Journal of *Cybersecurity and Privacy*

Article

A Survey on Botnets, Issues, Threats, Methods, Detection and Prevention

Harry Owen [1], Javad Zarrin [2,*] and Shahrzad M. Pour [3]

[1] School of Computing and Information Science, Anglia Ruskin University, Cambridge CB1 1PT, UK; ho223@student.aru.ac.uk
[2] School of Design and Informatics, Abertay University, Dundee DD1 1HG, UK
[3] Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), 2800 Kgs. Lyngby, Denmark; shmp@dtu.dk
[*] Correspondence: j.zarrin@abertay.ac.uk

Abstract: Botnets have become increasingly common and progressively dangerous to both business and domestic networks alike. Due to the Covid-19 pandemic, a large quantity of the population has been performing corporate activities from their homes. This leads to speculation that most computer users and employees working remotely do not have proper defences against botnets, resulting in botnet infection propagating to other devices connected to the target network. Consequently, not only did botnet infection occur within the target user's machine but also neighbouring devices. The focus of this paper is to review and investigate current state of the art and research works for both methods of infection, such as how a botnet could penetrate a system or network directly or indirectly, and standard detection strategies that had been used in the past. Furthermore, we investigate the capabilities of Artificial Intelligence (AI) to create innovative approaches for botnet detection to enable making predictions as to whether there are botnets present within a network. The paper also discusses methods that threat-actors may be used to infect target devices with botnet code. Machine learning algorithms are examined to determine how they may be used to assist AI-based detection and what advantages and disadvantages they would have to compare the most suitable algorithm businesses could use. Finally, current botnet prevention and countermeasures are discussed to determine how botnets can be prevented from corporate and domestic networks and ensure that future attacks can be prevented.

Keywords: botnets; botnet infection; botnet detection; machine learning for botnets; deep learning for botnets; botnet mitigation; botnet prevention

check for updates

Citation: Owen, H.; Zarrin, J.; Pour, S.M. A Survey on Botnets, Issues, Threats, Methods, Detection and Prevention. *J. Cybersecur. Priv.* **2022**, *2*, 74–88. https://doi.org/10.3390/jcp2010006

Academic Editor: Giorgio Giacinto

Received: 31 December 2021
Accepted: 21 February 2022
Published: 28 February 2022

1. Introduction

Botnets are tools used to deliver malware across networks. This defines them as major threats to corporate networks, which may always need data, applications and services available. Botnets are the primary cause of Denial-of-Service (DOS) attacks. They can prevent employees from accessing (sensitive and confidential) information stored within the networks. It may negatively affect the data's availability on the system and the data-handling credibility of the data handler. Despite how common DOS attacks are, businesses may not have any solutions to combat this threat. A well-known example of a botnet attack is the Mirai botnet attack in 2016 [1]. The Mirai bot-master (i.e., the malware owner) conducted reconnaissance of unsecured Internet of Things (IoT) devices that were connected to the target networks and then successfully launched the malware against unsecured ports. The Mirai botnet successfully accessed target devices due to poor IoT security configuration, e.g., with weak passwords being assigned. Also, the malware used brute-forced password cracking to access IoT devices. Kraken, another botnet attack that occurred in 2008, infected over 400,000 personal computers (PCs) and was predominantly used for spamming other network users to gain system access or cause system damage [2]. The number of botnet attacks is rapidly increasing with many new variants based on more advanced techniques which can't be easily defended using traditional and old defence strategies. Thus, mitigating botnets and providing businesses with the best solution to respond to the large number of various threats that may exploit them is becoming more challenging. AI is known to assist with the improvement of network security due to its decision-making capabilities. However, also, AI can be used by attackers to launch more sophisticated botnet attacks. This highlights the importance of AI to be considered as key enabler to design and develop more efficient botnet detection and prevention tools and strategies compared to the old methods. For example, AI-based methods are particularly necessary for IoT networks. Their in-built security may not be updated and able to counter botnet threats, since threat actors can create botnet code variations to bypass certain firewalls, allowing them unauthorised access to hosts. It is therefore important to consider what methods can be used to prevent and mitigate botnet attacks. This paper provides a review on current and past methods for botnet infections, followed by discussing traditional and modern AI-based methods and strategies for botnet detection and prevention.

The remainder of this paper is organized as follows: Section 2 discusses what a botnet is, along with what its capabilities are when it has successfully infected a target host. Section 3 explains the various methodologies bot-masters may use to infect hosts, either through code variation or vulnerability exploitation. Section 4 details how botnets can be detected by the system along with how AI and Machine Learning algorithms may contribute to botnet detection. Section 5 discusses prevention and countermeasures that businesses and individuals can use to avoid botnet-related threats. Section 6 discusses and evaluates the efficiency of some well-known Machine Learning algorithms for botnet detection. Finally, Section 7 provides an overall conclusion.

2. Definition of Botnets

Chen et al. [3] defines botnets as one of the main methods of transporting malware throughout networks. This methodology is responsible for a multitude of cyber-attacks such as Distributed Denial of Service (DDoS) attacks. Botnet DDoS attacks are performed by infecting a target network device. The infected device would then become known as a "zombie" a device under a threat actor's control. By using the zombie, bot-masters can then spread the botnet throughout the network and reach other connected devices. For DDoS attacks, the botnet can cause the network's bandwidth to be used up more than what is intended and will disrupt all services that each host would contain [4]. Since network services may be in failure, this may compromise data availability. Silva et al. [5] defines a botnet as a series of malware-infected devices that then create a network of said devices to propagate the target network further. The spread can be used to prevent services and disrupt the data availability. Therefore, denying permitted users to access specific information when it is needed. Botnets are utilized in a variety of cyber-crimes, be it enabled (where technology is used to commit them but are not relied on) or dependent (where technology is required to do so, such as DDoS or stealing sensitive information of their choice). Silva et al. [5] further mentions that the hacker instigating a botnet attack would be referred to as the bot-master, who is also responsible for controlling the bot network. Chen et al. [3] and Silva et al. [5] provided similar definitions into what a botnet is, along with discussing the rise in such attacks. Ref. [3] emphasizes that hackers would normally target Peer-to-Peer networks, a type of network that has computers with similar roles connected to the network with their data shared via wired networks lacking servers. Such infrastructure can be useful for Local Area Networks (LANs) [6]), as there would be more connectivity between network devices so the malware can go from one infected device to another device at a faster rate.

Businesses are common targets for botnets to commit information theft [7] enabling bot-masters to perform further criminal activities [3,5]. In the past, zombies (i.e., infected victim computers) were originally used for Internet Relay Chats (IRCs), allowing assistance with messaging throughout the internet [7]. An IRC channel enables users to join and communicate with other users who exist on that channel, behaving very similar to social media chat rooms. If the channel had many users, then bots would be created to monitor the channel. However, in such a case, the bot was not an infected machine as known today; rather, it was an automated program for IRC channel administration. Over time, botnets were used to attack IRC servers by sending a large number of network packets resulting in DDoS attacks as the traffic was too much for the target IRC server to manage.

