Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6640499, 24 pages
https://doi.org/10.1155/2021/6640499

Review Article

Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation

Ying Xing,[1,2] **Hui Shu**,[1] **Hao Zhao**,[1] **Dannong Li**,[3] **and Li Guo**[2]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, University of Information and Engineering, Zhengzhou 450000, China*
[2]*Software College, Zhongyuan University of Technology, Zhengzhou 450000, China*
[3]*Teaching and Research Support Center, PLA Strategic Support Force Information Engineering University, Zhengzhou 450000, China*

Correspondence should be addressed to Hui Shu; shuhui123@126.com

Received 17 November 2020; Revised 12 March 2021; Accepted 31 March 2021; Published 15 April 2021

Academic Editor: Jude Hemanth

With the continuous evolution of the Internet, as well as the development of the Internet of Things, smart terminals, cloud platforms, and social platforms, botnets showing the characteristics of platform diversification, communication concealment, and control intelligence. This survey analyzes and compares the most important efforts in the botnet detection area in recent years. It studies the mechanism characteristics of botnet architecture, life cycle, and command and control channel and provides a classification of botnet detection techniques. It focuses on the application of advanced technologies such as deep learning, complex network, swarm intelligence, moving target defense (MTD), and software-defined network (SDN) for botnet detection. From the four dimensions of service, intelligence, collaboration, and assistant, a common bot detection evaluation system (CBDES) is proposed, which defines a new global capability measurement standard. Combing with expert scores and objective weights, this survey proposes quantitative evaluation and gives a visual representation for typical detection methods. Finally, the challenges and future trends in the field of botnet detection are summarized.

1. Introduction

A botnet is an overlay network formed by many hosts (bots or zombies) infected by bots and controlled by an attacker (botmaster) for the purpose of malicious activities [1, 2]. Botmaster can control the server to control the bot and initiate various types of cyberattacks, such as distributed denial of service (DDoS), spam, phishing, click fraud, and information theft, which is one of the most serious security threats facing the Internet [3].

Given the security problems due to the continuous development of botnets, accurately identifying and detecting botnets, particularly unknown botnets in the incubation period, are the main challenging issues in academic and industrial research. Firstly, the C&C mechanism of botnets shows diversified and intelligent characteristics. Public service resources such as 5G, Internet of Things, smart terminals, cloud platforms, and social platforms have gradually emerged as fertile ground for botnets. Botnets use technologies such as zero-day vulnerabilities, P2P networks, phishing, fast flux, anonymous networks, bitcoin networks, and lightning networks as their means of utilization and spread [4–6]. Secondly, compared with conventional network security threats, botnets spread faster, have more infection channels, are more concealed, have a higher technical content, and have greater destructive power. Finally, because botnets are mostly in the silent state, they only maintain the connection state through C&C channels, without attacking and intruding, and often do not have conventional attack characteristics. Therefore, most intrusion detection systems cannot effectively identify botnets.

Deep learning theory has been rapidly developed, with significant advancements in related theoretical research and practical applications, particularly in speech recognition [7] and image recognition [8], etc. Deep learning methods can be used to solve conventional zombies. The low accuracy problem

in the case of multiclassification task detection and the complexity of feature engineering in network detection technology have become research hotspots. The characteristics of blockchain technology such as decentralization, anticensorship, and concealment, as well as smart contracts, digital signatures, and incentive mechanisms, provide a new paradigm for the construction of botnets and distributed detection. The community mining algorithm in the complex network discipline provides new ideas for behavior-based botnet analysis. Swarm intelligence algorithms, SDN, MTD, and integrated methods, are some of the new methods for botnet detection.

In the field of botnet detection in recent years, there is a lack of a comprehensive overview on the latest detection technologies. This survey is divided into six major parts: first, we analyze previous surveys; second, we study botnet background and new development of botnet construction mechanism; third, we classify botnet detection technologies from a new perspective; fourth, we analyze the latest and most advanced botnet detection technologies; fifth, we propose the common bot detection evaluation system (CBDES); sixth, we summary the challenges and future trends in the field of botnet detections.

The main contributions of this article are as follows:

(1) A novel summary of new developments in the construction mechanism of botnets

(2) A novel classification of botnet detection technologies

(3) A comprehensive analysis of the latest and most advanced botnet detection technologies, such as deep learning, complex networks, swarm intelligence, SDN, MTD, and blockchain.

(4) A common bot detection evaluation system is proposed, from the four dimensions of service, intelligence, collaboration, and assistant, drawing on the ideas of analytic hierarchy process (AHP)

(5) A new global capability metric € is defined, combined with expert scores and objective weights, to quantitatively evaluate eight typical detection methods and use spider diagrams to give a visual representation

The rest of the paper is organized as follows. Section 2 describes and analyzes previous botnet detection surveys. Section 3 studies botnet background and new development of botnet construction mechanism. Section 4 proposes a novel classification of botnet detection methods. Section 5 analyzes the most the latest and most advanced detection technologies. Section 6 proposes the common bot detection evaluation system. Section 7 discusses the challenges and prospects of the area, and Section 8 presents the conclusions.

2. Previous Surveys

There have been several surveys on botnet detection techniques in recent years, which are analyzed in this section.

The IoT botnet detection technologies are divided into host-based and network-based in [9]. Network-based detection is further divided into signature-based, DNS-based, traffic-based, anomaly-based, and mining-based methods. However, this review is not comprehensive enough because it targets one dimension of IoT botnet.

A detailed statistical analysis of IoT attack literature in recent years is summarized in [10]. The review outlines the existing proposed contributions, datasets utilized, network forensic methods utilized, and research focus of the primary selected studies. But it does not introduce the specific detection technology and compare and analyze the detection methods.

DNS-based botnet detection technologies are classified into five categories in [11]: flow-based, anomaly-based, flux-based, DGA-based, and bot infection-based. Essential attributes of a smart DNS-based botnet detection system are proposed. But the survey did not provide context for the botnet's construction mechanism.

A comprehensive botnet detection is analyzed in [12]. This survey classifies botnet detection techniques into four classes: signature-based, anomaly-based, DNS-based, and mining-based. Unfortunately, the summary is too simple and does not cover the introduction of the latest technology.

For botnet detection technologies based on DNS traffic analysis, the technologies are classified into two categories in [13]: honeypot-based and IDS-based. It mainly introduces passive technologies, including graph theory, statistical analysis, clustering, decision tree, and neural network. This literature is comprehensive, but it is old and has not been evaluated.

Evasion and detection techniques of DNS-based botnets are focused on [14]. This survey introduces Fast-Flux and DGA botnet detection technology. Also, the dimensions are relatively single and there is no evaluation.

The detection is divided into four categories in [15]: honeypot analysis, communication signature, anomaly, and log. The literature introduction content is relatively few, not comprehensive enough.

Each survey emphasized different aspects of the papers. The analysis of the surveys shows some limitations:

The surveys use different taxonomies and terminologies

A lot of surveys focus on one type or function, such as DNS and IoT, with a single dimension and lack of comprehensive analysis of new construction mechanism of botnets

Most surveys do not cover the most advanced method, and there is a lack of systematic introduction to the latest technologies

Most of the data lack a comprehensive evaluation of the detection methods

A comparison of our survey with other surveys is presented in Table 1. Our survey aims at knowing and understanding botnet detection and eliminates these limitations.

3. Background

Based on an in-depth understanding of the working mechanism and behavior characteristics of botnets, this section introduces the latest development in botnet

Table 1: Comparison with other surveys.

Survey	Published time	Detection targeted	Background	Detection methods/techniques	Evaluation
[9]	2020	IoT	(i) Architecture (ii) Life cycle	Neural networks data mining graph theory	(i) Not included
[10]	2020	IoT	(i) Not included	Machine learning Deep learning Statistical analysis Propagation model	(i) Measurement
[11]	2019	DNS	(i) Not included	Machine learning statistical analysis Whitelist/blacklist	(i) Not included
[12]	2018	Universal	(i) Architecture (ii) Life cycle	Signature-based Mining-based Graph theory	(i) Not included
[13]	2015	DNS	Life cycle	Statistical analysis Clustering Decision tree Neural network	(i) Not included
[14]	2017	DNS	C&C channel	Characteristics analysis statistical analysis	(i) Not included
[15]	2016	Universal	(i) Architecture (ii) Life cycle	Honeypot analysis statistical analysis	(i) Not included
Our method	—	Universal	(i) Architecture (ii)Life cycle (iii)C&C channel	Deep learning, complex network, swarm intelligence, MTD, SDN, blockchain, etc.	Common bot detection evaluation system

construction mechanisms in terms of the botnet architecture, life cycle, and C&C channel, as shown in Figure 1.

3.1. Architecture. The botnet C&C system architecture is mainly divided into the following three categories: centralized, distributed, and hybrid.

(1) *Centralized*. The centralized botnet architecture adopts a client-server model generally. The bot mainly obtains control commands from the control server in a polling manner, and the botmaster sends the control commands and resources to the zombie host through these servers. Centralized botnets have advantages such as simple implementation, high efficiency, and good coordination, but their control process is associated with a central node failure.

(2) *Distributed*. To improve the robustness of botnet, an attacker can use a decentralized structure of the P2P (peer to peer) mode as its channel architecture. Any node can act as a client and a server simultaneously, and the communication process does not rely on public network reachable server resources. Although the P2P botnet command issuance delay is higher than that of a centralized structure, the distributed structure is difficult to be hijacked, measured, and closed.

(3) *Hybrid*. A hybrid architecture typically means that botnets have both central and P2P structure. This can be divided into two categories: an overall central structure and a partial P2P structure. From an overall perspective, it still belongs to the centralized structure of the C/S model; however, a P2P structure is present between the service nodes. One is the overall P2P structure and the other is a local center structure. This type of network structure is conducive to the realization of regional differentiation management and control, and it is difficult for defenders to detect all the key nodes and the overall scale of the botnet.

3.2. Life Cycle. The life cycle of a botnet mainly includes the stages of propagation, rally, interaction, and malicious activities.

Propagation. As an independent runnable program, the spread of bots includes the use of conventional malicious code. The main propagation methods include shared media spread, vulnerability exploitation spread, social engineering spread, and password guessing.

