and false negatives). It quantifies the model's ability to identify all positive instances. A higher recall indicates a lower rate of false negatives.
- F1-Score: The F1-score is a harmonic mean of precision and recall and provides a balanced measure of a model's performance. It considers both the precision and recall values and is useful when the data has a class imbalance. A higher F1-Score indicates a better trade-off between precision and recall.
- Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's predictions by showing the number of true positives, true negatives, false positives, and false negatives. It provides insights into the types and frequency of misclassifications, enabling a deeper analysis of model performance.
- Computational Resources: Apart from traditional performance metrics, it is essential to consider the computational resources required by the model. Assessing factors such as training time, memory usage, and inference speed can help determine the practicality and scalability of the model in real-world deployment scenarios.

By considering these metrics and criteria, researchers can evaluate the performance of machine learning models for DDoS attack detection in IoT-based networks comprehensively. Understanding these evaluation measures enables researchers and practitioners to make informed decisions regarding model selection, comparison, and adaptation in response to emerging threats in the dynamic landscape.

2. Research Contribution

To guide our literature search, we developed three research questions:

1. What are the current approaches for detecting DDoS attacks in IoT-based networks using machine learning models?
2. What are the limitations of existing approaches?
3. What are the research directions for improving DDoS attack detection in IoT-based networks using machine learning models?

The selection criteria for studies included in the review paper were defined based on the research objectives and focus. The following explicit criteria guided the selection process:

1. Keyword search: The initial search was conducted using keywords such as "DDoS attack", "Internet of Things", "IoT-based networks", "machine learning", and "DDoS detection" to identify relevant papers in the literature.
2. Time frame: Studies published between 2016 and 2022 were considered to ensure the inclusion of recent research and advancements in the field.

3. Research objective: The selected studies were required to focus on DDoS attack detection in IoT-based networks using machine learning models. Only papers directly addressing this topic were included.
4. Investigation of machine learning models: The studies needed to investigate the effectiveness of machine learning models for DDoS attack detection in IoT-based networks. Papers exploring other detection methods or not employing machine learning were excluded.
5. Dataset analysis: Papers related to datasets generated for DDoS attack detection in IoT-based networks were included to explore their characteristics and shortcomings. Studies that discussed dataset properties, size, diversity, and relevance to real-world scenarios were given priority.

By following these explicit criteria, a comprehensive selection of studies was made to review the effectiveness of machine learning models, investigate feature selection, analyze performance metrics, and assess the characteristics of datasets used for DDoS attack detection in IoT-based networks. The following subsections review the main points and characteristics of each study.

2.1. Generated Distributed Denial of Service Attacks on Traffic Datasets

This subsection discusses the numerous studies concerned with datasets used in detecting DDoS attacks and datasets constructed in an IoT environment. The main common factor between these studies is the involvement of both old and modern datasets that contribute to the detection and, occasionally, mitigation of DDoS attacks. A few of these studies used existing and open-source datasets, and others built their own datasets. Some of the datasets mentioned in this section include IoT traffic, while others do not. Furthermore, some of these datasets were built in an actual setup, a synthetic setup, or a combination of both. Datasets are a common factor amongst these mentioned studies and reflect real traffic and, therefore, achieve a realistic and accurate detection of attacks.

We begin with Almaraz-Rivera et al. [29], who utilized the Bot-IoT dataset published in 2019 [30]. This dataset was constructed by simulating IoT devices that mimicked a smart home setup and other network devices. The Bot-IoT dataset considers various cyberattacks commonly used by botnets, including information gathering (OS fingerprinting and service scanning), DoS/DDoS (UDP, TCP, and HTTP), and information theft (data theft and keylogging). To execute several botnet scenarios, four Linux-based virtual machines (VMs) were used to launch the attacks simultaneously. Using the Node-Red tool, five smart devices were simulated and connected to the associated cloud infrastructure to produce benign traffic. Moreover, the Ostinato tool was utilized to produce a large volume of benign traffic with specified IPs and ports. In addition, the tshark tool was used to collect all benign and attack traffic transferred throughout the network. Thereafter, the features were extracted using the Argus tool, and then new additional features were created based on the features extracted from the Argus tool to improve the prediction of classifiers. However, class imbalance is a severe issue in Bot-IoT.

This bias problem was addressed by Almaraz-Rivera et al. [29], who developed a novel intrusion detection system (IDS) that leverages artificial intelligence (AI) models—that is, ML and DL—that focus on DDoS and DoS attacks. The authors of the dataset utilized the Argus tool data generator, through which the network traffic and features were produced. Moreover, they selected three feature sets to assess how the record timestamps affect prediction and prevent dependencies. The maximum, minimum, and mean are statistical variables that exist in all feature sets; on the contrary, the first feature set was the only one to involve timestamps. The Argus sequence number was not included in all feature sets; it was only included in the first and third feature sets. Furthermore, for training and testing, 18 of the most relevant features found in the feature sets were used. The results revealed that DT and RF have the best performance, with the DT method not being reliant on one feature, thereby exhibiting robust performance.

In another study, Liu et al. [31] developed their application through Android Studio. The application they developed had the objective of initializing smart sensors for the sake of capturing environmental data and sending it to a cloud-based database. A web server installed with Wireshark listened to the network traffic going through the smart devices. In addition to using two Raspberry Pi devices that ran an open-source operating system made for Android Things, they represented a general IoT device. Data were collected using Rainbow HAT sensor boards installed on Raspberry Pi devices. The data they generated was from both physical and virtual devices. Further, the majority of published datasets were created in a virtual environment, yet the datasets have been used to build upon network security solutions for scenarios ranging from smart homes to smart cities. CCD-INID-V1 is the dataset created in [31], which contains IoT network traffic collected through Wireshark or tcpdump. The most frequently found attacks on IoT networks were used in the dataset, including address resolution protocol (ARP) poisoning, ARP DoS, SlowLoris, Hydra, and UDP flood DoS. Study [31] combined RF and XGBoost embedded feature selection methods with a convolutional neural network (CNN) to create a lightweight hybrid model for detecting anomalies and classifying multi-attacks. Along with the NFStream python library to engineer the features on a CCD-INID-V1 dataset, 83 features were extracted, including source and destination addresses for both MAC and IP addresses. The purpose of the dataset they generated is to test the validity of their proposed method, and it was found that it can be a helpful classification and detection method. Using signature-based detection tools or systems discourages IoT DDoS attacks, as they cannot detect fresh attacks since their signature must be identified first and fed into the decision system. Due to the intricacy that hackers implement in their attacks, anomaly-based detection is considered appropriate.