3. Infection Methods

One of the main causes of botnet malware infection is open ports. Although open ports have a vital role in network device communication, allowing packets to be transmitted freely [8]. Nevertheless, open ports may result in bot-masters transmitting botnet packets throughout the network. This can also apply to black hat hackers using the open ports to gain access to the network and use the software that would coincide with the open port to carry out privilege escalation. Once the threat actor has gained access through the port, the botnet can then be launched from the device by using the open ports as a route for the infected packets to reach other devices. HTTP ports allow a packet to be transferred throughout the internet. If unnecessarily opened, HTTP ports are in the hands of a bot-master, they could transfer the botnet codes at a faster rate [9]. Botnets can use flags to allow for flooding HTTP packets [10]. Flooding is where the network router would send an outgoing multi-cast to other connected devices. This can be an advantage to bot-masters as the flooding process would allow more devices to be reached more quickly [11]. HTTP is a more commonly used protocol bot-masters would favour in terms of propagation. Furthermore, flooding is a technique widely used within LANs that are scaled greater than others, which would coincide with companies that would contain more hardware making HTTP flooding approaches more ideal for bot-masters to use.

Botnets have been recorded as capable of infecting Secure Shell (SSH) based sessions to propagate [10]. Despite the utilization of encryption to ensure security, SSH sessions can still be penetrated as the bot-master may create alterations within the botnet code to build a new variant. Within the variation, the botnet can conduct brute force attacks to access the SSH sessions as exploiting cases where weak passwords have been used [12–14]. Botnets can use dictionary-based attacks to be able to gain access to the sessions and allow for the infection to be performed [15]. Feily et al. [16] elaborates that botnets are like Trojan-based malware as both botnets and Trojans may penetrate networks via phishing. This is also spread out by Cooke et al. [7], as the paper mentions that many bot-masters find methods of code obfuscation to avoid being detected by anti-virus and anti-malware software.

Hong [17] defines phishing attacks as a cyber-attack that coincides with the concept of social engineering. Phishing is where attackers send emails to a target and deceive them into responding to gain data or run malware. Phishing occurs in three stages, picking a target, the target responding to the email, and the phishing email acting. Phishing emails commonly present an urgent threat to users, informing them that they only have a certain period to open the link, causing the user to panic and respond, thus executing the malware. Bot-masters are responsible for sending phishing emails that have been weaponized to trigger the botnets [7,15,17]. The phishing emails persuade users to open the infected attachments. Jeong et al. [15] discusses hyperlinked attachments, where users would unknowingly click on a hyperlink, causing the botnet malware to be downloaded by the unsuspecting user. Jeong et al. [15] also state that the hyperlinks may take victim users to phishing websites, which trick users into downloading the botnet malware unwittingly. Khonji et al. [18] further justifies [15] and states that download links within emails may be classified as a “Man in the Browser (MITB)” cyber-attacks, which, according to [19], are like Man-In-The-Middle attacks. However, in MITB attacks, the attack exists within the Hypertext Transfer Protocol (HTTP) or the Hypertext Transfer Protocol Secure (HTTPS). The threat actor intercepts the session and causes code to be injected. In the case of botnets, the MITB attack could contain botnet code, and when the session between the user and the infected website occurs, the MITB attack will trigger the dormant code. Abbas et al. [9]

mentions the reason hosts MITB vulnerability exists is because the HTTP/HTTPS may not be updated to be more secure against MITB attacks. This allows internet-based MITB-exploiting botnets to be more likely to infect victim web pages.

Jeong et al. [15] discusses a botnet infection method that is based on dictionary attacks. Dictionary attacks are a method of password cracking where all words existing within a dictionary are used to brute force a password [20]. From Nam et al. [21], dictionary attacks may be divided into either online or offline attacks. Online attacks are where each word insertion is checked by using a server that is connected to a computer. This has the drawback of being easy to detect for the attacker, since they are connected to the server, allowing their activities to be detected by detection mechanisms. Therefore, online attacks are not preferable for bot-masters to use [21]. Alternatively, dictionary attacks can take place offline. In this case, threat actors have obtained, and attempt to crack, password hashes from the targeted network (e.g., through sniffing the network during a reconnaissance phase) [22]. Other methods of password hash collection discussed by Miller [23] are that memory dumps may be observed (e.g., to find a Sequence Alignment Map (SAM) file). SAM files are relevant to attackers since password hashes are recorded into them, although this only applies to Windows machines. Once password hashes have been obtained, bot-masters can decipher them without using web servers [23]. Offline dictionary attacks are preferable since the threat actor would not be attempting to gain network access directly, which means that the network would not have access attempts registered within its security log, allowing for the bot-master potentially remain undetected [23].

Variants of dictionary attacks can also be used by threat actors to evade detection and breach systems [12,24]. The first dictionary attack type is slow motion-based. This is where a threat actor would conduct the attack, but the target sessions are constantly alternating, escaping detection once the attack is conducted. The other method for dictionary attacks is distributed, where the attack is launched through broadcasting its messages throughout the target network, behaving similarly to botnet attacks but can also coincide with dictionary attacks as both broadcasts through the target network then the attack can be performed on multiple machines to access at least one host. The botmaster can use the accessed host to conduct the botnet attack. Thereby allowing the botmaster to implant botnet code into the host and infect it [24].

4. Detection Methods

Karim et al. [25] discusses types of detection mechanisms that may be implemented within networks to alert administrators regarding incoming attacks. The examples addressed are Anomaly-based and Botnet based detection mechanisms and detection using Fuzzy Logic. In addition, phishing detection is covered due to phishing-related attacks being one of the main contributions to how botnets can infect networks.

4.1. Detection Types

Anomaly-based detection observes network traffic and identifies abnormal activity within communications [25]. In the case of botnets, malware code within packets is detectable. This is justified by Ahmed et al. [26] as the analysis technique would focus on the data patterns that exist within the network. When using anomaly detection, the main focus that [26] reveals is: a point anomaly which is a specific area of the data pattern that does not conform to the behaviour standard of the area it exists in would be called a point anomaly; in the case of Botnets it is important to observe the traffic and determine if any packet would have different behaviour. Contextual anomalies are a case where the data would have behaviour outside of its context by making comparisons to other data sets [27]. The last type of anomaly that Ahmed et al. [26] elaborates are collective anomalies; this is where a group of data contains similarities to each other would be analysed for anomalies which can be useful for data analysts to conduct a feature extraction and predict if there are any inconsistencies within the set. Chen et al. [28] discusses the signature-based detection, which is a method used to assist the detection of anomalies. However, Chen et al. [28]

states that botnet detection using signature analysis has drawbacks. The first is that if implemented on a large corporate network, the sheer network scale causes the detection system to not be as effective. The second is that the rules applied to the detection will not work because the botnet would cause changes to the network's communication. A recent detection method addressed by Chen et al. [28] is host-based detection. This is where hosts on a network, along with their contents (e.g., files and running processes) are kept under surveillance for abnormalities (e.g., botnet activity). However, a problem with this method is that novel botnet code adaptations may be applied by the bot-master thus rendering host-based detection useless since it is reliant on known signatures (and therefore bypassed by unknown novel code).