Rally. Rally refers to the behavior of bots in locating and controlling the server and its resources. Implementation methods are mainly divided into two categories: static and dynamic. Static addressing means that the C&C resources that bots try to access are static and unchanging. These resources are typically hard-coded in the body of the bot or stored in a hidden path of the infected machine, such as the registry. Dynamic addressing means that the access address is not fixed

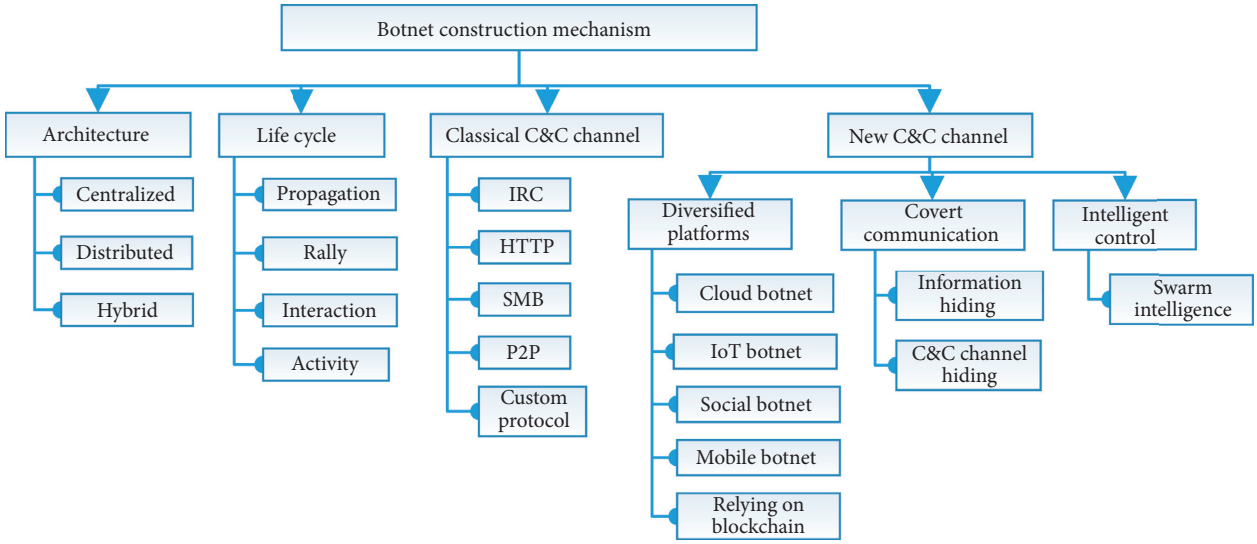

Figure 1: Botnet construction mechanism.

but needs to be dynamically generated based on the specific algorithm used.

Interaction. When the zombie host successfully discovers the available command control server or resource, it will establish a connection with the controller and begin to interact. This process is also called the command control phase, which mainly includes four activities: registration, file download, order distribution, and result feedback.

Malicious Attack. The main purpose of an attacker to build and control a botnet is to control many victim hosts to launch a variety of attacks. Common attack activities include DDoS, spam, spreading malware, information leakage, click fraud, phishing attacks, information collection, virtual currency mining, and encrypted blackmail [16].

A product development model was used to define the life cycle in [17], including concepts, recruitment, interaction, motivation, and attack execution (CRIME). Literature [18] proposed a fine-grained, hidden Markov model-based botnet life cycle model, describing the state transition of botnets from propagation to extinction and dividing the typical botnet life cycle into nine types of hidden states: infection/initialization/idle/propagation/attack/maintenance/offline/isolation/dead. The model used "state" instead of "stage" to describe the evolution of botnets and broke the conventional irreversible and abstract timing relationship. The model could better represent the migration and changes of botnets.

3.3. Command and Control Channel

3.3.1. Classical C&C Channel. The core of the botnet is communication, and the classical C&C channel is mainly implemented through IRC, HTTP, SMB, P2P, or other custom protocols [2]. Using IRC service as a centralized C&C channel is easy to implement and has low latency and good real-time performance; however, the centralized topology can easily be detected and blocked [19]. Bots that use the HTTP protocol to construct C&C channels can periodically access the botnet controller, obtain command files, parse, and perform corresponding operations, and can penetrate IDS and firewalls, with good versatility and concealment [20]. The Server Message Block (SMB) protocol is a protocol that hides communication under the typical traffic patterns in home and enterprise networks and is mainly used for communication in local area networks [21, 22]. The P2P protocol is used to construct distributed botnet control channels, which solves the single-point failure problem of botnet controllers, and has good robustness, stealthiest, and self-organization capabilities. The disadvantage is that it is vulnerable to index poisoning and Sybil attacks and initial vulnerability. Botnets that use custom protocols to communicate are stealthier, and the communication process is less likely to be detected.

3.3.2. New C&C Channel

(1) Diversified Platforms. The decentralization and concealment of public service resources represented by cloud platforms, social networking sites, and blockchains present natural advantages and have become fertile ground for botnets.

Cloud Botnet. The multitenant feature of cloud computing can provide computing resources to anyone. Botnet controllers pretend to be legitimate tenants of cloud services and use virtual machines of cloud service providers to quickly construct botnets and use them to launch attacks. At the RSA2014 conference, Ragan introduced a cloud botnet construction method [23], which can realize the mining of electronic money by controlling massive cloud computing resources.

IoT Botnet. The Internet of Things (IoT) has been implemented in various fields such as agriculture,

healthcare, food supply management, drug supply management, environmental monitoring, and smart homes. IoT has heterogeneous environments and resource-constrained devices, i.e., low memory, low computing power, and low original security performance, which increases the risk of infection. Mirai is a common IoT botnet, the main objective of which is to perform DDOS attacks, with a strong scale and attack capability [24–26].

Social Botnet. Botnets use social media sites such as Facebook, Twitter, or WeChat to build transmission channels or spread messages in social networks. The Flashback botnet [27] uses Twitter to construct a backup C&C channel. Once the main channel fails, the bot will search for the C&C domain name by searching for a specific identifier that is dynamically generated to restore communication with the controller. A botnet that is parasitic on social networking sites can imitate normal users to complete a variety of online social actions [28]. Attackers can control social bots to achieve rumor dissemination, advertisement push, and personal information collection [29].

Mobile Botnet. The portability of mobile devices and the increasing popularity of applications also have an impact on the threat pattern of botnets. Geinimi can steal IMEI [30], geographic location, SMS, address book, and other information and send spam SMS and install malware; in 2012, Dexter, the first POS machine botnet, uses a memory reading technology to steal users' payments card data; Zhao et al. [31] proposed a mobile botnet network based on Google Cloud to Device Message (C2DM).

Based on Blockchain. The literature [32] uses bitcoin blockchain floating C&C servers to propose a new type of resilient botnet. The literature [33] uses the bitcoin transaction propagation mechanism as the C&C infrastructure and proposes the use of subliminal channels [34, 35] to create a concealed method to repeatedly create signatures on transactions. Fbot botnet, one of the Mirai variants, uses Emercoin [13] domain name system based on a distributed blockchain to solve a key problem that conventional DGA-based botnets are easily detected by reverse engineering. The literature [36] proposes a new generation of hybrid two-layer botnet LNBot, which uses lightning network (LN) infrastructure to communicate between bots and C&C servers. The off-chain concept [37, 38] realizes almost instant bitcoin transactions.

(2) Covert Communication. The covert communication technology mainly includes information hiding and C&C channel hiding.

Information Hiding. This method modulates the secret information into the protocol redundancy field through various modulation methods, mainly including encryption, compression, and obfuscation, and steganography. Nagaraja et al. [39] proposed a botnet to use existing social networks as C&C channels and hide communications in JPEG pictures. Cui et al. [40] proposed a three-channel botnet model. The core idea was to use Domain-Flux, URL-Flux, and Cloud-Flux as subchannel protocol for the entire C&C based on registration, command issuance, and data return functions.

The C&C Channel Hiding. The covert channel technology based on DNS protocol is one of the mainstream network covert channel realization methods. The common methods are Domain-Flux and Fast-Flux [41]. Casenove et al. [42] introduced scalable and stealthy botnets based on anonymous networks and could observe the C&C traffic at the Internet Service Provider (ISP) level.

(3) Intelligent Control. The concept of complex systems, with their self-organization, resilience, and adaptability, can greatly help botnet communication protocol and architecture design. A heuristic algorithm based on ant colony optimization was proposed to construct a botnet C&C [43]. This could ensure spontaneous and intelligent collaboration between independent bot agents, which improves network fault tolerance and the ability to dynamically adapt to the network environment.

4. Classification

Conventional detection methods are no longer suitable for new botnet detection. The industry has a more in-depth understanding of the working mechanism and behavior characteristics of botnets, and various botnet detection methods have been proposed. This section divides the key technologies of botnet detection into three categories based on honeypot analysis, communication signatures, and abnormal behavior. We focus on the application of deep learning, complex networks, swarm intelligence, MTD, SDN, blockchain, and other cutting-edge technologies in botnet detection.

Botnet detection technology classification standards are different, and there are multidimensional classification methods. This article classifies the key technologies as shown in Figure 2.

5. Methods

5.1. Based on Honeypot Analysis. Based on the honeypot analysis and detection method, many malicious code samples can be obtained through honeypot trapping, i.e., the botnet binary files of the existing botnet, and the monitoring and analysis can be performed in a controlled environment, and the bots and their malicious behaviors can be discovered [44]. It is an active detection behavior. Representative honeypots include Snort, Ntop, Argos, Nepenthes, Sebek, and the Goddess of Hunting project led by Peking University Zhuge Jianwei [45]. The darknet is quickly becoming a popular alternative to using honeypots and is essentially a derivative of the honeynet.

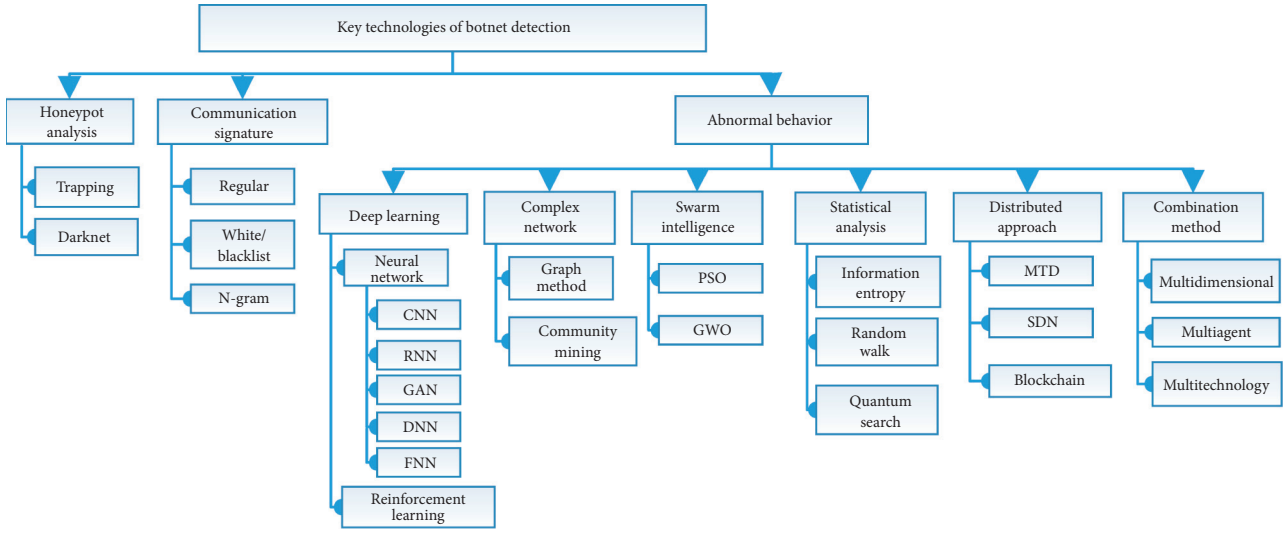

Figure 2: Key technologies of botnet detection.

Although the honeypot-based method has a high accuracy rate for known botnets, it cannot effectively identify encrypted traffic and detect unknown attacks. Moreover, it cannot easily find botnets that spread through social engineering and is useless for real-time systems. Because of the lack of user operations, it can be easily recognized by bots with an anti-honeypot function.

5.2. Based on Communication Signature. The method based on communication signature detection is a commonly used defense method, which detects bot activities based on predefined patterns and signatures retrieved from well-known bots [46]. Common methods include regular expressions [47], whitelists (or blacklists) [48], and N-gram models [49]. By configuring feature matching rules in advance, conventional intrusion detection systems such as Snort have a rich signature database, which can help quickly and accurately discover botnet activities.