Therefore, Ullah and Mahmoud [32] attempted to replicate the modern trend of IoT network traffic using a typical environment for smart homes. This environment consisted of SKT NGU and EZVIZ Wi-Fi cameras, which were used to produce the IoTID20 dataset. They were connected to a Wi-Fi router. Other devices connected to the router were laptops, tablets, and smartphones. However, the SKT NGU and EZVIZ Wi-Fi cameras were the victim devices, while the remainder were attacking devices. The traffic generated on the network was collected in the IoTID20 dataset, which consists of 83 features and various categories and sub-categories according to the type of attack that occurred. Mirai brute force, Mirai ACK/HTTP/UDP flooding, man-in-the-middle (MITM) scan host port, scan port OS, and DOS are the eight types of attacks present in the dataset. In addition, the following ML algorithms were used to analyze the performance of the dataset: SVM, Gaussian NB, LDA, logistic regression, DT, RF, and ensemble. The dataset's design and goal are to provide a basis for developing technologies to detect abnormal and malicious activities in IoT networks.

The lackluster security measures found in IoT-based ecosystems can be a significant reason for the demise of enterprises that incorporate IoT devices. Alsaed et al. [33] proposed a new IoT dataset known as TON_IoT that incorporated telemetry data and logs from operating systems and IoT network traffic. The authors of the papers prepared a testbed consisting of seven IoT and IIoT sensors to capture telemetry data. The testbed was developed to connect numerous virtual machines, physical systems, hacking platforms, cloud computing, fog computing, and IoT and IIoT sensors to mimic how complex and scalable the industrial IoT and the industrial networks are. What differentiates this dataset from all other existing ones is the variety of attacks and regular events, as well as the heterogeneity of data sources. Moreover, the labels in the dataset indicate whether the instance is regular traffic or an attack, while the type showcases sub-categories if the record is that of an attack instance. The performance analysis methods conducted on the dataset include accuracy, F-score, recall, and precision. Study [33] also considered training and testing time when evaluating their model on multiple open-source datasets, along with the TON_IoT dataset. RF, Naïve Bayes, SVM, LR, KNN, linear discriminant analysis (LDA), classification and regression trees (CART), and long short-term memory (LTSM) were the models utilized to test and train the dataset. This dataset was separated into seven datasets to acquire the best results when implementing the previous ML models, such as the fridge sensor, garage door, GPS sensor, Modbus, light motion, thermostat, and weather datasets. LR and SVM performed the worst on most datasets, as they showed an abundant number of false positives and took the most extended testing and training. In contrast, CART, RF, and KNN performed the best with the least time and the highest F-score, recall, and accuracy values. This dataset has been carefully examined by eliminating flow identifier attributes to avoid certain issues such as bias toward attacks and overfitting. Nevertheless, the TON_IoT dataset suffers from class imbalance, categorical and irrelevant features, and missing value issues.

The CICIDS2017 dataset in [34] attracted researchers who were interested in the analysis and development of ML models and algorithms. The dataset took over eight PCAP files that contained five days' worth of average and attack-filled traffic data from the Canadian Institute of Cybersecurity. CICIDS2017 contained benign traffic and the most recent common attacks, which replicated accurate world data. It included network traffic analysis results that used CICFlowMeter software, with features centered on the time stamp, source, and destination IP addresses, along with ports, protocols, and attack CSV files. To appropriately profile the behavior of humans with IoT devices and generate naturalistic benign background traffic, study [34] assessed the interactions that were based on the abstract behaviors of 25 distinct users extracted from HTTP, HTTPS, FTP, SSH, and email protocols. A methodology the authors used was relabeling classes and creating more classes by splitting the majority classes or forming a class by merging a few minority classes, thereby solving class imbalance issues by reducing the imbalance and improving the prevalence ratio [35].

Finally, Gopi et al. [36] presented a model for detecting DDoS attacks in multimedia IoT using a dimension-reduced ANN ML algorithm. The project was implemented on the Windows XP operating system using the KDD Cup99 dataset, which resulted from the network traffic dataset called DARPA98 and was created by accumulating each TCP packet into TCP connections. Every TCP packet comprises 60 characteristics and a label or tag to determine whether the status of the connection is benign or malicious. The dataset was used with 60,000 connections, including 20,000 typical connections, TCP SYN flood attack connections, and UDP flood attack connections. The system also applied the particle swarm optimization (PSO) paradigm, which is dedicated to training an ANN to configure the classifier for anomalies. An approach based on ANN was compared with an approach based on PSO, and the results exhibited that the ANN design scored higher in terms of accuracy and detection and had a low false-alarm rate.

2.2. Machine Learning Methods for Distributed Denial of Service Detection

In this subsection, we studied several papers on both ML and DL artificial techniques that were used to build or compare models to detect and classify DDoS attacks using common enriching and private datasets.

Amrish et al. [37] presented an ML methodology that classified normal and DDoS traffic to determine which classification model performed best in detecting malicious IP addresses. For training and testing purposes for the classification models, the CICDDoS2019 dataset was used. This dataset is comprised of several instances of DDoS attacks with 1 class attribute and 88 features, out of which the best 15 features were extracted. This work was evaluated using four algorithms: ANN, KNN, RF, and DT. ANN was concluded to be the best-performing model, which had an accuracy score of 99.95%. False positives were non-existent, while false negatives were more common.

In contrast, Gaur and Kumar [38] proposed a hybrid methodology calculated using the CICDDoS2019 dataset that integrated ML and feature selection (FS) algorithms to reduce the time needed for training and improve system performance. For the early identification of DDoS attacks on IoT devices, feature selection methods, chi-squared tests, analysis of variance (ANOVA), and extra tree were applied to four ML classifiers: RF, KNN, DT, and XGBoost. Initially, the accuracy of the classifiers with full features was recorded along with the performance of the classifiers without any feature selection. XGBoost exhibited the highest accuracy regardless of the applied feature selection algorithms. Extensive iterations of feature selection methods were performed, and essential features were selected to reduce data dimensionality. Consequently, XGBoost coupled with the ANOVA FS method attained the highest accuracy of 98.34% with 15 features. The tuning of hyper-parameters using different tuning methods, including grid search, Bayesian optimization, and random search, was presented as future work to further improve the performance parameters since the optimization of hyper-parameters deals with the issues of overfitting and underfitting.