There are also botnet detection tools like the SLINGbot [29]. Rahim and Bin Muhaya [29] describes SLINGbot as a python-coded tool capable of detecting botnets. The detection is done by using a Botnet Scenario Driver (BSD), to control the botnet, and a Composable Botnet Framework (CBF) to ensure that the botnet and information on the routing are connected. This tool can use the data gathered and make predictions on future attacks so that companies would be more prepared for those attacks. Furthermore, Python is regarded as a secure programming language, due to a lack of vulnerabilities that black-hat hackers could use to exploit. This would provide more assurance that the SLINGbot would be less prone to being infected or damaged. García et al. [30] mentions a type of anomaly detection called CAMNEP detection, which is used to analyse the activity of a network to locate anomalies. CAMNEP uses a three-layer methodology to ensure that the data gathered from the detection is reliable. The first layer is the anomaly detection layer which further relates to [26] as well as [3] since both sources discuss the use of anomaly detection, by analysing the network traffic using analysis algorithms. One such algorithm that could be used is the Hierarchical Temporal Memory (HTM) algorithm, which is defined by Lavin and Ahmad [31]. It is used to survey activity in actual time; this is useful in anomaly detection as the flowing network traffic occurs in real-time. The second layer is where a concept referred to as trust models are used, where the anomalies would then be clustered based on their behaviour. The third layer of the CAMNEP solution is where aggregation takes place, this is to generate a conclusion based on the average of each clustered data point and conduct the best solution for the anomalies based on said aggregations [30].

4.2. Machine Learning Based Detection

Considering the Internet of Things (IoT), Alhajri et al. [32] addresses that machine-based learning can also be factorial in the detection of botnets on an anomaly type basis. The Auto-Encoders is used as an algorithm to assist with the botnet detection. The detection as defined by Jordan [33] is a type of machine learning where a neural network is implemented with a bottleneck to divide the layers in the network. The detection is divided into two areas: one part would exist in the IoT's cloud to survey the data traffic from there and the other is applied to the sensors in the network, which can then be monitored for anomalies.

Another anomaly-based detection implemented an Intrusion Detection System (IDS) as mentioned by both [34,35]. An IDS is used to survey any unauthorised accesses within the network by implementing a signature-based detection approach. If there is any unauthorised access present, the IDS would activate and run the specific protocols to prevent system casualty. Although the issue that [34,36] encountered is that IDS will only be functional to known botnets meaning that it would need to be trained to recognise the different anomalies depending on the situation. Lawal et al. [37] further justifies the use of IDS as it can function in two areas, host-level where the hosts are surveyed, and the network-level to identify anomalies on the entire network. The system can make comparisons to behaviour to what is already registered within its database, but it can only succeed on known attacks [34,36]. Anomaly-based is used to detect anomalies behaving similarly to work proposed by Karim et al. [25]. Ashraf et al. [34] proposed a botnet detection that uses a function where it would conduct a feature extraction from the network traffic data. Feature extraction is where Machine Learning (ML) can use data and filter out certain features to ensure that there is a better prediction accuracy [38]. Feature engineering can be done on the network traffic data in order to specify highly relevant data that can be used to train machine learning models enabling botnet traffic detection. Tobiyama et al. [39] proposed a method based on Deep Neural Network (DNN) and Recurrent Neural Network (RNN) that conducts pattern recognition on the network traffic data and then use the identified patterns to generate predictions [40]. In terms of the feature extraction that is proposed by Tobiyama et al. [39], it uses the network traffic to observe its behaviour, which would then use RNN to find any recurring patterns within the traffic behaviour. The behaviour that is found would then be used as extracted features. If there is an anomaly it would classify the data and then cause an alert to the administrators.

Figure 1 shows interactions between IDS and IPS for intrusion detection through analysing network traffic. If the IDS can detect any traces of botnet packets within the traffic capture, then the IPS will be notified and will conduct the appropriate measures to counteract the threat. If not, then nothing will happen, and the processes will be repeated until an intrusion would be detected.

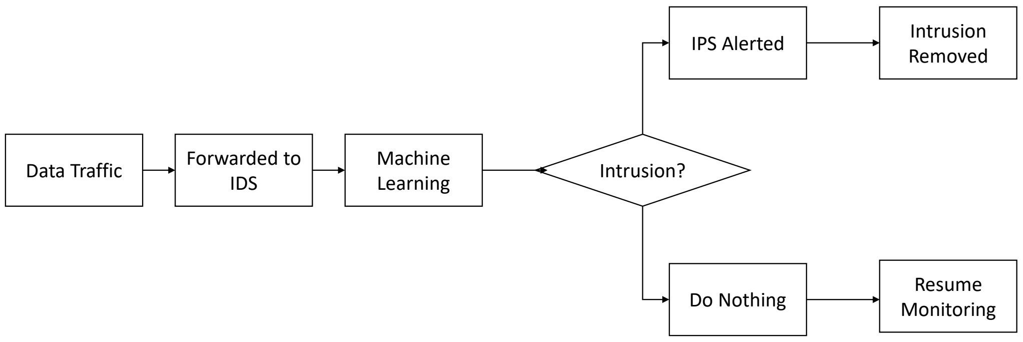

Figure 1. Interactions between IDS and IPS.

Figure 2 represents an example of how a Neural Network can be used to assist with detecting botnets. This would coincide with [34,39] proposals on the use of neural networks to detect botnet traffic. The neural network uses traffic data for training. By using synapses, the data would enter the hidden layer where deep learning would extract specific features and use the pattern recognition from the RNN to identify any similarities from the decomposed data. Further deep learning would then be conducted to identify if any anomalies are resulting in the output layer predicting if there is either an anomaly present or not.