The communication signature-based method is suitable for botnets with definite features, which helps further understand the communication mechanism and potential vulnerabilities of botnets. The disadvantage is that robots can avoid signature-based detection by using code obfuscation technology and cannot detect botnets with unknown features. The method needs to maintain and update the signature knowledge base continuously, which increases the cost of detection.

5.3. Based on Abnormal Behavior. Anomaly-based detection is an important research field in botnet detection. The basic idea is based on host behavior or network traffic abnormalities, such as the high network latency, large amounts of traffic, traffic on abnormal ports, and abnormal system behaviors and based on established systems. The deviation in the benign behavior or the similarity with the behavior of bots can be detected.

5.3.1. Deep Learning. In the past few decades, researchers have used various conventional machine learning methods to detect botnets [50–52] and have made great progress, such as Naive Bayes [53], support vector machines [54], random forests [55], and clustering algorithm (such as DBSCAN [56] and X-means [57]), based on a variety of characteristics to establish a model that can identify malicious network traffic. The characteristics are typically set by researchers through the experience before the model is established. Common dimensions include network flow properties, such as the number of data packets, the average byte of data packets, and the average interval between two adjacent data streams, and behavior, such as whether to access the same server. These detection models were found to have a low false-negative rate and false-positive rate in an experiment. However, there are some shortcomings. First, manual selection has higher requirements on the prior knowledge of the designer. The second is that fixed features also provide an opportunity for attackers. Attackers can use anti-machine learning ideas to change the characteristics of botnet traffic in a targeted manner, thereby evading model detection. The botnet shape and command control mechanism are gradually changing, and artificial feature selection is becoming increasingly difficult. With the rapid development of deep learning technologies, neural networks, reinforcement learning, knowledge graphs, and other methods are gradually being applied to the field of botnet detection, which represents new approaches.

(1) Neural Network. The basic idea is to extract network traffic features based on temporal and spatial similarities. This method involves mapping network traffic into a grayscale image or feature vector and sending it to a neural network model, extracting distinguishable features and patterns from the space and time dimensions, and automatically learning network traffic features.

CNN. The convolutional neural network (CNN) mainly learns spatial features from the spatial dimension through network traffic. Aiming at BotCloud, Guang et al. [58] first extracted basic features from the network stream and mapped them onto grayscale images. Subsequently, the

CNN algorithm LeNet-5 was used for feature learning, and more abstract features were extracted to express the hidden patterns and structural relationships in the network stream data; the algorithm was finally applied to detect BotCloud. There is no significant difference in traffic in the early stages of IoT botnet. Most botnet detection systems are not suitable for resource-constrained IoT devices. The literature [59] used side-channel power consumption information, such as power consumption, etc., to distinguish whether IoT devices are affected by malicious behaviors and proposed a CNN-based deep learning model to perceive the subtle differences in power consumption data. The literature [60] proposed an extensible framework that uses LSTM to collect DNS traffic data at the ISP level and detect DGA-based malware in real time. Deep learning ImageNet model was used to classify domain names generated by DGA [61].

RNN. Recurrent neural network (RNN) mainly learns the characteristics of network traffic in time series from the time dimension. The literature [62] applied RNN to detect botnets by modeling network communication behavior as a sequence of time-varying states. The behavior model of each flow is established based on four parameters: source and destination IP addresses, destination port, and protocol. The literature [63] proposed a solution to detect botnet activities in consumer IoT devices and networks, four attack vectors of Mirai were used as eigenvectors, and a detection model was established based on the RNN and bidirectional long- and short-term memory (BiLSTM-RNN). The literature [64] proposed an anomaly detection system for 5G adaptive real-time deep learning system, which includes two modules: abnormal symptom detection (ASD) and network anomaly detection (NAD). DBN (Dynamic Bayes) was used to realize the ASD time measurement process, and the LSTM network model was used to realize NAD. The literature [65] proposed a malicious domain name detection method based on knowledge graph. For DNS traffic, TransE took the embedded model of the system as input and completed the storage and representation of information in the knowledge graph, which not only included the embedding of entities and relationships, but also the embedding of attribute values. The advantage of combining the BiLSTM neural network to extract features for detection was that it can learn the context relationship of vectors in the sequence and better extract features for classification.

CNN + RNN. Reference [66] proposed a deep learning-based botnet detection system Bot Catcher, which extracted network traffic characteristics from the two dimensions of time and space automatically. Spatial feature learning was based on the application CNN LeNet-5 structure, which was used in the field of image recognition, and each stream data was converted into a 2D gray image. The data stream of 1024B (32×32) data was intercepted before each data stream. Typically, the data in front of a data stream mainly includes connection information (such as the three-way handshake in TCP connection and key exchange in TLS connection), and less part of the content exchange could better reflect the main characteristics of the entire data stream. To mine deeper into the characteristics of the data stream in the time series, Bot Catcher used the BiLSTM neural network in the RNN to learn the time characteristics and scanned each data stream in both forward and reverse directions. In [67], for Fast-Flux botnets, combined with convolutional neural network (DenseNet) and recurrent neural network (BiLSTM), the DNS data response packet in the network traffic based on analysis, a fast-flux botnet detection method based on the temporal and spatial characteristics of traffic was proposed.

GAN. The literature [68] proposed a botnet detection framework based on generative confrontation networks (Bot-GAN), which was different from other variants of generative confrontation networks. The framework focused more on discriminative models instead of generative models.

DNN. The literature [69] proposed a DGA domain name detection method that does not require extracting specific features, based on word-hashing technology to map strings to a high-dimensional space, and used deep neural network DNN to classify domain names. The literature [70] proposed a two-level deep learning framework for real-time detection of botnets. In the first-level framework, the Siamese network is used to estimate the similarity measure of DNS queries. In the second level, a domain generation algorithm (DGA) based on deep learning architecture is proposed to classify normal and abnormal domain names. The literature [71] constructed a deep learning framework based on dual-stream network (TS-ASRCaps), used multimodal information to reflect the characteristics of DGAs, and proposed an attention sliced recurrent neural network (ATTSRNN) to automatically mine the underlying semantics. A capsule network with dynamic routing (CapsNet) was used to model high-level visual information.

FNN. The literature [72] proposed a real-time online Fast-Flux botnet filtering system, aiming to improve the detection of unknown "zero-day" online fast-flux botnets. Fast-Flux botnet domain was distinguished from the legal domain in an online mode based on new rules, characteristics, or classes. Using the adaptive evolutionary fuzzy neural network algorithm, the first stage preprocessing includes feature analysis and stemming, and the A-Import feature represents the Fast-Flux botnet, and the second stage uses an evolutionary fuzzy neural network (EFUNN) algorithm to establish FF Hunter (FFH) system.

(2) Reinforcement Learning. Reinforcement learning is an algorithmic method used to solve sequential decision problems, in which agents (or decision makers) interact with the environment to learn to respond under different conditions. Reinforcement learning is used for botnet detection in three ways: firstly, combined with NN neural network for new feature extraction, the agent learns a strategy to maximize the total number of bots detected over time. Secondly, it is used for the deployment of distributed detectors to make intelligent decisions. The agent estimates the system status and operation rewards by monitoring the network activities in different network segments. Thirdly, deep reinforcement learning is used for evading machine learning detection.

The literature [73] proposed a method that combines reinforcement learning technology to detect botnets as early as possible in the propagation stage or before any malicious activities are initiated by the bot. It included four stages: network traffic capture and packet filtering, feature extraction, malicious activity detection, and bot behavior detection using reinforcement learning. The network traffic feature extraction was carried out in three levels: data packet level, data flow level, and connection level. Malicious activity detection included three stages: offline stage (training), online detection stage, and reinforcement learning stage. The literature [74] proposed an online clustering-enhanced botnet detection method using reinforcement learning PRCL, which could detect botnets in real time with high accuracy. Research on adversarial machine learning has shown that botnet attackers can bypass the detection model by constructing specific samples, and many algorithms are susceptible to less input disturbances. The literature [75] proposed a new anti-botnet traffic generator framework based on deep reinforcement learning (DRL), which could effectively generate reverse traffic flow through the RL algorithm and Markov decision process (MDP). The agent could add disturbances to the flow and changed the spatial and temporal properties of the network traffic and automatically added disturbances to the samples to try to fool the target detector. This research could help inspectors to find defects and improve the robustness of the system.

The summary of typical botnet detection techniques based on deep learning is shown in Table 2.

5.3.2. Complex Network. Botnet communication is associated with both similarity and stability. The relatively frequent communication activities based on the heartbeat mechanism will form a correlation graph. For abnormal behaviors, complex network methods are used to conduct community mining to detect botnets. The methods used can be typically divided into two categories: graph methods and community mining algorithms.

(1) Graph Method. There are two main ideas for graph-based methods. One is for the behavior of executable files, such as control flow graphs, call graphs, and code graphs, to model graphs. The other is based on the behavior of nodes in network traffic, e.g., the IP-domain name mapping relationship is modeled on the graph, and then classified and detected on this basis.

The literature [83] proposed a new high-order subgraph feature based on PSI (printable characters) extracted from malicious code to detect large-scale botnets. These features had precise behavior descriptions and less space requirements. Aiming to Large Scale Spamming, BotGraph [84] revealed the correlation between botnet activities by constructing a large user graph, including two components: detector registration and behavior connection. The first component ensures that the total number of bots was limited, and the second component was based on constructing a user-user random undirected graph to detect invisible robot users, and then it detected bots through abnormal behavior [85]. Based on the topological features of the nodes in the graph, a novel botnet detection method was proposed, which extracts in-degree, out-degree, weight, degree weight, clustering coefficient, internodes and feature vector centrality, and based on these features, a self-organizing map clustering method SOM was used to establish the clustering of nodes in the network. This method can isolate bots in small clusters. In literature [86], for high-speed networks, it correlates NetFlow-related data and uses host-dependent models for advanced data mining, extends the popular link analysis algorithm PageRank [87] for cluster processing, and uses P2P communication infrastructure to effectively detect invisible botnets without a significant amount of traffic. Aiming at the anonymity of botnets, based on the DNS query response, a mapping relationship between its domain name and IP was extracted through the DNSmap tool to construct a DNS association map in [88] named XIONG. Moreover, the authors analyzed the structural characteristics, FQDN (full domain name was called fully qualified domain name) node and IP node characteristics, and connection edge characteristics of the graph and integrated the blacklist statistical characteristics to realize the multifeature analysis of the graph components and selected the light GBM algorithm to complete diagram component classification. It also proposed a prototype system for Fast-Flux and Domain-Flux botnet detection under high-speed networks. The architecture was analyzed from the vertical perspective of data flow transmission, which was divided into data access layer, data storage layer, processing unit layer, and user interface layer.

(2) Community Mining. The literature [89] considered three types of community behavior: traffic statistics characteristics, digital community characteristics, and structural community characteristics, and proposed an early method based on community behavior analysis, PeerHunter, which uses complex networks for community detection. Louvain method algorithm could detect botnets communicating through the P2P structure. In [90], advanced features were extracted from network traffic to detect P2P botnets in real time. By jointly considering flow-level traffic statistics and network connection patterns, a dynamic group behavior analysis (DGBA) was applied to distinguish P2P bot-infected hosts from legitimate P2P hosts, and a new dynamic group behavior analysis was conducted to extract the collective and dynamic join patterns for each group. Wang and Paschalidis [91] proposed a new two-stage method for detecting the existence of botnets and identifying damaged nodes. The first stage detected anomalies by using large deviations in the empirical distribution. Two methods for creating empirical distributions were proposed: the flow-based method to estimate the histogram of the quantized flow and the graph-based method to estimate the degree distribution of the node interaction graph, including the ER graph and the unscaled graph. The second stage used the idea of social network community detection in the network to detect robots. This graph captured the relationship between the interactions between nodes over time and conducted community detection by maximizing the modularity metric in this graph.