Similarly, the XGBoost algorithm was the highest performing algorithm in [39], which addressed kernel hyperparameter tuning optimization to improve model efficiency. In this study, ML classifiers with model optimization were proposed for DDoS attack prediction and types of classification. Therefore, two powerful supervised ML models, RF and XGBoost, were used to address the classification problem—that is, intrusion detection. For model optimization, data mining techniques were applied to enhance the data quality since the researchers used old datasets. The proposed framework utilized an old KDD dataset, UNWS-NB15 (100 GB), provided by the Australian Centre for Cyber Security (ACCS). UNWS-NB15 covers 9 modern types of attacks with 49 attributes. Based on a generated confusion matrix that identified the model's performance, the first classification achieved an average accuracy (AC) of approximately 89%, while the second classification was approximately 90%. The authors emphasized the importance of using the latest datasets to detect DDoS attacks using ML models.

Additionally, XGBoost produced the highest accuracy in [40], which proposed data mining and an ML-based DDoS detection model. Using the CICDDoS2019 dataset, the top 10 features with the highest impact on the detection of DDoS attacks were identified to minimize overfitting the model, increase its speed, and enhance the accuracy and training of the detection model. The top 10 features were "Flow ID", "Source and Destination IP", "Source Port", "Timestamp", "Bwd Packet Length Max", "Bwd Packet Length Min", "Total Backward Packets", "Total Length of Fwd Packets", and "Fwd Packet Length Max". Further, different ML algorithms were implemented and tested, such as SVM, naïve Bayes, XGBoost, AdaBoost, KNN, and RF. Both AdaBoost and XGBoost performed relatively well, with accuracies of 100%. However, XGBoost showed slightly better training and detection time.

The DT algorithm, J48, resulted in the highest accuracy in [41,42] with a generated dataset. Study [42] proposed an ML methodology to categorize the different kinds of DDoS attacks. The proposed methodology was affirmed using the dataset constructed in [43], which contained several types of modern attacks, including HTTP, UDP, SIDDoS, and Smurf. Moreover, the dataset was comprised of 27 features and 5 different classes. An ML tool called WEKA was used for identifying DDoS attacks. Four classifiers were implemented: naïve Bayes, RF, multilayer perception (MLP), and J48. For each class, recall and precision were calculated. Among the four algorithms, J48 outperformed other ML algorithms with 98.64% accuracy. Moreover, MLP achieved an accuracy of 98.63%; however, J48 showed less time complexity as compared to MLP.

Additionally, J-48 was tested by Aysa et al. [41]. The proposed framework aimed to detect DDoS on IoT devices. The authors emulated DDoS attacks (Mirai and BASHLITE) to build a standard dataset extracted from three distinct Wi-Fi IoT camera devices: Philips (infant screen), Equipment (security camera), and Simple Home (Safety Camera). The data were collected prior to and after the infection of the two types of attacks. The dataset obtained 40 selected features using the Pearson coefficient technique. The attack was examined five times at intervals of 0.01, 0.1, 1, 3, and 5 s. J-48, linear support vector machine (LSVM), neural network (NN), and RF ML algorithms were utilized. However, it was found that RF coupled with DT achieved the highest accuracy. Similarly, RF outperformed other models in [44–46]. Pande et al. [44] built a model that detected DDoS attacks using ML. The attack was launched using the techniques of the ping of death, and the detection occurred through ML using WEKA tool version 3.8. The dataset used in the experiment was NSL-

KDD, which contained 22,544 training instances and 42 attributes. The algorithm used was RF, which resulted in 99.76% accuracy. The model's building time was 8.71 s, but the testing time was 1.28 s. The authors aim to implement DL techniques, for example, classification.

Gupta et al. [45] analyzed features of IoT networks to suggest a machine learning-based method for detecting DDoS-infested traffic in consumer IoT (CIoT). This study implied that the traffic patterns of IoT devices differ in specific fundamental ways from those of traditional Internet-connected devices. Less processing overhead and less resource utilization were the outcomes of the proposed method, as it operated on a local router. The authors simulated an IoT network to capture the ordinary and attack traffic patterns and create a dataset. Based on the obtained dataset, six ML models were used for detection: the NB, DT, SVM, RF, KNN, and logistics regression (LR) algorithms. As an additional mitigation precaution, the local router filtered out malicious traffic. The accuracy of the experimental findings fell in the range of 0.97 and 0.99. However, RF was the classifier with the highest performance, and it was also the most accurate and reliable for attack detection. It obtained a precision value of 0.967, a recall value of 0.989, and an F-measure value of 0.97. Additionally, it had a low false-positive rate of 0.008, thereby indicating that it generated few false alarms.

Mihoub et al. [46] presented an ML technique for IoT DoS/DDoS attack detection and mitigation. A multi-class classifier, which utilized the looking-back methodology, was tested on a Bot-IoT dataset and made use of the looking-back methodology. Both primary and looking-back techniques were evaluated using ML algorithms, including RF, KNN, and DT, and DL models, such as MLP, LSTM, and recurrent neural networks (RNN). This research attempted to classify DoS/DDoS attacks into subcategories and, consequently, classify the attacks and the associated mitigation techniques. Six subcategories of attacks (DoS-TCP, DoS-HTTP, DoS-UDP, DDoS-TCP, DDoS-UDP, and DDoS-HTTP) are constituted of three diverse types of packets and two attack categories. The detection identifies the type of attack and the packet it employs. The associated mitigation countermeasure can be applied to specified packets. The KNN algorithm, which does not use a looking-back technique, had the highest accuracy of 99.93%, but it also appeared to take the longest time to train and test. One of the models that provided the best balance between training and testing periods was RF, which, when used with the looking-back-enabled approach, obtained an accuracy of 99.81%.

Pokhrel et al. [47] employed an ML method to establish a methodology for detecting and mitigating DDoS botnet-based attacks in IoT networks. To decrease system processing overhead, this study demonstrated that feature reduction is feasible with relatively little influence on accuracy. A model was trained using the Bot-IoT dataset and the naive Bayes, KNN, and multi-layer perception artificial neural network (MLP ANN) algorithms. The classes of this dataset were rather unbalanced; they were balanced using the synthetic minority oversampling technique (SMOTE). All ML methods underwent data training utilizing unbalanced and class-balanced datasets, and the influence of the imbalanced dataset on the algorithms was examined to identify the best algorithms. Although the use of an unbalanced dataset yielded a high accuracy result, the recall and F1-score could have been better, which indicated that the accuracy achieved from the imbalanced dataset may be deceiving.