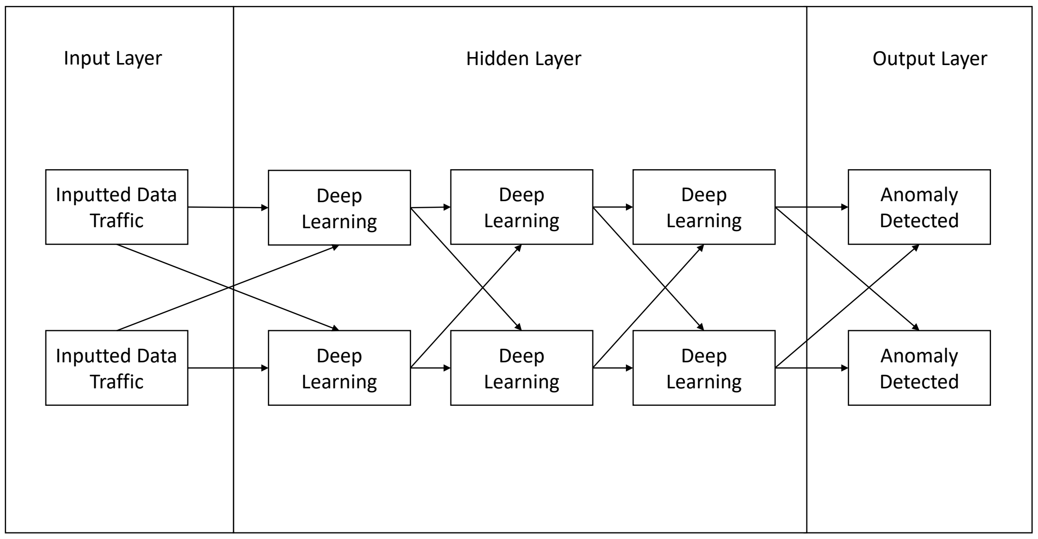

Figure 2. Neural Networks for botnet detection.

Botnet detection can also coincide with decision tree classification. Firdausi et al. [41] proposed a method that conducts a prediction on input data that involves network traffic and would then uses feature selection to create decision trees on whether there is a botnet or malware present. Feature extraction through the hidden layers can contribute to the performance, and the overall outcomes of the prediction [39]. Performing feature extraction can ensure that the data sets would be reduced in size to only the relevant data features. This will make the accuracy much higher as there would be the possibility of missing values existing in the frame analysis for some part of the traffic data [41].

Clustering is another type of ML algorithms that can detect anomalies. This is where points of data that originate from a data set would be positioned into a table depending on the features of specific points that are assigned to it. Other points of data would be positioned in different areas of the graph. If the following points are like others, then they would be closer to the corresponding points and would form a cluster [42]. In the case of botnet detection, clustering can be useful as it would be simple to develop, which would allow analysts to behave more time dedicated to observing the results shown over programming [43]. Despite both the simplicity and accessibility that clustering can provide, Seither [43] mentions that there are drawbacks to clustering data. Such as K-means based clustering would only be able to organize data that is numerically valued. When predicting botnets, analysts would consider features such as the flags that appear within the data packets or both the source and destination IP addresses which would be classified as string data types. If K-means clustering is performed, then the string features within the data set will need to be converted into numerical values. Label encoders can be used to convert the strings into float data. However, it can be inconvenient as the numerical conversion could result in the data being difficult for the analysts to understand. Thereby K-means based clustering would not be preferred by analysts since the predictions would consider the string variables rather than just using integers or floating data.

Support Vector Machines (SVMs) are another consideration that could be used to predict the presence of botnets. An SVM is an algorithm that would incorporate supervised data; which in this case would be the typical network traffic behaviour for the company [44]. Zhang et al. [44] further discusses that the SVM would use a hyper-plane to divide data points that exist within the graphs to classify each point to determine if there are any anomalies. Some advantages that SVM's can provide for botnet classification include if the Botnet data set, cannot be separated through a single-dimensional means, then a process referred to as the kernel trick is implemented to increase the number of dimensions to ensure that a more accurate classification can be made. This is relevant to botnet prediction as the data frames containing the typical traffic information would contain more extracted features. This means that the prediction dimension might not be performed using the features required to make the prediction The kernel trick is carried out by K(x,y) = <f(x), f(y) [45]. Afonja [45] defines the formula as "K" being represented as the Kernel while the "x" and "y" values would be treated as the number of dimension inputs (how many dimensions would be used). "f" is the number of dimensions that would exist within the kernel. Finally, the "<x,y>" section of the formula is the dot product of the algorithm [45]. Another advantage that Statinfer [46] elaborates is that SVM's are also suitable for unsupervised classification. In the case of predicting Botnet presence, the outcomes as to whether a data point is a botnet would not be known to the analyst. Thereby conducting a feature extraction would allow the SVM to use every detail within the data set and then conduct an automatic division for each point.

Although there are disadvantages that Statinfer [46] addresses such as training the SVM may be more time-consuming as botnet data sets would have a large scale since network traffic would have a large data packet quantity. It can also be difficult to select the most suitable kernel for classifying the traffic data, as there are multiple kernel types.

Logistic regression is an algorithm that can be used to classify if the data is malware (1) or not (0). Kumar et al. [47] used a training set that allows for the data to be organized into a confusion matrix (See Table 1). The True Positive (TP) in this case would be that the traffic would be actual malware, False Positive (FP) is non-botnet packets being incorrectly labelled as a botnet. True Negative (TN) is data packets that are correctly labelled as non-botnets, while False Negative (FN) is actual botnet data being treated as non-botnet data. To ensure that the predictions are accurate Kumar et al. [47] proposed the use of Precision which is where the True Positive is divided by the sum of both the True and False Positives to determine how precise the matrix is creating the formula of (p = TP/FP). The recall is also considered where the True Positive is divided by the sum of True Positive and False Negative (R = TP/(TP + FN)) [48]. Another method to improve the accuracy of the regression is penalties. They would be applied for each incorrect classification and ensure the model does not repeat the mistakes made. These include applying more weight to certain data points to ensure that the matrix would be more trained against the poorly predicted areas. Using logistic regression data packets can be classified to determine if the examined packets would contain botnet data or not. Thereby, logistic regression can be considered an efficient solution to classify botnets as the accuracy improvements are simple to implement and not require selecting a kernel as opposed to SVM's or using longer model training periods that clustering algorithm would need.

Table 1. Botnet classification confusion matrix.

	Positive	**Negative**
Positive	Malware Packet	Normal Packet but labelled as Botnet Packet
Negative	Botnet Packet labelled as a Non-botnet Packet	Normal Packet

A fog-based framework is a framework that functions by having a fog layer, which is used to cover the entirety of the network, ensuring all data storage's within the network are covered, Fog networks are highly compatible with IoT systems [36,49]. Fog computing is advantageous as the response time can be faster as well as an increase in scale, allowing the network to be altered more efficiently [37,50]. The fog-based framework does have disadvantages as well. According to George et al. [51], Fog-based computing would require all data within the network to be accessed by the framework. This would result in confidential information being revealed to unauthorized users, which could leading to specific information being leaked to other sources. Another drawback that is addressed by George et al. [51] is that the framework would be time-consuming due to developers and administrators needing to scale the network as well as ensure that there would be enough bandwidth present to gain a faster result. To assist with increasing network bandwidth, one way is to delete programs or files that would use large quantities of bandwidth. However, the issue with this mitigation is that some files might be an asset to the company. This can also apply to programs as there is the possibility that they would be instrumental to conducting work performance. Therefore, developers would need to select carefully which files and programs would need to be deleted.