Table 2: Summary of typical botnet detection techniques based on deep learning.

Papers	Mechanism	Algorithm/ model	Dataset	Advantage	Drawback
BotCloud [58]	Basic features were extracted from the network stream and mapped onto grayscale images, CNN was adopted for feature learning, and SVM was used for classification detection	LeNet-5	CTU-13 [76]	(i) Suitable for BotCloud environments (ii) Automated extraction of network traffic characteristics (iii) Detected unknown botnets	(i) Attackers can use counter machine learning ideas to escape
[59]	Channel information (such as power consumption) was collected on the side of the Internet of Things, and CNN was used to perceive subtle differences in power consumption data	8-layers CNN	Collected data itself	(i) Lightweight detection systems (ii) Suitable for resource-constrained IoT devices	(i) The detection object is relatively single (ii) Difficult to collect side-channel information
[63]	The four attack vectors of Mirai were sent into RNN as feature vectors and detected from both positive and negative directions	BiLSTM-RNN	Collected data itself	(i) Attack vector text feature recognition (ii) High detection accuracy	(i) The two-way approach increases the time overhead
Bot Catcher [66]	The network flow was converted into NetFlow, and data flow 1024 B (32×32) before each data flow was intercepted and converted into grayscale images. CNN is to learn the spatial features from the spatial dimension through network traffic. RNN is used to learn the time characteristics of the data stream from the time dimension	Le Net-5 BiLSTM	CTU-13	(i) General-purpose flow detection (ii) Automated extraction of network traffic characteristics (iii) Detect unknown botnets	(i) The model is complex (ii) For massive data, the training speed is slightly slower
[69]	For the DGA, based on word-hashing technology, domain names were represented by binary syntax strings, and domain names were mapped into higher dimensional vector space using the word bag model. Then, 5-layer DNN was used to classify domain names	Word-hashing DNN	Alexa [77]	(i) Hidden patterns and features at different levels of abstraction can be discovered from training data (ii) High detection accuracy	(i) It takes a lot of training data (ii) The dataset is relatively unitary
[71]	For DGA, a deep learning framework (TS-ASRCAPS) based on double-stream networks was constructed, which used multimodal information to reflect the characteristics of DGAS, and an attention sliced recurrent neural network (ATTSRNN) was proposed to automatically mine the underlying semantics. Capsule network (CapsNet) with dynamic routing is used to model high-level visual information	ATTSRNN CapsNet	Alexa; OSINT [78]; Lab360 [79]; Andrey Abakumov [80]	(i) Automatically learn multimodal representations from the data, bypassing the human effort of feature engineering.	(i) The training speed is slower
Fast-Flux hunter [72]	Fast-Flux botnet domain was distinguished from the legal domain in an online mode based on new rules, characteristics, or classes for enhanced learning using the EFUNN algorithm	EFUNN	ISOT [81]	(i) A mix between supervised and unsupervised knowledge-based online learning systems (ii) Real-time detection can deal with zero-day attacks	(i) The dataset is too single (ii) Difficult to detect deep latency botnets

Table 2: Continued.

Papers	Mechanism	Algorithm/ model	Dataset	Advantage	Drawback
PRCL [74]	Dynamic optimization and deployment of detectors through reinforcement learning	POMDP	ISCX [82]	(i) On-line real-time detection (ii) High precision (iii) Fast processing speed	(i) This rule is used to detect botnets with classes less than 5000, not suitable for large botnets

The summary of typical botnet detection techniques based on complex networks is shown in Table 3.

5.3.3. Swarm Intelligence. The swarm intelligence optimization algorithm mainly simulates the group behavior of insects, herds, birds, or fish, which search for food in a cooperative manner. Each member of the group constantly changes the search direction through learning experience. The main idea of this type of botnet detection method is to use heuristic biological behavior to search and find abnormal points, perform feature extraction, and then combine with classifiers for detection.

(1) PSO. The literature [95] proposed a botnet detection method (BD-PSO-V), which was a hybrid particle swarm algorithm and voting system. The PSO algorithm was used for feature selection of network stream data. The feature was considered as particles, and the birds found the best particles. The voting system, including a deep neural network algorithm, support vector machine (SVM), and decision tree C4.5, based on the maximum number of votes, was utilized to identify botnets and classify samples. Six well-known adversarial attacks, including Fast Gradient Sign Method (FGSM), were evaluated on the ISOT and Bot-IoT datasets. The literature [96] proposed a detection model based on multiobjective particle swarm optimization (MOPSO) to identify malicious behaviors in bulk network traffic. In [97], a smart adaptive particle swarm optimization support vector machine (SAPSO-SVM) algorithm was proposed for Android botnet detection application. The algorithm used the changes in each stage of the execution process of the personal best and the global best to specify a new evolution factor value and then eliminated the interference of the inertial weight interval.

(2) GWO. The literature [98] proposed a new unsupervised evolutionary IoT botnet detection method. By using the latest Grey Wolf Optimization (GWO) swarm intelligence algorithm to optimize the hyperparameters of one-class support vector machine (OCSVM), it detected botnet attacks launched from compromised IoT devices.

The summary of typical botnet detection techniques based on swarm intelligence is shown in Table 4.

The summary of typical botnet detection techniques based on statistical analysis is shown in Table 5.

5.3.4. Statistical Analysis. The statistical method is mainly based on the data modeling of its statistical attributes to find outliers and estimate whether the test sample is a bot. The literature [102] proposed a spatial snapshot rapid flux detection system (SSFD), which relied on spatial distribution estimation and spatial service relationship evaluation. The space of distinction and information entropy were combined to measure the equivalent distribution of the nodes in each time zone. The benign areas tended to be distributed in the same time zone, whereas the fast-flux nodes were widely distributed in multiple time zones. [103] Aiming at the Internet of Things botnet DGA, a lightweight system was proposed to detect IoT-based botnets through the flow of the rapid recognition algorithm generation domain (AGDS). Threshold random walk (TRW) was used to quickly classify NXDOMAIN (a large set of random nonexistent domain names) query flows to create opportunities to interrupt C&C connections.

Botnets can hide periodic behavior characteristics by changing the communication interval. When the interval is too large, a time-series analysis cannot detect a periodic communication behavior. Based on the periodic communication detection method along with sequential hypothesis testing, Wang et al. [104] proposed a botnet periodic communication behavior detection algorithm and introduced a fast quantum search algorithm called the Grover quantum state to better realize parallel processing and improve the algorithm speed. This method can complete botnet detection with less query time and improved detection speed.

5.3.5. Distributed Approach. To increase the detection accuracy and improve the flexibility of the detection system, some literature studies have designed the distributed detector to collect massive and multidimensional data for detection.

(1) MTD. The literature [105] proposed a new type of botnet defense mechanism based on a combination of honeypot and network-based strategies, MTD (moving target defense), and reinforcement learning technology. The MTD method is used to periodically change the position of the detector, constantly reshaping the attack surface of the system and increasing the complexity and cost of the attacker. Using reinforcement learning to optimize and dynamically deploy detectors in an iterative manner, the agent learns a strategy to maximize the detection and removal of bots over time.

(2) SDN. SDN technology realizes the separation of control plane and data planes, and the visibility and programmability provided are typically used to implement

Table 3: Summary of typical botnet detection techniques based on complex networks.

Papers	Mechanism	Algorithm/model	Dataset	Advantage	Drawback
PSI [83]	High-order subgraph features based on PSI were extracted from malicious code, combined the classifier to detect the Internet of Things botnet	SVM, RF, Bagging, DT, kNN	IoTPOT [92]; VirusShare [93]	(i) Extracted the PSI subgraph from malicious code (ii) The detection accuracy is greater than 97%	(i) It is difficult to capture malicious samples
BotGraph [84]	For Large Scale Spamming botnet, a large user graph was constructed to reveal the correlation between botnet activities and bots were detected through abnormal behavior	MapReduce; Exponential Weighted Moving Average (EWMA)	Hotmail registration log	(i) Proposed a novel graph-based method to detect new Web account abuse attacks (ii) A new distributed programming model, MapReduce, for building and analyzing large images	(i) The topology of the graph is large (ii) The message cannot be detected before transmission
XIONG [88]	In view of the anonymity of the zombie, according to the DNS query response, the mapping relationship between the domain name and IP was extracted by DNSmap tool, and the DNS correlation map was constructed	DNSmap	CTU	(i) DNS traffic is small, detect early	(i) For P2P botnets that do not conduct DNS query, the effect is not good
	Light GBM algorithm is used to complete the classification of graph components	Light GBM	ISCX-bot	(ii) High-speed flow detection	
PeerHunter [89]	First, P2P traffic was filtered, MCG was constructed, and the community was mined through the community mining algorithm. Then, detection is carried out according to the statistical characteristics of the community	MapReduce	Collected P2P data itself	(i) An early approach to community behavior analysis	(i) Need to manually adjust the community statistics
		Louvain		(ii) Good elasticity	(ii) Does not work well against deep latency botnets
[91]	According to the traffic, histogram and graph method were used to extract the key abnormal nodes, and then social association graph (SCG) was constructed, and community detection idea was used to detect robots	SCG	CTU-13	(i) Key nodes were identified and community similarity analysis is carried out	(i) Not suitable for small-scale graph
		Louvain	CAIDA [94] xml	(ii) The modularity measure function was optimized	

Table 4: Summary of typical botnet detection techniques based on swarm intelligence.

Papers	Mechanism	Algorithm/ model	Dataset	Advantage	Drawback
[95]	A hybrid particle swarm optimization (PSO) and voting system botnet detection method (BD-PSO-V) was proposed PSO algorithm was used for feature selection of network stream data. The voting system was used to identify botnets and classify samples	PSO	ISOT	(i) Adaptive flow feature selection method	(i) High time complexity
		DDN SVM C4.5	Bot-IoT [99]	(ii) Detect during the attack phase	
[97]	For Android botnet, a smart adaptive particle swarm optimization support vector machine (SAPSO-SVM) algorithm was used for detection	SAPSO	28 Standard Android Botnet Dataset (28-SABD) [100]	(i) Automatically extract Android botnet features	(i) High time complexity
		SVM		(ii) High detection accuracy	
[98]	GWO swarm intelligence algorithm was used to optimize the hyperparameters of OCSVM to detect botnet attacks from damaged IoT devices	GWO	N-BaIoT [101]	(i) Deal with heterogeneous IoT devices	(i) IoT devices are increasing rapidly
		OCSVM		(ii) A new unsupervised evolutionary Internet of Things botnet detection method	(ii) Cannot detect unknown botnets

Table 5: Summary of typical botnet detection techniques based on statistical analysis.