Multiple ML models were tested for DDoS detection in [48], which presented a pipeline for detecting a DDoS attack. The pipeline was intended to collect data, extract features, and classify IoT network traffic for DDoS attack detection. The three most popular DDoS attack types—a UDP flood, a TCP SYN flood, and an HTTP GET flood—were all reproduced. Furthermore, multiple classifiers were compared to detect an attack, including RF, KNN, SVM, DT, and NN. To generate training data on classifiers, a local network was established that consisted of a router and standard IoT consumer devices. The pipeline was designed to function on middleboxes to identify anomalies that corresponded to a specific device. Experimental results revealed that all five algorithms obtained an accuracy higher than 0.99.

Similarly, Islam et al. [49] proposed a framework for detecting DDoS attacks on financial organizations by comparing SVM, KNN, and RF ML models on open-source banking datasets on the Kaggle platform. The homogeneity measure (k-means clustering) was used to extract essential features, and the five-fold cross-validation technique was utilized to improve model performance and prediction. Each ML algorithm was implemented and tested on a distinct dataset, and each model performed with 99.5%, 97.5%, and 98.74% accuracy in detecting DDoS attacks. According to the findings, SVM showed robust performance. However, the suggested framework had a few drawbacks, since the training phase required significant computing power with specialized hardware, such as a GPU, and the detection was limited to offline datasets.

In addition, SVM was tested by Roopak et al. [50], who proposed a DL technique to classify attacks. A jumping gene-adapted non-dominated sorting genetic algorithm (NSGA-II-aJG) was used to select features after feeding it with normalized preprocessed network data containing both normal and malicious traffic. The proposed algorithm, NSGA-II-aJG, had six objectives: maximize accuracy, minimize redundancy, maximize relevance, minimize the number of features, maximize recall, and maximize precision. The algorithm was evaluated using the modified CICIDS2017 dataset, and the evaluation was performed with a high-performance computer (HPC). The results revealed that NSGA-II-aJG outperformed the remaining methods, accomplishing 99.03% accuracy; in comparison, MLP obtained 88.74% accuracy, SVM algorithm achieved 94.50% accuracy, Bayes attained 94.19%, and RF achieved 93.64% accuracy. A suggestion for future research would be to implement this proposed work in the fog-node model.

The empirical study in [51] proposed a voting-based multimode ML framework (VMFCVD) to detect and protect against volumetric DDoS attacks. The study mentioned that volumetric DDoS attacks are the most common type of attack. The framework was tested and evaluated through extensive experiments using and comparing different botnet and DDoS attack datasets. Comparisons were made using a couple of datasets, namely CICIDS2017 and CSE-CIC-IDS2018, both of which contained botnet and DDoS attacks. CICDDoS2019 included attacks that are carried out using DNS, LDAP, SSDP, and SYN; CIRA-CIC-DoHBrw-2020 had two layers for capturing benign and malicious DoH traffic along with non-DoH traffic; NBaIoT2018 and UNSW2018 Bot-IoT could both detect IoT botnet attacks (e.g., Mirai); and UNSW NB15 was a network intrusion dataset that contained nine different attacks. The VMFCVD framework was based on a triad of modes, which are described below:

1. Fast detection mode (FDM): This mode classified network traffic when the server was under attack.
2. Defensive fast detection mode (DFDM): This mode was an extensive version of FDM that tightened the detection technique to enhance the detection of malicious network traffic.
3. High-accuracy mode (HAM): This mode was activated when the server was stable.

The proposed framework was compared with various ML algorithms and state-of-the-art baselines. The accuracy rate of each mode of the framework (i.e., FDM, DFDM, and HAM) was compared with the accuracy rates of ML algorithms (i.e., AdaBoost, Bagging, GB, KNN, and RF) on the previously mentioned datasets. The experiment's results revealed that the VMFCVD framework, on most occasions, outperformed ML algorithms in classification accuracy. The experiments also revealed that when the framework was operating in HAM mode, it outperformed all different modes of VMFCVD as well as ML models and obtained an accuracy of 100% while incorporated with the UNSW2018 Bot-IoT. The accuracy of ML algorithms was also compared with the average accuracy rate of VMFCVD, which revealed that the accuracy of ML models ranged from 43% to 100%, whereas VMFCVD always maintained an accuracy of above 98.7%, with an average accuracy of 99.82%. Regarding future research, the study mentioned the possibility of broadening the scope of the framework by creating more generic DDoS and botnet datasets and running the framework on live servers, in addition to expanding the capability of the framework to further block the devices in case they are identified as responsible for generating multiple malicious attacks.

2.3. Fog Layer Distributed Denial of Services Detection for an Internet-of-Things-Based Network

In this subsection, papers that highlight the intersection of the fog layer and IoT devices that serve in the detection of DDoS will be introduced.

Within the subject of fog layer detection, reference [52] contributed to the usefulness of the three-layer architecture in IoT-based networks, the model's identification of attacks based on protocol, and the importance of several factors in identifying anomalous IoT traffic. The model's first stage involved data preprocessing, with feature extraction from packets depending on the state of the network. The second stage was a principal component analysis (PCA), in which all characteristics were converted into principal components. Finally, anomaly detection in the dataset occurred using a statistical metric named continuous ranked probability score (CRPS). This model successfully detected both ICMP and TCP-SYN attacks within the dataset. Going deeper into fog computing, DDoS detection was implemented in the fog layer in [53]. The two primary sections of the approach used in the proposed study were the layer that contains IoT devices and the fog layer. Raspberry Pi 400 with 4GB RAM was used to simulate IoT devices, and the irregular data collected from devices needed to first be transformed into regular data. Studying the unpredictability of network traffic using entropy is essential for spotting DDoS assaults. Entropy groups of packets were calculated and resulted in either 0 for total random packets or 1 for identical packets. Then, the progress reached the classification step; the training dataset used was the CIC-DDoS2019 benchmark along with the KNN algorithm. This model of work resulted in 100% accuracy for both UDP and ICMP DDoS attacks and 98.79% accuracy for TCP DDoS attacks. For future work, the authors intend to identify new DDoS attacks or achieve higher detection rates.

3. Discussion and Gap Analysis

Despite examining the related research that treats the main concept of this paper differently, a research gap remains. This section includes a gap analysis, a table summarizing the studies, and another table that compares the datasets we have covered.