Another tool that can be useful in terms of detecting botnets is honeypots. Garre et al. [12] defines honeypots as a network tool that can lure threat actors into conducting exploits. Tsikerdekis et al. [52] justifies that honeypots are for exploitation analysis and assist with providing more prevention strategies for the network holders. It can also coincide with Machine Learning as the data gathered from the honeypot analysis would then record the information into the ML database to allow the system to adapt to the newer threats [32]. According to Mukherjee [53], honeypots are classified as three different types; High interaction would incorporate services used on an actual network as well as the host's Operating System (OS) to seem like an authentic host which would then increase the likelihood of the honeypot being infected. Medium interaction is not as complex as high interaction type honeypots but will strike a balance between interacting with resources from higher to a lower rate. There could be lower interaction in specific areas because it can allow security admins to have enough time to be able to respond to threat actors. Finally, Low interaction honeypots would contain the least number of services to maintain security. This allows them to be the simplest of the three types to develop and conduct maintenance on, although the drawback comes with the honeypot not being as convincing to threat actors as opposed to High and Medium Interaction.

4.3. Fuzzy Logic Based Detection

Joshi et al. [54] proposed a strategy based on fuzzy logic. Defined by Chai [55], it is an AI-based method that does not use Boolean logic to its full extent precisely. However, it does incorporate true and false to assist with decision making when the outcome is unknown and control the system when there is a certain amount of input's being involved and allows the AI to create its own decision on how the outcome would operate. Joshi et al. [54] proposed an approach that analyses the network traffic and would conduct pre-processing from the traffic's data set. Joshi et al. [54] used a data set that contains a mix of both types of traffic. Then using fuzzy logic, enables the system to contain feature's that it would not previously have that can complement the traffic data set more effectively. The fuzzy logic's conclusions and the newly established features allow it to observe any anomalies that are within the network and then make connections to what is in its database and detect the botnet activity on its own.

4.4. Phishing Detection

Phishing detection can also be a crucial part in both prevention and detection of botnets, Moghimi and Varjani [56] provided two separate methodologies in terms of a web page's authenticity since infected website links that are attached to emails can be one of the causes of the botnet's being triggered into the system and infecting the host. In the case of how Moghimi and Varjani [56] accomplishes the detection by using a pre-processing methodology for a data set [54] by filtering out unwanted data as well as data that would be redundant in the machine learning. The phishing data would then be sampled to train the machine learning for the detection. Then the features for the phishing detection model would be implemented, such as what features are already contained within the model [56]. By determining from the data if it is True Positive or False Positive for phishing, said features can be used to evaluate whether a website is an actual or phishing type; this represents whether it is used for phishing attacks or is an actual website. The phishing detection model proposed by Moghimi and Varjani [56] used a database and generates its features on how phishing websites are and accordingly uses the information that has been gathered to generate its conclusions on whether it is real or not. According to Abu-Nimeh et al. [57], ML can also be used to assist with phishing detection via Neural Networking. Within the input layer, potential phishing emails can be inserted and via synapses can transition through the hidden layer of the network, through the hidden layer the emails would conduct functions depending on the node further relating to [34,39], discussing using neural networks in the case of whether data packets would contain botnet code or not as an unsupervised prediction. Weights should also be necessary to ensure that there would be a better classification on the emails to ensure that the hidden layer nodes are weaker, in the case of the email's the weaker nodes can cause incorrect predictions. In addition, neural networks can also be difficult to implement as explained by Abu-Nimeh et al. [57] can be due to the specific regularization involved, therefore conducting these predictions through a Neural Network should be conducted if the analyst is experienced with the construction as well as the analysis. Features that can be extracted for the machine learning would be if the wording within the email will be poorly spelt or if the grammar is along with if the message would contain the phrase "Dear Customer" at the start [58]. Although that would mean that when using training data, it would need to consider all the features contained within an email, such as the proper spelling, file attachments, etc. Furthermore, it would also be important to implement a spam filter within the network, a defence mechanism that uses machine learning to classify if the email is spam or an actual email [59]. When relating to domestic networks, it can also cause the implication that there would be no resources to generate a Neural Network construction. Furthermore, it would be time-consuming for users to develop and obstruct business progress.

Dedeturk and Akay [60] mentions that Spam Filtering would use a logistic regression algorithm to classify if the email would be spam or not through binary means. Spam emails can be defined as one (1) and real emails can be labelled as (0) values. However, Logistic Regression may classify the emails incorrectly as legitimate emails would be labelled as spam and vice-versa. The results of potential phishing emails are shown through a confusion matrix labelling each as a TP would be equal to a spam email, TN would be treated as a non-spam email FP is where an email is classified as a spam email but not, and FN is an email that is spam but is classified as a non-spam email (See Table 2).

Table 2. Spam filtering confusion matrix.

	Positive	**Negative**
Positive	Spam	Spam Email classified as Non-Spam
Negative	Non-Spam Email but classified as Spam	Non-Spam

Although relating to Kumar et al. [47], precision and recall can be used to mitigate the incorrect classifications from the first model and then implement said corrections to the newer models along with applying weights to help with strengthening the weaker areas of the regression.

4.5. Packer Detection

Packers are widely used tools to ensure that botnets would have obfuscation from signature detection and IDS and IPS. To counteract the use of packers and ensure that the malware is detected, Devi and Nandi [61] used a concept referred to as the Size of Code (SOC). SOC is defined as being the use of calculating the individual sizes of all the sections that would exist in the malware packet's code [62]. This means that since the malware code is packed it would cause the code to be split to a certain size. Therefore, calculating the packet sizes would cause the packer to become unpacked which is the opposite of the Botnet data being packed. Devi and Nandi [61] extended this by checking what the uninitialized data within the packed malware is. Since the packed malware is compressed, it would cause difficulty for the firewalls to detect the botnet code as they may be configured to detect as malware code would originally contain a large quantity of data being stored within the packets. Furthermore, since packers would intend on using missing data Fuzzy logic can also assist with filling in the missing values to further conduct the classifications on the presence of malware packer's [55].

5. Prevention and Mitigation Strategies

Relating to Feily et al. [16] in terms of botnet infections originating from phishing-based attacks, Alexander and Wanner [63] mentions that social engineering attacks can be avoided by providing staff with training on how to respond to different social engineering-based threats. Alexander and Wanner [63] defines this as cyber-attacks that depend on deceiving human users. The training procedures, discussed by Saleem and Hammoudeh [64], are essential to prevent phishing-based attacks as providing user training on how to respond properly to unknown emails. Bhandari [65] and Alexander and Wanner [63] discuss methods to ensure links that have been emailed to users are checked and not accessed. However, Bhandari [65] highlighted an issue that the source does not go into further details on the prevention side of botnets. However, the use of email attachments needs to be examined before access [63].