Papers	Mechanism	Algorithm/ Model	Dataset	Advantage	Drawback
[102]	For the fast-flux network, the equivalent distribution of nodes in each time region was measured by the combination of spatial distribution estimation and spatial service relationship evaluation	The information entropy	Collected data itself	(i) Simple (ii) High efficiency	(i) The accuracy is not particularly high
[103]	For Internet of Things botnet DGA, a rapid classification of NXDOMAIN (a large set of random nonexistent domain names) query stream was created by using a threshold random walk (TRW) to create an opportunity to break a C&C connection	Threshold random walk	Collected data itself	(i) Not relying on expert knowledge (ii) Lightweight detection is faster (iii) Can detect unknown botnets	(i) Statistics cannot be applied to heterogeneous data, only to quantitative data
[104]	According to the periodic communication behavior of botnet, based on sequential hypothesis periodic communication detection, a fast quantum search algorithm Grover quantum state was introduced to better realize parallel processing	Grover	Mixed 10 datasets	(i) Random periodic behavior can be detected (ii) Speed up the algorithm	(i) Difficult to resist traffic-based adversarial learning

various security detection and attack mitigation schemes. The basic idea of the SDN-based botnet detection mechanism is based on Open vSwitch, a virtual switch that implements the OpenFlow protocol, combined with classifiers to detect bots and identify malicious traffic. OFX [106] proposed by Sonchack et al. can deploy security functions in the existing OpenFlow infrastructure, allowing control applications to dynamically load security modules directly into unmodified SDN-compatible switches. Zha et al. [107] proposed an SDN-based scalable, accurate, and online data center bot detection framework BotSifter, which distributed detection tasks across the edge of the network in Open vSwitch, and the use of centralized learning (DNN) and distributed detection enhances the robustness of detection. The literature [108] proposed a lightweight real-time botnet detection scheme BotGuard under SDN, using the idea of graph matching algorithm, and proposed a convex lens imaging model graph to describe the topology feature of the botnet. It allowed the SDN controller to independently locate the attack location while reducing the network load. The Mininet platform was used for simulation evaluation.

(3) Blockchain. The basic idea of using blockchain technology for botnet detection is to the use of smart contracts, digital signatures, incentive mechanisms, and other technologies, based on proxy or collaborative detection, to achieve trust information exchange or voting among different detectors. In [109], AutoBotCatcher used BFT (Byzantine Fault Tolerant) to perform dynamic and collaborative botnet detection on large networks and used the community detection algorithm Louvain method to detect communities. The literature [110] proposed a blockchain trust model (BTM) for malicious node detection in wireless sensor networks, which used blockchain smart contracts and WSN quadrilateral measurement and positioning methods to achieve malicious node detection in the 3D space, with good traceability. Based on a consensus mechanism blockchain, Spathoulas et al. [111] used lightweight agents installed at multiple IoT locations to collaboratively detect DDoS attacks carried out by a botnet of IoT devices. The literature [112] proposed an incentive platform SmartRetro driven by blockchain smart contracts and PoW consensus schemes, which could incentivize and attract more distributed detectors to participate in traceable vulnerability detection and contribute their detection results.

The summary of typical botnet detection techniques based on distributed approach is shown in Table 6.

5.3.6. Combination Method. The evolution of botnets presents characteristics such as diversified platforms, concealed communications, and intelligent control. A single abnormal behavior detection method cannot meet the actual requirements. Multidimensional, multiagent, and multitechnology combined detection methods have therefore emerged.

Multidimensional refers to the combination of multiple detection objects, mainly referring to the combination of network traffic and signature detection. The literature [113] proposed a hybrid botnet detection method HANABot based on a host-side and network analysis; this is a general technology that can detect new botnets in the early stage. The system contains three components: network analysis component, host analysis component, and a test report. The document [114] proposed an effective two-stage traffic classification method based on a non-P2P traffic filtering mechanism and session feature-based machine learning technology to detect P2P botnet traffic. In the first stage, non-p2p packets are filtered, and network traffic was reduced through well-known ports, DNS queries, and flow counting. In the second stage, the session features were extracted based on the data stream characteristics and stream similarity, and the P2P botnet was successfully detected by the machine learning classifier. The literature [115] proposed a signature generation method based on the similarity of HTTP botnet header information, which could

Table 6: Summary of typical botnet detection techniques based on distributed approach.

Papers	Mechanism	Algorithm/model	Dataset	Advantage	Drawback
[105]	A new botnet defense mechanism was proposed based on honeypot and network-based strategy, MTD, and reinforcement learning technology	MTD; reinforcement learning	Collected data itself	(i) Detect covert botnets (ii) Good elasticity	(i) High time complexity
BotSifter [107]	A BOT detection framework based on SDN, BotSifter, was proposed to distribute the detection tasks across the network edge in Open vSwitch, using centralized learning (deep neural network) and distributed detection	SDN; deep neural network	Collected data itself	(i) Distributed detection (ii) Enhanced robustness	(i) SDN deployment issues
[111]	Blockchain-based consensus mechanism, using lightweight agents installed at multiple IoT locations to collaboratively detect DDoS attacks by IoT device botnets	PoW	Collected data itself	(i) Lightweight agent (ii) Distributed detection	(i) The detection object is relatively single (ii) The cost of blockchain technology

automatically generate high-quality network signatures. This method combined the advantages of network traffic and data packet detection, and TCP flow was used as the object to extract the size statistical characteristics of the HTTP first request packet and the first response packet (referred to as "a question, one answer packet") and combine the HTTP header field content. Statistical analysis could detect "silent" state bots. The literature [116] proposed a multistage detection method for domain fluxing, fast-flux service network (FFSN), and domain generation algorithm (DGA). The first stage used NX domain and server failure errors to detect DNS tunnel C&C server calls. In the second stage, a signature matching technology was used to detect the DNS tunnel SSH handshake between BOTS and C&C server.

Multiagent (Agent) refers to the combination of multiple agents [117]. A multiagent Robot Detection System (MABDS) [118] was a hybrid technology that associates an event log analyzer with a host-based intrusion detection system (HIDS). It used multiagent technology to combine management agent, user agent, honeypot agent, system analysis, and knowledge database.

Multitechnology refers to the combination of multiple technologies or algorithms. The literature [119] used the characteristics of a graph and a neural network for detection. They generated network communication graphs at regular intervals by modeling the graph features over time, extracted the graph-based statistics and central features, assembled a time series of the features for each host (identified by IP address), and trained the time-series classification model. Ten graph features were extracted for each node: out-degree, in-degree, adjacency, neighbor, PageRank centrality, intermediate centrality, feature vector centrality, authority, and hub centrality, using the graph tool library [120] local clustering coefficient. Using the time-series data in the network is suitable for a real-time detection.

The summary of typical botnet detection techniques based on combination method is shown in Table 7.

We mainly focus on the comparison of botnet detection techniques based on abnormal behavior. The basic ideas, advantages, and disadvantages of various methods are summarized in Table 8.

6. Evaluation

The classification of botnet detection systems has more dimensions. This section draws on AHP, from the four dimensions of $D_{\text{Service}}(t)$, $D_{\text{Intelligent}}(t)$, $D_{\text{Collaboration}}(t)$, and $D_{\text{Assistant}}(t)$, a general botnet detection system performance evaluation system CBDES (common bot detection evaluation System) is designed. By constructing the judgment matrix and checking the consistency, the performance of the subsystems is independent of each other, and the weight of each index is calculated. Combining expert scores and objective weights, a new global capability metric € is defined, and eight typical detection methods are quantified and evaluated. Finally, a visual representation is given using spider graphs.

6.1. Index System. The performance index system of the botnet detection system is mainly divided into four dimensions: $D_{\text{Service}}(t)$, $D_{\text{Intelligent}}(t)$, $D_{\text{Collaboration}}(t)$, and $D_{\text{Assistant}}(t)$, as shown in Figure 3.

6.2. Evaluation Index System

(1) $D_{\text{Service}}(t)$ refers to the basic performance of the botnet detection system. The indicators are divided into three subindices: accuracy, scenes, and stage.

The accuracy $F_{\text{Service(ac)}}$ refers to the accuracy of botnet detection. The scenes $F_{\text{service(sc)}}$ refers to which scenes the method is suitable for. Stage $F_{\text{service(st)}}$ refers to which stage of the botnet's life cycle is detected.

In the $D_{\text{Service}}(t)$ dimension, the weight is used to give the importance of each indicator, the quantitative formula of this dimension is obtained, and the result is normalized to [0,1]:

Table 7: Summary of typical botnet detection techniques based on combination method.

Papers	Mechanism	Algorithm/ Model	Dataset	Advantage	Drawback
[113]	HANABot, a hybrid botnet detection method based on host and network analysis, was proposed, which could detect new botnets in the early stage	NB,DT	Collected data itself	(i) Detection can be performed at an early stage (ii) Detection of multidimensional data	(i) Dynamically updating rules, configuration files, or signatures is still difficult
[118]	MABDS associated the event log analyzer with a host-based intrusion detection system (HIDS)	A variety of techniques	Collected data itself	(i) Used multiagent technology to combine administrative agent, user agent, honeypot agent, system analysis, and knowledge database	(i) Lack of proper composition can result in high computing costs (ii) It is a complicated method
[119]	Based on the traffic, the network communication graph is generated at regular intervals by modeling graphic features over time, and the statistics and central features based on the graph are extracted and classified	LSTM	CTU-13	(i) The characteristics of graph and neural network are used to detect the image (ii) Inclusion evolution feature	(i) The dataset is relatively unitary (ii) Not universal

Table 8: Comparison of botnet detection technology methods based on abnormal behavior.

	The basic idea	Advantage	Disadvantage
Deep learning	(i) Using neural network to extract network traffic features based on temporal and spatial similarity. Map the network traffic into a grayscale image or feature vector and send it to the neural network model, extract distinguishable features and patterns from the two dimensions of time, space, or time and space, and automatically learn network traffic characteristics	(i) It does not rely on any prior knowledge about the protocol and topology, does not need to manually select features, and automate feature extraction	(i) Attackers can use anti-machine learning ideas to escape
	(ii) Using reinforcement learning for new feature extraction, or detector distributed strategy placement	(ii) It has certain detection capabilities against unknown botnets and encryption protocol botnets (iii) High accuracy (iv) Improve the detection and prediction of unknown "zero-day" online Fast-Flux botnets	(ii) For massive data, the training speed is slightly slower (iii) Difficult to detect deep latent botnets
Complex network	(i) Graph-based methods are mainly aimed at the behavior of executable files, such as control flow graphs, call graphs, and code graphs, to model graphs; or to graph based on node behaviors in network traffic, such as IP-domain mapping relationships modeling, classification and detection are carried out on this basis	(i) Better display of behavioral associations, combined with visualization methods to help researchers detect	(i) Pre-established rules are required to detect botnets from the graph
	(ii) Based on the relatively frequent communication activities of zombies, a correlation graph will be formed, and based on the analysis of abnormal community behaviors, the complex network method is used to mine to detect botnets	(ii) Effectively detect invisible botnets without a lot of traffic	(ii) The accuracy of determining the behavioral association threshold is unstable—if the dataset is large, the computational cost of the detection method is usually high
Swarm intelligence	(i) Using heuristic biological behavior to search, feature extraction, and then combined with classifiers for detection	(i) It can be extracted from multiple aspects, without prior knowledge of system behavior, with high accuracy (ii) Can detect unknown botnets	(i) The disadvantage is the high time complexity. The heuristic rules require a lot of time to check the data against all the rules.

Table 8: Continued.