Many researchers have worked on detecting DDoS attacks on IoT-based networks using ML; Table 1 presents a summary of these studies. After reviewing the research based on three levels, that is, the dataset level, the machine learning level, and the fog layer, the results revealed a few limitations in the literature. Most of the research aimed to detect a DDoS attack on old datasets such as Bot-IoT, CICDDoS2019, UNSW-NB15, NSL-KDD, and CCD-INID-V. The majority of published IoT datasets were designed to validate IoT-network-based IDSs. Nevertheless, the aim of our research was to address the current state of DDoS attack severity and the complexity of IoT devices, as well as address the shortcomings of the existing dataset. Table 2 illustrates the comparison between the existing datasets used to detect DDoS attacks in IoT networks and presents the shortcomings and limitations of the existing datasets, as it becomes a necessity to update those datasets to reflect current novel attack techniques. The testbed employed in certain scenarios of datasets was unrealistic, while in others, the attack scenarios were not diverse. In addition, multiple benchmark datasets were not generated in an IoT context, such as CIC-IDS2017, CICDDoS2019, UNSW-NB15, and NSL-KDD. Further, multiple datasets, including Bot-IoT, TON_IoT, and IoTID20, suffer from class imbalance, which may lead to poor accuracy and/or biased data, thereby affecting the identification of the attacks. In most of the research, the Bot-IoT dataset was used. The Bot-IoT dataset contains realistic malicious and benign traffic (amounting to a total of seventy-two million records) with a multiclass classification hereby being heavily used. However, Bot-IoT suffers from an intense class imbalance, with just a few thousand (approximately 9000) benign flows; this creates a major contrast in this dataset, as it contains malicious traffic of over 99%, while benign traffic is less than 1%. Similarly, the IoTID20 dataset with intrusion activity records 15 times the normal traffic. Thus, including a more benign flow maintains and enhances the realism of the dataset. Additionally, none of the DDoS attack types were considered in the CCD-INID-V1, UNSW-NB15, and NSL-KDD (KDD Cup99) datasets, although these were used for DDoS detection. In contrast, a limited number of DDoS attack types was covered in Bot-IoT (TCP, UDP, and HTTP), IoTID20 (Mirai ACK Flooding, Mirai Brute force, Mirai HTTP Flooding, and Mirai UDP Flooding), and N-BaIoT2018. Therefore, the direction for innovative research often incorporates an element of dataset generation along with a pioneering framework or methodology to solve a prominent problem. For detection, multiple ML algorithms were investigated in numerous studies, including SVM, KNN, RF, DT, Naïve Bayes, and ANN. However, the ML algorithms XGBoost and AdaBoost were not adequately investigated in the detection of DDoS attacks in IoT networks. In the era of AI and ML, the XGBoost classifier is known to be the queen by scientific and academic researchers as it is considered a weapon for and considered reliable for big data utilization, in addition to possessing notable speed, efficiency, scalability, and simplicity.

Advancements in machine learning techniques can enhance the accuracy of DDoS attack detection in IoT-based networks. Advanced algorithms, such as deep learning and ensemble methods, can effectively learn complex patterns and detect subtle anomalies, leading to more precise and reliable detection outcomes, while the advancements in IoT technology, including edge computing and fog computing, enable real-time and scalable monitoring solutions. By deploying machine learning models directly on edge devices or fog nodes, network traffic can be analyzed locally, reducing latency and enabling faster response times. The use of federated learning allows machine learning models to be trained collaboratively without sharing sensitive data. This decentralized approach addresses privacy concerns in IoT networks, enabling the development of effective DDoS attack detection models while preserving privacy. Additionally, by combining machine learning techniques with contextual analysis and IoT device profiling, detection capabilities can be enhanced. Contextual analysis considers various factors such as device behavior and network interactions, while device profiling creates device-specific profiles, enabling the identification of anomalous behavior associated with DDoS attacks. However, implementing advanced machine learning techniques in IoT-based networks may require significant computational resources and expertise. The complexity of training and deploying these models can pose challenges for resource-constrained devices. Furthermore, integrating new machine learning techniques into existing IoT infrastructure can be challenging due to compatibility issues and the need for seamless integration. Ensuring the scalability of interoperations and compatibility between different components and devices is crucial for successful implementation. While machine learning models can enhance DDoS attack detection, they often rely on large-scale datasets, raising concerns about data privacy and security. Additionally, ML models are susceptible to adversarial attacks, such as those that manipulate the input data to evade detection. With current dynamic and evolving attack patterns, machine learning models trained on past attack patterns may struggle to identify novel or zero-day attacks that differ significantly from the training data as DDoS attack techniques are constantly evolving and adapting to circumvent detection mechanisms. Therefore, proper measures must be implemented to safeguard sensitive information and address privacy vulnerabilities in IoT networks.

Based on the identified gaps, the key gaps of this research can be outlined as follows:

- There is a lack of state-of-the-art datasets that are constantly updated to tackle current DDoS complexity and advancement.
- Multiple benchmark datasets suffer from severe class imbalance issues.
- There is a need to investigate a novel set of machine learning models in a comparative analysis study to assess their effectiveness.

Table 1. Summary of Studies.

Ref. No.	Year	Dataset	No. of Features	ML/DL Models	Highest ML/DL Model Accuracy	Performance Rate
[37]	2022	CICDDoS2019	Best 15 features selected by ExtraTreesClassifier from among 88 features	KNN, DT, RF, ANN	ANN	Accuracy (99.95%), precision (99.95%), recall (100%), F1-Score (99.97%)
[39]	2022	UNWS-NB15	-	RF and XGBoost	XGBoost	Average accuracy (AC = 90%), precision (90%), recall (90), F1-Score (90%)
[49]	2022	Kaggle banking dataset	Homogeneity measure (k-means clustering) for choosing important features	SVM, KNN, RF	SVM	Accuracy (99.8%), precision (99.07%), recall (98.32%), F1-Score (98.5%)
[29]	2022	BoT-IoT	Three different feature sets from 35 variables ranged from 15 and 18 each	ML = SVM, DT, RF DL = RNN, LSTM, GRU, MLP	DT (robust) and RF outperforms the DL models	Average accuracy (99%) and (100%) accuracy, precision, recall, and F1-Score
[51]	2022	CICIDS2017, CSE-CIC-IDS2018, CICDDoS2019, DoHBrw2020, NBaIoT2018, UNSW2018 BoTIoT, and UNSW NB15.	Two	AdaBoost, Bagging, GB, KNN, RF, VMFCVD (HAM, FDM, and DFDM modes).	VMFCVD (HAM mode)	Average accuracy (99%), precision (99.99%), and F1-Score (99.99%).
[36]	2022	KDD Cup99	60	Only ANN	-	-
[46]	2022	BoT-IoT	10 best features	DT, RF, KNN, MLP, RNN, LSTM	KNN, RF	KNN Accuracy (99.93) when not applying looking-back approach; RF Accuracy (99.81%) when applying looking-back approach
[38]	2021	CICDDoS2019	ANOVA, chi-squared test, and extra tree used to select the best features from among 79 features	RF, DT, KNN, XGBoost	XGBoost + ANOVA = 15 features	XGBoosta + ANOVA: accuracy (98.347%), precision (99%), recall (99%), F1-Score (99%); XGBoost + chi-squared: accuracy (92.67%); XGBoost + extra tree: accuracy (92.78%)
[31]	2021	CCD-INID-V1	83	XGBoost, RF	XGBoost coupled with RF	-