Another aspect highlighted by Bhandari [65] is that the open ports that the system does not use must be closed. This relates to Tunggal [8] as bot-masters would use open ports to be able to access the network hosts. This means that all unused ports should be closed and only allow one that is secure and needed to communicate with other devices to be left open. In the case of brute-force attacks, one such method that is mentioned by both Kirushnaamoni [66] and Wang et al. [67] are implementing the use of account locking in sessions. This is where password insertion for authentication is given a limited number of times. The maximum would usually be set to three attempts for users to authenticate themselves before they are forced out of the session. In the case of botnets, those that would attempt to access SSH sessions would only be able to use three insertions, allowing for sessions to be more secure and decreasing the chances of being vulnerable to brute-force attacks. Although, this comes with the drawback of limiting the attempts for legitimate users on the network if they would forget the SSH passwords. Another measure would be to use secure passwords by using a mix of letters, numbers and special characters at a specific length, thereby it can also mitigate dictionary attacks as strong passwords may not be actual words [65]. This is validated by Kirushnaamoni [66] as it provided a similar idea of using an attempt limit on the sessions. Although [66] also has a unique aspect of mentioning the use of providing a delayed response. Lengthening the duration of the password check on the system can also ensure that the botnet conducting the brute force attack will not be able to keep attempting at a rapid rate.

Bhandari [65] suggests ensuring that updated firewalls are implemented into networks. Botnets can cause anomalies into standard traffic flow within the targeted network, which would mean that there would be suspicious packets within the traffic [3,25]. Therefore, configuring the firewalls would allow for the suspicious packets to be analysed. If the packet contains malware code, the security system filters the packet and denies access to the host it is trying to enter. Although this comes with the drawback of networks placing firewalls in the wrong areas and would allow the bot-master to use the vulnerable parts to perform the exploitation [68]. This can also coincide with the vulnerability of security misconfiguration, meaning that the security needs to be implemented properly and applying the firewalls correctly to ensure that botnet attacks, and cyber-attacks in general, do not harm the system.

Botnets are likely to traverse the network through using open services that will not be used, in the targeted network such as using UDP protocols along with using ports of a specific protocol [69]. Therefore the mitigation, in this case, is to ensure that only ports that are necessary to the network are left open. Furthermore, Gupta et al. [69] suggests the use of ensuring regular security system and firewall updates to ensure that traffic that contains botnet code is denied access. This can be further justified by applying Access Control Lists (ACLs), a series of lists that define how traffic can flow through the network within Local Area Network (LAN). The lists would categorize if the computers had permission to communicate with certain devices or denied [10]. ACLs have similar functionalities to firewalls due to capabilities for traffic permission, and denial [69]. Despite firewall capabilities, it is important to ensure updates, as botmasters can create variations in their malware coding to penetrate the firewalls and elude detection. Vayansky and Kumar [70] suggests using filtering for both emails from outside the company. This will ensure that the implication of infected emails will not be sent to any host on the network along with filtering infected phishing sites. Vayansky and Kumar [70] also justifies staff training to ensure that the proper measures are taken if emails have been sent to staff members.

6. Machine Learning Algorithms Comparisons

Tuan et al. [71] used the UNBS-NB data-set to make accurate comparisons between SVM, K-means and Decision Tree classifier algorithms, from the results that have been gathered by Tuan et al. [71] methodology, it shows that the K-Means algorithm had the most accuracy in classifying botnets within the data-set being evaluated at “94.78” which relating to Kaushik [42], the data-set was properly cleaned, and the algorithm was able to use the features to differentiate botnets packets from the regular packets properly. The second most accurate algorithm was Decision Tree’s, which had the accuracy of “94.43” based on Tuan et al. [71] methodology. However, the Decision Tree classifier has a sensitivity rating of “94.52” for the data set, which is higher than the clustering algorithms being set to “89.78” for the data sensitivity. On the other hand, SVM, has the lowest accuracy of the three, where it has an accuracy score of “84.32” and sensitivity of “99.08” making SVM have the highest sensitivity between the three algorithms. The pattern that Tuan et al. [71] identifies from the results table is that the higher the accuracy of the algorithms would mean that there would be more sensitivity present. Although, Nguyen et al. [72] argued that Decision Tree classifiers are the most accurate from their prediction scores. However the main contrast between both [71], and ref. [72] is that Random Forest was considered within [72] approach which showed the highest accuracy, the accuracy further improves once the optimized model proposed by Nguyen et al. [72] achieved "99.04" accuracy. Although since Random Forest is an ensemble variant for Decision Tree, the accuracy would justify the effectiveness of decision tree classifiers.