	The basic idea	Advantage	Disadvantage
Statistical analysis	(i) Modeling based on the statistical properties of zombie behaviors and estimating samples	(i) Statistical analysis can be quantified and analyzed relatively quickly	(i) Botnets change quickly and have complex features, which adds difficulty to statistical analysis (ii) Statistics cannot be applied to heterogeneous data, only quantitative data
Distributed detection	(i) Design and deploy multiple detectors to improve the flexibility of the detection system and collect massive and multidimensional data for detection	(i) Improve accuracy (ii) Improve the flexibility of the detection system	(i) It is difficult to choose a comprehensive deployment strategy (ii) Time-consuming
Combination method	(i) Multidimensional (ii) Multiagent (iii) Multiple technologies	(i) Helps to detect under high-speed network environment (ii) Helps to detect at an early stage (iii) Has good flexibility (iv) Can detect unknown attacks.	(i) Lack of proper combination may lead to high computational cost

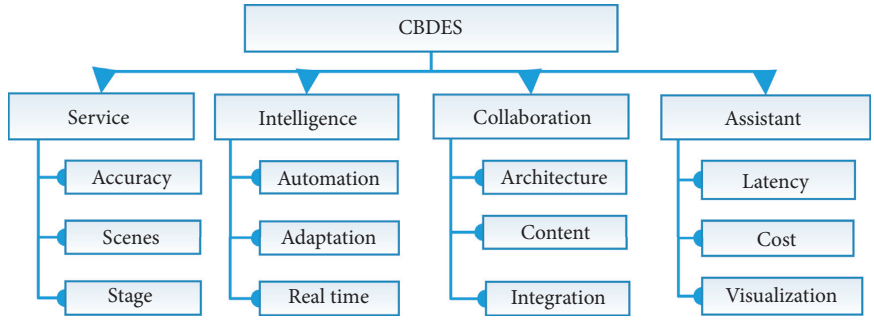

Figure 3: The performance index system of the botnet detection system.

$$D_{\text{Service}}(t) = w_{\text{ac}*}F_{\text{Servicey(ac)}}(t) + w_{\text{sc}*}F_{\text{Service(sc)}}(t) + w_{\text{st}*}F_{\text{Service(st)}}(t). \quad (1)$$

(2) $D_{\text{Intelligent}}(t)$ refers to the degree of automation of the detection system, which is divided into three subindexes: automation, adaptability, and real time.

Automated $F_{\text{Intelligent(au)}}$ refers to the degree of automation of feature extraction in the detection process. Adaptive $F_{\text{Intelligent(ad)}}$ refers to whether the detection model can detect unknown types of botnets. Real-time $F_{\text{Intelligent(re)}}$ refers to whether the detection system can be performed in real time.

In the dimension, the weight is used to assign the importance of each indicator to get the quantitative formula of this dimension, and the result is normalized to [0,1]:

$$D_{\text{Intelligent}}(t) = w_{\text{au}*}F_{\text{Intelligent(au)}}(t) + w_{\text{ad}*}F_{\text{Intelligent(ad)}}(t) + w_{\text{re}*}F_{\text{Intelligent(re)}}(t), \quad (2)$$

(3) $D_{\text{Collaboration}}(t)$ refers to the system's synergy and scalability and is divided into subindices such as architecture, content, and integration.

$F_{\text{Collaboration(ar)}}$ refers to the organizational structure of the detection system, which is divided into centralized and distributed. Content $F_{\text{Collaboration(co)}}$ refers to the category of detection content, which is divided into single type, such as host log or network traffic. Diversity refers to the detection of host and network combination, or code and traffic combined with multiple data. Integration $F_{\text{Collaboration(in)}}$ refers to the use of multiple types of detection methods.

In the $D_{\text{Collaboration}}(t)$ dimension, the weight is used to give the importance of each indicator, the quantitative formula of this dimension is obtained, and the result is normalized to [0,1]:

$$D_{\text{Collaboration}}(t) = w_{\text{ar}*}F_{\text{Collaboration(ar)}}(t) + w_{\text{co}*}F_{\text{Collaboration(co)}}(t) + w_{\text{in}*}F_{\text{Collaboration(in)}}(t). \quad (3)$$

(4) $D_{\text{Assistant}}(t)$ refers to some other indicators, mainly latent, cost, and visualization.

Latent $F_{\text{Assistant(la)}}$ refers to whether it can detect deep latent botnets. Cost $F_{\text{Assistant(cos)}}$ refers to the consumption cost of the detector, such as GPU and bandwidth consumption. $F_{\text{Assistant(vi)}}$ refers to the visualization of data information or botnet detection through visualization methods.

In the $D_{\text{Assistant}}(t)$ dimension, the weight is used to assign the importance of each indicator to get the quantitative formula of this dimension, and the result is normalized to [0,1]:

$$D_{\text{Assistant}}(t) = w_{la*}F_{\text{Assistant(la)}}(t) + w_{\cos *}F_{\text{Assistant(cos)}}(t) + w_{vi*}F_{\text{Assistant(vi)}}(t). \tag{4}$$

According to the calculated value of the four dimensions, it can be abstracted as a polygonal area. This paper uses Gauss's area formula to define a global measure.

A new global competency metric € is defined with expert ratings and objective weights as follows:

$$€ = D_{\text{Service}}(t) + D_{\text{Intelligent}}(t) + D_{\text{Collaboration}}(t) + D_{\text{Assistant}}(t). \tag{5}$$

6.3. Quantitative Assessment

(1) AHP (analytic hierarchy process) is used to determine various weight indicators

Step 1: construct a judgment matrix. According to the AHP hierarchical structure, the judgment matrix of the criterion layer to the target layer is constructed from top to bottom. Generally, the factors of the lower layer are used to evaluate the factors of the upper layer. The judgment matrix is composed of the results of the comparison of the factors of the lower layer. As shown in Table 9, the scale is 1–9.

First construct the judgment matrix of criterion layer and index layer:

$$w = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 1/2 & 1 & 2 & 3 \\ 1/3 & 1/2 & 1 & 2 \\ 1/5 & 1/3 & 1/2 & 1 \end{bmatrix},$$

$$w_{\text{se}} = \begin{bmatrix} 1 & 3 & 4 \\ 1/3 & 1 & 3 \\ 1/4 & 1/3 & 1 \end{bmatrix} w_{\text{in}} = \begin{bmatrix} 1 & 2 & 2 \\ 1/2 & 1 & 2 \\ 1/2 & 1/2 & 1 \end{bmatrix} w_{\text{co}} = \begin{bmatrix} 1 & 2 & 2 \\ 1/2 & 1 & 2 \\ 1/2 & 1/2 & 1 \end{bmatrix},$$

$$w_{\text{as}} = \begin{bmatrix} 1 & 2 & 2 \\ 1/2 & 1 & 3 \\ 1/2 & 1/3 & 1 \end{bmatrix}. \tag{6}$$

Taking the criterion layer as an example, the calculation method of weight value is introduced in detail.

Using the L1 paradigm, the W elements are normalized by column to get

$$\tilde{w} = \begin{bmatrix} 0.492 & 0.522 & 0.462 & 0.455 \\ 0.246 & 0.261 & 0.308 & 0.273 \\ 0.164 & 0.130 & 0.154 & 0.182 \\ 0.098 & 0.087 & 0.077 & 0.091 \end{bmatrix}. \tag{7}$$

Step 2: calculate the weight of each performance index.

This paper uses the maximum eigenvalue max corresponding to the components of the standard eigenvector as the weight of each factor.

$\lambda_{\max} = 4.245$, The feature vector is $[0.482 \;\; 0.272 \;\; 0.157 \;\; 0.088]^T$.

Step 3: the consistency correction of the judgment matrix.

In addition to human factors, the consistency of the judgment matrix is different in the acceptance range of the consistency of the judgment matrix of different orders. By using the consistency index, RI revises CI to achieve its goal. The *RI* value is shown in Table 10.

$$\text{CI} = \frac{\lambda_{\max} - n}{n-1} = \frac{4.245 - 4}{3} = 0.082, \quad \text{thus RI} = 0.09,$$

$$\text{CR} = \frac{\text{CI}}{\text{RI}} \times 100\% = 0.091 < 0.1, \tag{8}$$

it passed the inspection.

Therefore, the weight vector of the criterion layer is $[0.482 \;\; 0.272 \;\; 0.157 \;\; 0.088]^T$.

Step 4: use the same method to calculate the weight of the indicator layer, and then multiply the weight of the above layer to get the value of all weights, as shown in Table 11.

(2) Description of various indicators of botnet detection system is shown in Table 12.

(3) Evaluation of typical botnet detection system

According to the botnet detection methods introduced in Section 3, eight typical detection methods are evaluated.

Step 1: expert scores are as shown in Table 13, where EV represents evaluation vector.
Step 2: calculate the four-dimensional indicators according to the weight values obtained by the analytic hierarchy process and the calculation of the global metrics. Table 14 is obtained and sorted. According to the quantitative evaluation proposed in this article, the top four methods with good detection performance are PRCL, BotSifter, PeerHunter, and Bot Catcher.
Step 3: use spider diagrams for visual representation of the top four as shown in Figure 4.

7. Challenges and Prospects

7.1. Challenges. In the process of attacking security organizations and overcoming government supervision, botnets are constantly evolving. To solve problems, such as concealment, survivability, and survivability, new terminal botnets, such as IoT and smart mobile devices, have also become the main source of various types of Internet security threats. Nevertheless, the industry has a more in-depth understanding of the working mechanism and behavior characteristics of botnets, and a variety of botnet detection methods have been proposed. This article summarizes the challenges faced by detection methods:

(1) Multisource information collection and fusion: because of the concealment and cross-platform nature of botnets, its traces are often hidden in various information scattered at different dimensions, such as personal hosts, regional networks, and backbone networks, and stored in different formats. Multisource information contains various redundant information. Data collection must have the characteristics of collaboration, distribution, and intelligence. Any combined method should be highly accurate and have a low complexity, provide unified data representation and storage, and then perform data processing. It should also dynamically adjust the collection based on the actual scene strategy.

(2) Deep latent command and control channel: botnet detection is typically in the communication and attack stage of its life cycle. In the communication stage, the focus is on traffic data. The centralized structure of botnets shows strong similarity and correlation characteristics, and the detection effect is evident. However, for third-party channels, such as P2P, cloud platform, and blockchain technology, there is a lack of effective detection methods. Detecting deep latent botnets is challenging in the early stages, such as the spread and infection stages.

(3) High-speed network real-time detection: the backbone network has the characteristics of high bandwidth, large traffic, and limited storage. These factors have led to the slow development of real-time detection technology. Lightweight real-time detection is an important content of future research.

(4) Detection system structure coordination: the existing botnet detection system architecture has some problems. First, the centralized structure is unsuitable for large-scale network environments; the second is that the feature extraction method is not flexible, the structure is single, and multiple methods cannot be integrated; the third is the lack of coordination in the architecture function, and although some systems implement distributed detection, they lack effective information sharing and cooperation, and the coordination method is single, and they cannot respond quickly to botnet activities. The detection system framework must meet the requirements of distributed, scalable, and extended models and should realize the coordination between the detection system and other security systems.

7.2. Prospects. As an evolution of conventional malicious code, botnets provide controllers with a flexible and efficient command control mechanism, which is an ideal platform for DDoS, spam, information theft, click fraud, and malware distribution. With the convergence of the network era with the advent, botnets have seen changes in terms of infection targets, management and control technologies, and malicious behaviors, which pose a greater threat to future Internet security. Future research directions and technical difficulties in the field of botnet detection include the following:

(1) Botnet multidimensional data representation: based on the knowledge graph technology, for DNS traffic, the entity-relationship and entity-attribute modeling embedding vectors are simultaneously performed, and the collected data are represented in multiple dimensions.