Table 1. *Cont.*

Ref. No.	Year	Dataset	No. of Features	ML/DL Models	Highest ML/DL Model Accuracy	Performance Rate
[40]	2021	CICDDoS2019	10 best features	Naïve Bayes, SVM, AdaBoost, XGBoost, KNN, RF	AdaBoost and XGBoost	Accuracy (100%), F1-Score (1.0)
[47]	2021	BoT-IoT	Top eight features were selected based on the Chi-Square	KNN, Gaussian Naïve Bayes, MLP ANN	KNN	Accuracy (92.1%), ROC AUC (92.2%) on highly imbalanced real-time data
[44]	2021	NSL-KDD	-	RF	RF	Accuracy (99.76%)
[41]	2020	Generated	40 out of 115 were selected	LSVM, NN, DT, RF	RF coupled with DT	TPR (99.7%), FPR (0.3%), precision (99.7%), recall (99.7%), F1-Score (99.7%), error MAE (DT = 0.31%, RF = 0.37%)
[42]	2020	Generated	27	RF, MLP, J48, naïve Bayes	J48 decision tree	Accuracy (98.64%)
[50]	2020	Modified CICIDS2017	15 out of 85 features	NSGA-II-aJG IDS, MLP, SVM, Bayes, RF	SVM	Accuracy (94.50%)
[48]	2018	Generated	Three stateless features, three stateful features	RF, KNN, LSVM, DT, NN	All five algorithms	Accuracy, precision, recall, F1-Score higher than (99.0%.)
[34]	2017	CIC-IDS2017	80	KNN, RF, ID3, Adaboost, MLP, naive Bayes, QDA	KNN, RF, and ID3	-
[45]	2022	Generated	26 features symbolize IoT network behavior and depend on the network flow information	SVM, RF, DT, KNN, NB, LR	RF	Accuracy (99.2%), precision (96.7%), recall (98.9%), F-Score (97.8%)

Table 2. Datasets Comparison.

Dataset	Year	Publicly Available	Hardware Utilized	Software Utilized	Limitations	Contains IoT Traffic	Setup (Real, Synthetic)
CCD-INID-V1	2021	-	Raspberry Pi, Rainbow HAT, smart sensors	Wireshark	Made in consideration with IDS. Does not contain DDoS traffic, only DoS. The dataset did not contain diverse internet usage cases, such as a person surfing the internet.	Yes	Real
TON_IoT	2020	Yes	Telemetry sensors, smart fridges, thermostats	Nmap, Nessus, Python Scapy NSX-VMware platform & Network audit logs	Imbalanced data, and does not provide intrusion assessment using different machine learning techniques with the proposed dataset to validate it.	Yes	Real
CIC-IDS2017	2017	Yes	Modems, routers, switches, firewalls	Ncrack, Metasploit modules, Nmap NSE, CICFlowMeter, Pcap Analyzer, CSV Generator	High-class imbalance, a huge volume of data, and missing values. Focus on multiple attacks including DDoS. Not focused on IoT context.	No	Real
BOT-IoT	2019	Yes	Four Kali attacking machines targeting Ubuntu server, Ubuntu mobile, Windows 7, and Metasploitable VMs. The firewall was used to ensure the validity of the dataset-labeling process	GoldenEye + hping3, Argus tool, and Ostinato tool	Severe class imbalance. Does not cover all types of DDoS attacks.	Yes	Real

Table 2. *Cont.*

Dataset	Year	Publicly Available	Hardware Utilized	Software Utilized	Limitations	Contains IoT Traffic	Setup (Real, Synthetic)
IoTID20	2020	Yes	Smart home devices, WiFi cameras, laptops, tablets, smartphones	-	Suffers from class imbalance. Does not cover all types of DDoS attacks (only Mirai DDoS).	Yes	Real
CSE-CIC-IDS2018	2018	Yes	Windows and Linux-based workstations	CICFlowMeter, Slowloris, LOIC, HOIC, DVWA, Patator, Hulk, GoldenEye, Slowhttptest, Selenium Framework, Dropbox, Nmap, portscan, Ares, Low Orbit Ion Canon (LOIC), AWS computing platform, OpenSSL, Adobe Acrobat Reader 9.	Seven different attack scenarios: brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside.	No	Real
CIC-DDoS-2019	2019	Yes	Servers, firewalls, switches, PCs	CICFlowMeter, scikit-learn	Not focused on IoT context.	No	Real
CIRA-CIC-DoHBrw-2020	2020	Yes	Servers, switches, routers.	Google Chrome, Mozilla Firefox, dns2tcp, DNSCat2, Iodine, AdGuard, Cloudflare, Google DNS, Quad9, DoHLyzer, GeckoDriver, DoH Data Collector, DoHMeter	Implementing DoH protocol within an application using five different browsers and tools and four servers to capture Benign DoH, Malicious DoH, and non-DoH traffic.	No	Synthetic
N-BaIoT2018	2018	Yes	Servers, switches, wires, routers, commercial IoT devices, scanners, and loaders	Wireshark, Keras	Focused on Mirai and Bashlite DDoS attack types only.	No	Real

Table 2. *Cont.*

Dataset	Year	Publicly Available	Hardware Utilized	Software Utilized	Limitations	Contains IoT Traffic	Setup (Real, Synthetic)
UNSW NB15	2015	Yes	Servers, routers, firewalls.	IXIA, tcpdump, Argus, Bro-IDS	DDoS attacks are not taken into consideration. Not focused on IoT context.	No	Synthetic
KDD Cup99	1999	Yes	Servers, PCs	MATLAB, DARPA'98 IDS Evaluation Program	Outdated dataset. Redundant records. Not focused on IoT context.	No	Synthetic
NSL-KDD	2009	Yes	KDD Cup99 setup	KDD Cup99 setup	All types of DDoS attacks are not considered in this dataset. Not focused on IoT context.	No	Synthetic