7. Conclusions

From the evidence gathered, the primary causes of botnets being able to penetrate network systems are through both phishing attacks and the use of brute-forcing sessions within packet transition. This means that to ensure the reduction of the risk for botnets, it is important for the network administrators to equip firewalls against botmaster's variations on the malware code and update firewalls constantly. It is also important for IDS and IPS to be implemented if the botnet was able to penetrate the firewalls. Furthermore, password strength will also need to be considered for SSH sessions. Using long passwords with encryption can assure that sessions would not be cracked and allow the bot-master to conduct an insertion throughout the host communication. ACLs are useful as they can enable botnets to have a limited propagation on hosts and isolate the infected hosts to ensure that machines with more sensitive information and data would be more secure from any other related attacks. Phishing attacks can be prevented via staff training on how to appropriately respond to emails as well as links that originate from unknown sources. For further confirmation on emails being sent to them, staff can ask the sender personally if they have sent an email to them. Other measures that can be taken are email and website filtering. AI has also been shown to have a key role in detecting infections using fuzzy logic. It would be able to consider missing binary values within data packets during traffic during flow time to detect the presence of malicious code by using its decision-making capabilities. This would also correlate to packers, as they can allow the botnet code to be compressed, causing the data to have missing values. Furthermore, the IDS and IPS can ensure whether the bot-master can penetrate the network. The IDS can alert administrators to the bot-master accessing the network and allow the IPS to ensure that the bot-master is removed from the network. Through using aspects such as fuzzy logic and ML-based IDS/IPS, AI can contribute to the network and IoT security through protecting them from botnets or malware threats in general. Moreover, research results that leveraged machine learning for botnet detection show that some ML methods and algorithms like Deep Neural Networks, Decision Trees and K-Means function more effectively with better prediction accuracy. In contrast, other algorithms like SVM are less efficient.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhang, X.; Upton, O.; Beebe, N.L.; Choo, K.K.R. IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 300926. [CrossRef]
2. Kabay, M. Kraken the Botnet: The Ethics of Counter-Hacking. 2009. Available online: https://www.networkworld.com/article/2265704/kraken-the-botnet--the-ethics-of-counter-hacking.html (accessed on 30 December 2021).
3. Chen, S.C.; Chen, Y.R.; Tzeng, W.G. Effective Botnet Detection Through Neural Networks on Convolutional Features. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 372–378. [CrossRef]
4. Prowell, S.; Kraus, R.; Borkin, M. CHAPTER 1—Denial of Service. In *Seven Deadliest Network Attacks*; Prowell, S., Kraus, R., Borkin, M., Eds.; Syngress: Boston, MA, USA, 2010; pp. 1–21. [CrossRef]
5. Silva, S.S.; Silva, R.M.; Pinto, R.C.; Salles, R.M. Botnets: A survey. *Comput. Netw.* **2013**, *57*, 378–403. [CrossRef]
6. Cope, J. What's a Peer-to-Peer (P2P) Network? 2002. Available online: https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html (accessed on 30 December 2021).
7. Cooke, E.; Jahanian, F.; McPherson, D. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. *SRUTI* **2005**, *5*, 6.
8. Tunggal, A.T. What is an Open Port?: Definition and Free Checking Tools for 2021: UpGuard. 2021. Available online: https://www.upguard.com/blog/open-port (accessed on 30 December 2021).
9. Abbas, S.G.; Hashmat, F.; Shah, G.A.; Zafar, K. Generic signature development for IoT Botnet families. *Forensic Sci. Int. Digit. Investig.* **2021**, *38*, 301224. [CrossRef]
10. Liu, D.; Barber, B.; DiGrande, L. CHAPTER 9—Access Control Lists. In *Cisco CCNA/CCENT Exam 640-802, 640-822, 640-816 Preparation Kit*; Liu, D., Barber, B., DiGrande, L., Eds.; Syngress: Boston, MA, USA, 2009; pp. 331–384. [CrossRef]
11. Hanna, K.T. What Is Network Flooding and How Does It Work? 2021. Available online: https://www.techtarget.com/searchnetworking/definition/flooding (accessed on 30 December 2021).
12. Garre, J.T.M.; Pérez, M.G.; Ruiz-Martínez, A. A novel Machine Learning-based approach for the detection of SSH botnet infection. *Future Gener. Comput. Syst.* **2021**, *115*, 387–396. [CrossRef]
13. Chakraverty, S.; Goel, A.; Misra, S. *Towards Extensible and Adaptable Methods in Computing*; Springer: Berlin/Heidelberg, Germany, 2018.
14. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
15. Jeong, O.R.; Kim, C.; Kim, W.; So, J. Botnets: Threats and responses. *Int. J. Web Inf. Syst.* **2011**, *7*, 6–17. [CrossRef]
16. Feily, M.; Shahrestani, A.; Ramadass, S. A survey of botnet and botnet detection. In Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Greece, 18–23 June 2009; pp. 268–273.
17. Hong, J. The state of phishing attacks. *Commun. ACM* **2012**, *55*, 74–81. [CrossRef]
18. Khonji, M.; Iraqi, Y.; Jones, A. Phishing detection: A literature survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2091–2121. [CrossRef]
19. Jia, Y.; Chen, Y.; Dong, X.; Saxena, P.; Mao, J.; Liang, Z. Man-in-the-browser-cache: Persisting HTTPS attacks via browser cache poisoning. *Comput. Secur.* **2015**, *55*, 62–80. [CrossRef]
20. Scott, B. What Is a Dictionary Attack? 2020. Available online: https://www.techtarget.com/searchsecurity/definition/dictionary-attack (accessed on 30 December 2021).
21. Nam, J.; Choo, K.K.R.; Kim, M.; Paik, J.; Won, D. Dictionary attacks against password-based authenticated three-party key exchange protocols. *KSII Trans. Internet Inf. Syst. (TIIS)* **2013**, *7*, 3244–3260.
22. Mitchell, B. What Is a Network Sniffer? 2021. Available online: https://www.lifewire.com/definition-of-sniffer-817996 (accessed on 30 December 2021).
23. Miller, M. What's the Difference between Offline and Online Password Attacks? 2021. Available online: https://www.triaxiomsecurity.com/whats-the-difference-between-offline-and-online-password-attacks (accessed on 30 December 2021).
24. Satoh, A.; Nakamura, Y.; Ikenaga, T. A flow-based detection method for stealthy dictionary attacks against Secure Shell. *J. Inf. Secur. Appl.* **2015**, *21*, 31–41. [CrossRef]
25. Karim, A.; Salleh, R.B.; Shiraz, M.; Shah, S.A.A.; Awan, I.; Anuar, N.B. Botnet detection techniques: Review, future trends, and issues. *J. Zhejiang Univ. Sci. C* **2014**, *15*, 943–983. [CrossRef]
26. Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
27. Hayes, M.A.; Capretz, M.A. Contextual anomaly detection framework for big sensor data. *J. Big Data* **2015**, *2*, 1–22. [CrossRef]
28. Chen, R.; Niu, W.; Zhang, X.; Zhuo, Z.; Lv, F. An effective conversation-based botnet detection method. *Math. Probl. Eng.* **2017**, *2017*, 4934082. [CrossRef]
29. Rahim, A.; Bin Muhaya, F.T. Discovering the botnet detection techniques. In *Security Technology, Disaster Recovery and Business Continuity*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 231–235.
30. García, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. *Comput. Secur.* **2014**, *45*, 100–123. [CrossRef]