(2) A method based on the combination of code and traffic analysis: this is conducive to the detection of botnets in all directions, improves the detection accuracy, and is suitable for the actual environment.

(3) Lightweight deep learning model: existing traffic feature extraction methods that combine the spatiotemporal characteristics are commonly used in CNN + RNN. The model is more complex, and the processing speed is not high [104]. A fast quantum search algorithm called Grover and a cutting-edge lightweight deep neural network can help improve the feature extraction speed.

(4) Efficient community mining algorithm and visual detection technology: the visualization of botnet behavior and efficient data mining algorithms in the field of social networks help botnet group detection.

(5) Lightweight real-time detection: the DNS traffic in the network traffic is relatively less. For zombie DNS

Table 9: Scale.

Score	Meaning
1	Both performance indicators are equally important
3	The former indicator is slightly more important than the latter
5	The former indicator is more important than the latter
7	The former indicator is much more important than the latter
9	The former indicator is extremely more important than the latter
2\4\6\8	The score set in the intermediate states of the two judgments
	If the two performance indicators are reversed, the score is reciprocal

Table 10: Average random consistency index (RI).

n	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Table 11: Weight value.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$W_{Service}$	0.482	w_{ac}	0.293	w_{se}	0.131	w_{st}	0.058
$W_{Intelligent}$	0.272	w_{au}	0.133	w_{ad}	0.085	w_{re}	0.054
$W_{Collaboration}$	0.157	w_{ar}	0.077	w_{co}	0.049	w_{ln}	0.031
$W_{Assistant}$	0.088	w_{la}	0.042	w_{cos}	0.031	w_{vi}	0.015

Table 12: Indicator description.

Evaluation group	Attribute value	Index	Description	Score
Service	Accuracy	Low	Detection accuracy of botnet is below 70%	0.7
		Middle	Detection accuracy is between 70% and 80%	0.8
		High	Detection accuracy is above 90%	0.9
		Very high	Detection accuracy is above 95%	0.95
	Scenes	General	General botnet detection	0.9
		Special	Special botnet detection	0.8
	Stage	Early	Refers to detection during the botnet propagation or addressing phase	0.7
		Interaction	Refers to testing at the interactive stage	0.5
Intelligent	Automation	Low	Professionals are required to manually extract features	0.5
		Middle	Partial feature automatic extraction	0.7
		High	Fully automated feature extraction	0.9
	Adaptation	Low	Cannot detect unknown botnets	0.6
		High	Can detect unknown botnets	0.9
	Real time	No	Failed to perform real-time detection	0.6
		Yes	Real-time detection possible	0.9
Collaboration	Architecture	Centralized	Centralized inspection system architecture	0.6
		Distributed	Distributed detection system architecture, better flexibility	0.9
	Content	Single	Detect single data such as host, log, or traffic information	0.7
		Multiple	Detect multiple data such as host logs, network traffic, and codes	0.9
	Integration	No	Adopt a single approach	0.7
		Yes	Adopt a variety of approaches	0.9
Assistant	Latency	No	Cannot detect deep latent BOT	0.6
		Yes	Can detect deep latent BOT	0.8
	Cost	Low	Normal power consumption	0.9
		Middle	Hardware requirements such as GPU	0.7
		High	More detectors are deployed, requiring more hardware and bandwidth resources	0.5
	Visualization	No	No visual display	0.5
		Yes	Visualize data information or botnet detection through visualization methods	0.6

Table 13: Evaluation vector score.

Evaluation group	Attribute value	Bot Catcher		PRCL		XIONG		PeerHunter		PSO		ConnSpoiler		BotSifter		HANABot	
		EV	Score	EV	Score	EV	Score	EV	Score	EV	Score	EV	Score	EV	Score	EV	Score
Service	Accuracy	Ac: VL	0.95	Ac: L	0.9	Ac: L	0.9	Ac:vL	0.95	Ac: vL	0.95	Ac: L	0.9	Ac: L	0.9	Ac:vL	0.95
	Scenes	Se: G	0.9	Se: G	0.9	Se: G	0.9	Se: G	0.9	Se: G	0.9	Se: S	0.8	Se: G	0.9	Se: G	0.9
	Lifecycle stage	St: I	0.5	St: I	0.5	St: I	0.5	St: E	0.7	St: I	0.5	St: E	0.7	St: I	0.5	St: E	0.7
Intelligent	Automation	Au: H	0.9	Au: H	0.9	Au: M	0.7	Au: M	0.7	Au: M	0.7	Au: M	0.7	Au: M	0.7	Au: M	0.7
	Adaptation	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9	Ad: Y	0.9
	Real-time	Re: N	0.6	Re: Y	0.9	Re: N	0.6	Re: N	0.6	Re: N	0.6	Re: N	0.6	Re: Y	0.9	Re: N	0.6
Collaboration	Architecture	Ar: C	0.6	Ar: C	0.6	Ar: C	0.6	Ar: C	0.6	Ar: C	0.6	Ar: C	0.6	Ar: D	0.9	Ar: C	0.6
	Content	Co: S	0.7	Co: S	0.7	Co: S	0.7	Co: M	0.9	Co: S	0.7	Co: S	0.7	Co: S	0.7	Co: M	0.9
	Deep latent latency	In: N	0.7	In: N	0.7	In: N	0.7	In: N	0.7	In: Y	0.9	In: N	0.7	In: Y	0.9	In: N	0.7
Assistant	Lurking	La: N	0.6	La: Y	0.8	La: N	0.6	La: N	0.6	La: N	0.6	La: N	0.6	La: N	0.6	La: N	0.6
	Cost	Cos: H	0.5	Cos: H	0.5	Cos: L	0.9	Cos: M	0.7	Cos: L	0.9	Cos: L	0.9	Cos: H	0.5	Cos: H	0.5
	Visualization	Vi: N	0.5	Vi: N	0.5	Vi: Y	0.6	Vi: N	0.5	Vi: N	0.5	Vi: N	0.5	Vi: N	0.5	Vi: N	0.5

Table 14: The sorting table.

Detection methods	€	$D_{Service}(t)$	$D_{Intelligent}(t)$	$D_{Collaboration}(t)$	$D_{Assistant}(t)$
PRCL	0.814	0.411	0.245	0.102	0.057
BotSifter	0.809	0.411	0.218	0.132	0.048
PeerHunter	0.805	0.437	0.202	0.112	0.054
Bot Catcher	0.804	0.425	0.229	0.102	0.048
HANABot	0.799	0.437	0.202	0.112	0.048
PSO	0.796	0.425	0.202	0.108	0.061
XIONG	0.777	0.411	0.202	0.102	0.062
ConnSpoiler	0.774	0.409	0.202	0.102	0.061

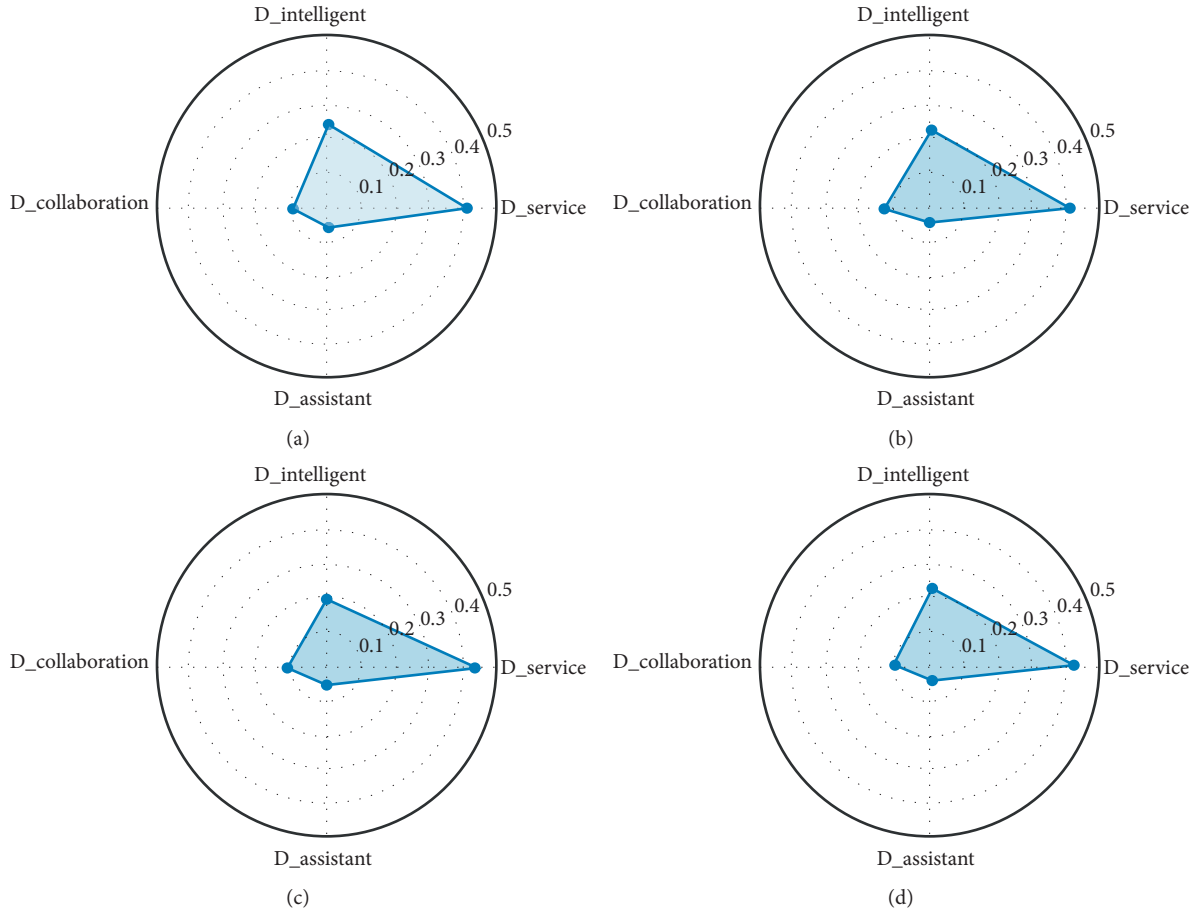

Figure 4: Spider diagrams for visual representation of the top four methods. (a) PRCL. (b) BotSifter. (c) PeerHunter. (d) Bot Catcher.

queries, community mining or deep learning methods are used for real-time detection.

(6) Distributed elastic detection framework: based on blockchain, MTD, reinforcement learning, and other technologies, the detectors can be dynamically deployed to realize information sharing between detectors and improve the flexibility of the detection system.

8. Conclusion

This survey introduces the new construction mechanism of botnet, summarizes the latest technologies in the field of botnet detection, and makes a comparative analysis of the key technologies based on anomaly. One of the contributions of this paper is to propose an evaluation system for the comprehensive evaluation of detection techniques. New botnets emerge one after another, and new technologies and their comprehensive applications will be the research focus on the field in the future. This survey is of great significance for security personnel to analyze and defend botnets, and it may help the research community to produce better tools and techniques for mitigating the threat of botnets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Key Research and Development Project (2016YFB08011601). The authors would like to acknowledge the support.

References

[1] B. Fang, X. Cui, and W. Wang, "Survey of botnets," *Journal of Computer Research and Development*, vol. 48, no. 8, pp. 1315–1331, 2011, (in Chinese).

[2] G. Vormayr, T. Zseby, and J. Fabini, "Botnet communication patterns," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2768–2796, 2017.