Looking forward, it is apparent that the threat of DDoS attacks by IoT devices is consistently growing. With the increasing adoption of IoT devices in homes and businesses, the potential attack surface is expanding at an alarming rate. Additionally, the continued development of new technologies and the emergence of new attack vectors will present new challenges regarding such attacks. The potential future trends in this field are expected to be the following:

- The persistence of DDoS attacks conducted by IoT devices is apparent with the continuity of lack of security measures on IoT devices and the rapid distribution of botnet viruses that continue to evolve each year.
- DDoS attacks in general are going to become even more frequent, more complicated, and relatively inexpensive to perform.
- The use of obscure IoT devices, such as smart refrigerators, thermostats, and CCTV cameras, was previously uncommon. Yet, such devices currently pose a dominant threat that can be utilized as botnets to perform DDoS attacks and take down a target's services or disrupt them.

4. Conclusions

To conclude, there has been an increased usage of IoT devices in recent years; this increase will continue because of the importance of IoT devices in serving the modern technology ecosystems that surround us. The weakness of security features in such devices may result in huge damage, such as the most common DDoS attack. This attack utilizes IoT devices with small computational resources to work under the control of the attacker and launch attacks on the remaining resources in the network. In this paper, we studied and compared different frameworks for detecting DDoS attacks in an IoT environment using ML models. The future trends in the field of DDoS attacks by IoT devices present significant challenges and opportunities. As IoT devices become more ubiquitous, the potential attack surface will continue to expand, thus making it more essential than ever to develop effective strategies for mitigating and detecting DDoS attacks. Based on our research gap analysis, we found that creating a new open-source IoT-based dataset would be a valuable contribution to the research field for detecting IoT DDoS attacks using ML models. Moreover, it is recommended that the number of IoT devices included in the setup be increased to come close to a realistic attack scenario. Additionally, exploring machine learning models that have received limited attention in this context, such as AdaBoost, could yield significant insights for future research in this field.

Author Contributions: Conceptualization, A.A.A., M.A. and F.A.; methodology, A.A.A., M.A., O.B.A., L.A.M., D.J.A., G.E.R. and S.A.B.; Work analysis, O.B.A., L.A.M., D.J.A., G.E.R. and S.A.B.; investigation, A.A.A., M.A. and F.A.; writing—original draft preparation, O.B.A., L.A.M., D.J.A., G.E.R. and S.A.B.; writing—review and editing, A.A.A., M.A. and F.A.; supervision, A.A.A. and M.A.; project administration, A.A.A., M.A. and F.A. funding acquisition, A.A.A., M.A. and F.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University for funding this project.

Data Availability Statement: Not applicable.

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

References

1. Wang, J.; Liu, Y.; Feng, H. IFACNN: Efficient DDoS attack detection based on improved firefly algorithm to optimize convolutional neural networks. *Math. Biosci. Eng.* **2021**, *19*, 1280–1303. [CrossRef]
2. Hasan, M. State of IOT 2022: Number of Connected IOT Devices Growing 18% to 14.4 Billion Globally. IoT Analytics. 2022. Available online: https://iot-analytics.com/number-connected-iot-devices/ (accessed on 14 May 2023).

3. Kim, H.S. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. 2015. Available online: https://www.researchgate.net/profile/Mohamed-Mourad-Lafifi/post/Is_there_any_simulation_tool_for_fog_computing/attachment/59d638c079197b8077995f4c/AS%3A398883160117248%401472112564706/download/Fog+Computing+and+the+Internet+of+Things++Extend+the+Cloud+to+Where+the+Things+Are.pdf (accessed on 28 August 2022).
4. Ma, K.; Bagula, A.; Nyirenda, C.; Ajayi, O. An IoT-Based Fog Computing Model. *Sensors* **2019**, *19*, 2783. [CrossRef]
5. Džaferović, E.; Sokol, A.; Abd Almisreb, A.; Norzeli, S.M. DoS and DDoS vulnerability of IoT: A review. *Sustain. Eng. Innov.* **2019**, *1*, 43–48. [CrossRef]
6. Mansfield-Devine, S. The evolution of DDoS. *Comput. Fraud Secur.* **2014**, *2014*, 15–20. [CrossRef]
7. Sieklik, B.; Macfarlane, R.; Buchanan, W. Evaluation of TFTP DDoS amplification attack. *Comput. Secur.* **2016**, *57*, 67–92. [CrossRef]
8. Lukaseder, T.; Stölzle, K.; Kleber, S.; Erb, B.; Kargl, F. An SDN-based Approach for Defending against Reflective DDoS Attacks. In Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 1–4 October 2018.
9. Neelam, D.; Prasenjit, M.; Shashank, S.; Rahamatullah, K. Research Trends in Security and DDoS in SDN. *Secur. Commun. Netw.* **2016**, *9*, 6386–6411.
10. Ehrenkranz, T.; Li, J. On the State of IP Spoofing Defense. *ACM Trans. Internet Technol.* **2009**, *9*, 1–29. [CrossRef]
11. Vishwakarma, R.; Jain, A.K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommun. Syst.* **2019**, *73*, 3–25. [CrossRef]
12. McGregory, S. Preparing for the next DDoS attack. *Netw. Secur.* **2013**, *2013*, 5–6. [CrossRef]
13. Dantas, Y.G.; Nigam, V.; Fonseca, I.E. A selective defense for application layer ddos attacks. In Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference, The Hague, The Netherlands, 24–26 September 2014.
14. Decision Tree Classification Algorithm. JavaTpoint. Available online: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed on 26 September 2022).
15. Random Forest Algorithm. JavaTpoint. Available online: https://www.javatpoint.com/machine-learning-random-forest-algorithm (accessed on 29 September 2022).
16. Yiu, T. Understanding Random Forest. Towardsdatascience. 2019. Available online: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed on 10 October 2022).
17. What Is a Random Forest? Available online: https://www.tibco.com/reference-center/what-is-a-random-forest (accessed on 26 September 2022).
18. K-Nearest Neighbor(KNN) Algorithm for Machine Learning. Javatpoint. Available online: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (accessed on 10 October 2022).
19. What Is XGBoost? NVIDIA Data Science Glossary. Available online: https://www.nvidia.com/en-us/glossary/data-science/xgboost/ (accessed on 10 October 2022).
20. XGBoost. Geeksforgeeks. 2022. Available online: https://www.geeksforgeeks.org/xgboost/ (accessed on 29 September 2022).
21. Ghatak, K. XGBoost Algorithm in Machine Learning. Naukri Learning. 2022. Available online: https://www.shiksha.com/online-courses/articles/xgboost-algorithm-in-machine-learning/ (accessed on 9 January 2023).
22. Artificial Neural Network Tutorial. Javatpoint. Available online: https://www.javatpoint.com/artificial-neural-network (accessed on 10 October 2022).
23. Recurrent Neural Network Algorithms Overview. BUSINESS & AI: Artificial Intelligence for Better Decision Making. Available online: https://www.business-and-ai.com/recurrent-neural-network-algorithms-overview/ (accessed on 26 September 2022).
24. Support Vector Machine Algorithm. Javatpoint. Available online: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (accessed on 10 October 2022).
25. Introduction to Support Vector Machines (SVM). Geeksforgeeks. 2022. Available online: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/ (accessed on 29 September 2022).
26. The Ultimate Guide to AdaBoost Algorithm | What Is AdaBoost Algorithm? Great Learning. 2022. Available online: https://www.mygreatlearning.com/blog/adaboost-algorithm/ (accessed on 29 September 2022).
27. Boosting in Machine Learning | Boosting and AdaBoost. Geeksforgeeks. 2022. Available online: https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/ (accessed on 29 September 2022).
28. Saini, A. AdaBoost Algorithm—A Complete Guide for Beginners. Analytics Vidhya. 2021. Available online: https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/ (accessed on 29 September 2022).
29. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors* **2022**, *22*, 3367. [CrossRef]
30. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.P. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
31. Liu, Z.; Thapa, N.; Shaver, A.; Roy, K.; Siddula, M.; Yuan, X.; Yu, A. Using Embedded Feature Selection and CNN for Classification on CCD-INID-V1—A New IoT Dataset. *Sensors* **2021**, *21*, 4834. [CrossRef]
32. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In *Advances in Artificial Intelligence*; Springer: Cham, Switzerland, 2020.
33. Alsaed, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A NewGeneration Dataset of IoT and IIoT forDatadriven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]