31. Lavin, A.; Ahmad, S. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 9–11 December 2015; pp. 38–44.
32. Alhajri, R.; Zagrouba, R.; Al-Haidari, F. Survey for anomaly detection of IoT botnets using machine learning auto-encoders. *Int. J. Appl. Eng. Res.* **2019**, *14*, 2417–2421.
33. Jordan, J. Introduction to Autoencoders. 2018. Available online: https://www.jeremyjordan.me/autoencoders/ (accessed on 30 December 2021).
34. Ashraf, J.; Keshk, M.; Moustafa, N.; Abdel-Basset, M.; Khurshid, H.; Bakhshi, A.D.; Mostafa, R.R. IoTBoT-IDS: A Novel Statistical Learning-enabled Botnet Detection Framework for Protecting Networks of Smart Cities. *Sustain. Cities Soc.* **2021**, *27*, 103041. [CrossRef]
35. Mahmoud, M.S.; Xia, Y. Chapter 9—Cyberphysical Security Methods. In *Networked Control Systems*; Mahmoud, M.S., Xia, Y., Eds.; Butterworth-Heinemann: Oxford, UK, 2019; pp. 389–456. [CrossRef]
36. Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. *Appl. Soft Comput.* **2018**, *72*, 79–89. [CrossRef]
37. Lawal, M.A.; Shaikh, R.A.; Hassan, S.R. An anomaly mitigation framework for iot using fog computing. *Electronics* **2020**, *9*, 1565. [CrossRef]
38. Ippolito, P.P. Feature Extraction Techniques. 2019. Available online: https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be (accessed on 30 December 2021).
39. Tobiyama, S.; Yamaguchi, Y.; Shimada, H.; Ikuse, T.; Yagi, T. Malware detection with deep neural network using process behavior. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; Volume 2, pp. 577–582.
40. Laskowski, N.; Contributor, T. What Are Recurrent Neural Networks and How Do They Work? 2021. Available online: https://www.techtarget.com/searchenterpriseai/definition/recurrent-neural-networks (accessed on 30 December 2021).
41. Firdausi, I.; Lim, C.; Erwin, A.; Nugroho, A.S. Analysis of machine learning techniques used in behavior-based malware detection. In Proceedings of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, Jakarta, Indonesia, 2–3 December 2010; pp. 201–203.
42. Kaushik, S. An Introduction to Clustering and Different Methods of Clustering. 2016. Available online: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/ (accessed on 30 December 2021).
43. Seither, J. Anomaly Detection: (Dis-)advantages of k-Means Clustering. 2017. Available online: https://www.inovex.de/de/blog/disadvantages-of-k-means-clustering/ (accessed on 30 December 2021).
44. Zhang, X.; Gu, C.; Lin, J. Support Vector Machines for Anomaly Detection. In Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; pp. 2594–2598.
45. Afonja, T. Kernel Functions. 2018. Available online: https://towardsdatascience.com/kernel-function-6f1d2be6091 (accessed on 30 December 2021).
46. Statinfer. 204.6.8 SVM: Advantages Disadvantages and Applications. 2019. Available online: https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/ (accessed on 30 December 2021).
47. Kumar, B.J.; Naveen, H.; Kumar, B.P.; Sharma, S.S.; Villegas, J. Logistic regression for polymorphic malware detection using ANOVA F-test. In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 17–18 March 2017; pp. 1–5.
48. Brownlee, J. How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. 2020. Available online: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/ (accessed on 30 December 2021).
49. Butler, B. What Is Fog Computing? Connecting the Cloud to Things. 2018. Available online: https://www.networkworld.com/article/3243111/what-is-fog-computing-connecting-the-cloud-to-things.html (accessed on 30 December 2021).
50. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 13–17 August 2012; pp. 13–16.
51. George, A.; Dhanasekaran, H.; Chittiappa, J.; Challagundla, L.; Nikkam, S.; Abuzaghleh, O. Internet of Things in health care using fog computing. In Proceedings of the 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 4 May 2018; pp. 1–6. [CrossRef]
52. Tsikerdekis, M.; Zeadally, S.; Schlesener, A.; Sklavos, N. Approaches for preventing honeypot detection and compromise. In Proceedings of the 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 23–25 October 2018; pp. 1–6.
53. Mukherjee, L. What Is a Honeypot in Network Security? Definition, Types and Uses. Available online: https://sectigostore.com/blog/what-is-a-honeypot-in-network-security-definition-types-uses (accessed on 30 December 2021).
54. Joshi, C.; Ranjan, R.K.; Bharti, V. A Fuzzy Logic based feature engineering approach for Botnet detection using ANN. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *in press*. [CrossRef]
55. Chai, W. What Is Fuzzy Logic? 2021. Available online: https://www.techtarget.com/searchenterpriseai/definition/fuzzy-logic (accessed on 30 December 2021).
56. Moghimi, M.; Varjani, A.Y. New rule-based phishing detection method. *Expert Syst. Appl.* **2016**, *53*, 231–242. [CrossRef]

57. Abu-Nimeh, S.; Nappa, D.; Wang, X.; Nair, S. A comparison of machine learning techniques for phishing detection. In Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, Pittsburgh, PA, USA, 4–5 October 2007; pp. 60–69.
58. Kaspersky. All about Phishing Scams and Prevention: What You Need to Know. 2021. Available online: https://www.kaspersky.com/resource-center/preemptive-safety/phishing-prevention-tips (accessed on 30 December 2021).
59. Guzella, T.S.; Caminhas, W.M. A review of machine learning approaches to spam filtering. *Expert Syst. Appl.* **2009**, *36*, 10206–10222. [CrossRef]
60. Dedeturk, B.K.; Akay, B. Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. *Appl. Soft Comput.* **2020**, *91*, 106229. [CrossRef]
61. Devi, D.; Nandi, S. Detection of packed malware. In Proceedings of the First International Conference on Security of Internet of Things, Kollam, India, 17–19 August 2012; pp. 22–26.
62. Bergenholtz, E.; Casalicchio, E.; Ilie, D.; Moss, A. Detection of metamorphic malware packers using multilayered LSTM networks. In Proceedings of the International Conference on Information and Communications Security, Copenhagen, Denmark, 24–26 August 2020; pp. 36–53.
63. Alexander, M.; Wanner, R. Methods for understanding and reducing social engineering attacks. *SANS Inst.* **2016**, *1*, 1–32.
64. Saleem, J.; Hammoudeh, M. Defense methods against social engineering attacks. In *Computer and Network Security Essentials*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 603–618.
65. Bhandari, P. Botnet Detection and Prevention Techniques: A Quick Guide. 2021. Available online: https://www.xenonstack.com/insights/what-are-botnets (accessed on 30 December 2021).
66. Kirushnaamoni, R. Defenses to curb online password guessing attacks. In Proceedings of the 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 21–22 February 2013; pp. 317–322.
67. Wang, X.; Yan, Z.; Zhang, R.; Zhang, P. Attacks and defenses in user authentication systems: A survey. *J. Netw. Comput. Appl.* **2021**, *188*, 103080. [CrossRef]
68. Klein, D. Relying on firewalls? Here's why you'll be hacked. *Netw. Secur.* **2021**, *2021*, 9–12. [CrossRef]
69. Gupta, B.B.; Joshi, R.C.; Misra, M. Distributed denial of service prevention techniques. *arXiv* **2012**, arXiv:1208.3557.
70. Vayansky, I.; Kumar, S. Phishing—Challenges and solutions. *Comput. Fraud. Secur.* **2018**, *2018*, 15–20. [CrossRef]
71. Tuan, T.A.; Long, H.V.; Son, L.H.; Kumar, R.; Priyadarshini, I.; Son, N.T.K. Performance evaluation of Botnet DDoS attack detection using machine learning. *Evol. Intell.* **2020**, *13*, 283–294. [CrossRef]
72. Nguyen, G.L.; Dumba, B.; Ngo, Q.D.; Le, H.V.; Nguyen, T.N. A collaborative approach to early detection of IoT Botnet. *Comput. Electr. Eng.* **2022**, *97*, 107525. [CrossRef]