34. Panigrahi, R.; Borah, S. A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *Int. J. Eng. Technol.* **2018**, *7*, 479–482.
35. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the International Conference on Information Systems Security And Privacy (ICISSP), Funchal, Portugal, 22–24 January 2018; Volume 7, pp. 479–482.
36. Gopi, R.; Sathiyamoorthi, V.; Selvakumar, S.; Manikandan, R.; Chatterjee, P.; Jhanjhi, N.Z.; Luhach, A.K. Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things. *Multimed. Tools Appl.* **2021**, *24*, 26739–26757. [CrossRef]
37. Amrish, R.; Bavapriyan, K.; Gopinaath, V.; Jawahar, A.; Vinoth, C.K. DDoS Detection using Machine Learning Techniques. *J. IoT Soc. Mob. Anal. Cloud* **2022**, *4*, 24–32. [CrossRef]
38. Gaur, V.; Kumar, R. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. *Arab. J. Sci. Eng.* **2022**, *47*, 1353–1374. [CrossRef]
39. Ismail, M.I.; Mohmand, H.; Hussain, A.A.; Khan, U.; Ullah, M.; Zakarya, A.; Ahmed, M.; Raza, I.; Rahman, U.; Haleem, M. A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks. *IEEE Access* **2022**, *10*, 21443–21454. [CrossRef]
40. Seifousadati, A.; Ghasemshirazi, S.; Fathian, M. A Machine Learning Approach for DDoS Detection on IoT Devices. *arXiv* **2021**, arXiv:2110.14911.
41. Aysa, M.H.; Ibrahim, A.A.; Mohammed, A.H. IoT Ddos Attack Detection Using Machine Learning. In Proceedings of the 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 22–24 October 2020; pp. 1–7.
42. Saini, P.S.; Behal, S.; Bhatia, S. Detection of DDoS Attacks using Machine Learning Algorithms. In Proceedings of the 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 12–14 March 2020; Volume 78, pp. 16–21.
43. Alkasassbeh, M.; Al-Naymat, G.; Hassanat, A.B.; Almseidin, M. Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2016**, *7*, 436–445. [CrossRef]
44. Pande, S.; Khamparia, A.; Gupta, D.; Thanh, D.N.H. DDOS Detection Using Machine Learning Technique. In *Recent Studies on Computational Intelligence*; Studies in Computational Intelligence; Springer: Singapore, 2021; Volume 921.
45. Gupta, B.B.; Chaudhary, P.; Chang, X.; Nedjah, N. Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers. *Comput. Electr. Eng.* **2022**, *98*, 107726. [CrossRef]
46. Mihoub, A.; Fredj, O.B.; Cheikhrouhou, O.; Derhab, A.; Krichen, M. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Comput. Electr. Eng.* **2022**, *98*, 107716. [CrossRef]
47. Pokhrel, S.; Abbas, R.; Aryal, B. IoT Security: Botnet detection in IoT using Machine learning. *arXiv* **2021**, arXiv:2104.02231.
48. Doshi, R.; Apthorpe, N.; Feamster, N. Machine Learning DDoS Detection for Consumer Internet of Things Devices. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 29–35.
49. Islam, U.; Muhammad, A.; Mansoor, R.; Hossain, M.S.; Ahmad, I.; Tageldin, E.; Khan, J.A.; Rehman, A.U.; Shafiq, M. Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. *Sustainability* **2022**, *14*, 8374. [CrossRef]
50. Roopak, M.; Tian, G.Y.; Chambers, J. An Intrusion Detection System Against DDoS Attacks in IoT Networks. In Proceedings of the 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 562–567.
51. Prasad, A.; Chandra, S. VMFCVD: An Optimized Framework to Combat Volumetric DDoS Attacks using Machine Learning. *Arab. J. Sci. Eng.* **2022**, *47*, 9965–9983. [CrossRef]
52. Sharma, D.K.; Dhankha, T.; Agrawal, G.; Singh, S.K.; Gupta, D.; Nebhen, J.; Razzak, I. Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Netw.* **2021**, *121*, 102603. [CrossRef]
53. Hassan, K.F.; Manna, M.E. Detection and mitigation of DDoS attacks in the Internet of things using a fog computing hybrid approach. *Bull. Electr. Eng. Inform.* **2022**, *11*, 1604–1613. [CrossRef]

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