[3] A. Karim, R. B. Salleh, M. Shiraz et al., "Botnet detection techniques: review, future trends, and issues," *Journal of Zhejiang University Science*, vol. 15, no. 11, pp. 943–983, 2014.

[4] M. Casenove and A. Miraglia, "Botnet over tor: the illusion of hiding," in *Proceedings of the 6th international conference on cyber conflict, CyCon 2014, tallinn,Estonia*, pp. 273–282, Tallinn, Estoni, June 2014.

[5] T. Curran and D. Geist, "Using the bitcoin blockchain as a botnet resilience mechanism," 2016, https://www.os3.nl/media/2016-2017/courses/ot/dana/tom.pdf.

[6] A. Kurt, E. Erdin, M. Cebe, K. Akkaya, and A. S. Uluagac, "LNBot: a covert hybrid botnet on bitcoin lightning network for fun and profit," in *Computer Security – ESORICS 2020. ESORICS 2020*, L. Chen, N. Li, K. Liang, and S. Schneider, Eds., Springer, Berlin, Germany, 2020.

[7] P. F. Cui, Y. Qiu, and R. Sun, "Research on image recognition technology for the network content security," *Netinfo Security*, vol. 9, pp. 154–157, 2015.

[8] K. S. Q. Gul, J. Z. Yin, L. M. Pan et al., "Research on the algorithm of named entity recognition based on deep neural network," *Netinfo Security*, vol. 10, pp. 29–35, 2017.

[9] S. Dange and M. Chatterjee, "IoT botnet: the largest threat to the IoT network," in *Advances in Intelligent Systems and Computing*, L. Jain, G. Tsihrintzis, V. Balas, and D. Sharma, Eds., Springer, Berlin, Germany, 2020.

[10] I. Ali, A. I. A. Ahmed, A. Almogren et al., "Systematic literature review on IoT-based botnet attack," *IEEE Access*, vol. 8, pp. 212220–212232, 2020.

[11] M. Singh, M. Singh, and S. Kaur, "Issues and challenges in DNS based botnet detection: a survey," *Computers & Security*, vol. 86, pp. 28–52, 2019.

[12] M. Sandip Sonawane, "A survey of botnet and botnet detection methods," *Nternational Journal of Engineering Research & Technology (IJERT), ISSN*, vol. 7, no. 12, 2018.

[13] K. Alieyan, A. Almomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," *Neural Computing and Applications*, vol. 28, no. 7, pp. 1541–1558, 2017.

[14] X. Li, J. Wang, and X. Zhang, "Botnet detection technology based on DNS," *Future Internet*, vol. 9, no. 4, p. 55, 2017.

[15] K. Li, B. Fang, X. Cui, and Q. Liu, "Study of botnets trends," *Computer Research and Development*, vol. 53, no. 10, pp. 2189–2206, 2016.

[16] C. Y. Liu, C. H. Peng, and I. C. Lin, "A survey of botnet architecture and batnet detection techniques," *International Journal of Network Security*, vol. 16, no. 2, pp. 81–89, 2014.

[17] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–33, 2013.

[18] K. Li, *Research on Botnet Countermeasures Based on Behavioral Analysis*, Beijing University of Posts and Telecommunications, Beijing, China, 2017.

[19] J. Canavan, "The evolution of malicious IRC bots," in *Proceedings of the Virus Bulletin Conference*, pp. 104–114, Dublin, Ireland, October 2005.

[20] R. Fielding, J. Gettys, J. Mogul et al., "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, 1999.

[21] J. Stewart. (2004, Mar.) Phatbot Trojan Analysis. SecureWorks. http://web.archive.org/web/20080917193007/http://www.secureworks.com/research/threats/phatbot/.

[22] R. Sharpe, "Just what Is SMB," V1.2, Oct. 2002.

[23] Higgins K. J.. Smartphone weather app builds a mobile botnet [EB/OL]. (2010-03-05) [2016-06-14].-http://www.darkreading.com/risk/smartphone-weather-app-builds-a-mobile-botnet/d/d-id/1133138"http://http//www.darkreading,%20com/risk/smartphone-weather-app-builds-a-mobile-botnet/d/d-id/1133138%20.

[24] S. Dange and M. Chatterjee, "IoT botnet: the largest threat to the iot network," in *Data Communication and Networks. Advances in Intelligent Systems and Computing*Springer, Berlin, Germany, 2020.

[25] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the iot: mirai and other botnets," *CyberTrust by IEEE Computer Society*, vol. 43, 2017.

[26] S. Benzarti, B. Triki, and O. Korbaa, "A survey on attacks in internet of things," in *Proceedings of IEEE International Conference on Engineering & MIS*, Monastir, Tunisia, May 2017.

[27] Prince B.. Flashback Botnet Updated to Include Twitter as C&C [EB/OL]. http://www. securityweek.com/flashback-botnet-updated-include-twitter-cc.

[28] Y. Boshmaf, I. Muslukhov, K. Beznosov et al., "The socialbot network: when, bots socialize for fame and money," in *Proceedings of the 27th Annual Computer Security Applications*, pp. 93–102, New York, NY, USA, March 2011.

[29] Y. Boshmaf, I. Muslukhov, K. Beznosov et al., "Key challenges in. defending against malicious socialbots," in *Proceedings of the 5th USENIX Conference on. Large-Scale Exploits and Emergent Threats*, Berkeley, CA, USA, May 2012.

[30] Wyatt T.. Security Alert: Geinimi, Sophisticated New Android Trojan Found in Wild [EB/OL]. (2010-09-29) [2016-06-14].https://blog, lookout. com/blog/2010/12/29/gein.imi_trojan.

[31] S. Zhao, P. Lee, J. Lui et al., "Cloud-based push-styled mobile botnets: a case study of exploiting the cloud to device messaging service," in *Proceedings of the of the 28th Annual Computer Security Applications Conference*, New York, NY, USA, March 2012.

[32] D. Kamenski, A. Shaghaghi, M. Warren et al., "Attacking with bitcoin: Using bitcoin to build resilient botnet armies," in *Proceedings of the Conference on Complex, Intelligent, and Software Intensive System*, pp. 3–12, Springer, Lodz, Poland, July 2020.

[33] S. T. Ali, P. McCorry, P. H. Lee, and F. Hao, "Zombiecoin: powering next-generation botnets with bitcoin," in *Proceedings of the financial cryptography and data security - FC 2015 international workshops*, pp. 34–48, San Juan, Puerto Rico, 2015.

[34] G. J. Simmons, "The prisoners problem and the subliminal channel," in *Advances in Cryptology*, pp. 51–67, Springer, Berlin, Germany, 1984.

[35] G. J. Simmons, "The subliminal channel and digital signatures," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 364–378, Springer, Berlin, Germany, 1984.

[36] A. Kurt, E. Erdin, M. Cebe, K. Akkaya, and A. S. Uluagac, "LNBot: a covert hybrid botnet on bitcoin lightning network for fun and profit," in *Computer Security – ESORICS 2020*, L. Chen, N. Li, K. Liang, and S. Schneider, Eds., Springer, Berlin, Germany, 2020.

[37] R. Pass, "Micropayments for decentralized currencies," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 207–218, New York, NY USA, October 2015.

[38] B. Wiki, "Rapidly-adjusted (micro) payments to a predetermined party," 2019, https://en.bitcoin.it/wiki/Contract#.

[39] S. Nagaraja and A. Houmansadr, "Stegobot: a covert social network botnet," in *Information Hiding using Steganography and LSB Technique*, pp. 299–313, Springer, Berlin, Germany, 2011.

[40] X. Cui, B. Fang, J. Shi et al., "Botnet triple-channel model: towards resilient and efficient bidirectional communication, botnets," in *Proceedings of the 9th International Conference on.Security and Privacy in. Communication. Networks*, pp. 53–68, London, UK, September 2013.

[41] L. Böck, N. Alexopoulos, E. Saracoglu, M. Mühlhäuser, and E. Vasilomanolakis, "Assessing the threat of blockchain-based botnets," in *2019 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–11, Pittsburgh, PA, USA, May 2019.

[42] M. Casenove and A. Miraglia, "Botnet over tor: the illusion of hiding," in *6th international conference on cyber conflict, CyCon 2014*, pp. 273–282, Tallinn, Estonia, June 2014.

[43] N. Aniello Castiglionea, A. Roberto De Prisco, A. Alfredo De Santis, B. Ugo Fiore, and F. Palmieric, "A botnet-based command and control approach relying on swarm intelligence," *Journal of Network and Computer Applications*, vol. 38, pp. 22–33, 2014.

[44] K. Li, B. Fang, X. Cui, and Q. Liu, "Research on the development of botnets," *Computer Research and Development*, vol. 53, no. 10, pp. 2189–2206, 2016.

[45] P. Wang, L. Wu, R. Cunningham et al., "Honeypot detection, in advanced botnet attacks," *International Journal of Information and Computer Security*, vol. 4, no. 1, pp. 30–51, 2010.

[46] H. Wang, Z. Gong, and J. Hou, "Overview of botnet detection," *Computer Research and Development*, vol. 47, no. 12, pp. 2037–2048, 2010.

[47] Y. Xie, Y. Fang, and K. Achan, "Spamming botnets signatures and characteristics," *Computer Communication Review*, vol. 38, no. 4, pp. 171–182, 2008.

[48] L. Liu, S. Chen, G. Yan et al., "Bot Tracer: execution-based bot-like malware detection," in *Proceedings of the 11th international conference on Information Security*, pp. 97–113, Taipei, Taiwan, September 2008.

[49] G. Gu, P. Porras, V. Yegneswaran et al., "BotHunter: datecting malware infection through ids-driven dialog correlation," in *Proceedings of the of the 16th USENIX Security Symp(Security'07)*, pp. 167–182, Berkeley, CA, USA, August 2017.

[50] A. H. Lashkari, G. D. Gil, J. E. Keenan et al., "A survey leading to a new evaluation framework for network-based botnet detection," in *Proceedings of the 2017 the 7th international conference on communication and network security*, pp. 59–66, Bombay, India, July 2017.

[51] J. Wang and Y. Chen, "Botnet detection method based on permutation entropy and clustering variance," *DEStech Transactions on Engineering and Technology Research*, vol. 71, 2017 (ismii).

[52] X. Yu, X. Dong, G. Yu et al., "Data-adaptive clustering analysis for online botnet detection," in *Proceedings of the Computational science and optimization (CSO), 2010 third international joint conference on. IEEE*, pp. 456–460, Auhui, China, May 2010.

[53] C. Livadas, R. Walsh, D. Lapsley et al., "Usilng machine learning technliques to identify botnet traffic," in *Proceedings of the 31st IEEE Conference on Local Computer Networks*, pp. 967–974, Tampa, FL, USA, December 2006.

[54] S. Kondo and N. Sato, "Botnet traffic detection techniques by c&c session classification using svm," in *Proceedings of the International Workshop on Security*, pp. 91–104, Nara, Japan, October 2007.

[55] L. Bilge, D. Balzarotti, W. Robertson et al., "Detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 129–138, World Resort, FL, USA, December 2012.

[56] J. François, S. Wang, and T. Engel, "Bot Track: tracking botnets using net flow and page rank," in *Proceedings of the International Conference on Research in Networking*, pp. 1–14, Bangalore, India, January 2011.

[57] G. Gu, R. Perdisci, J. Zhang et al., "Bot miner: clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proceedings of the USENIX Security Symposium*, pp. 139–154, San Jose, CL, USA, May 2008.

[58] K. Guang, G. Tang, S. Wang, H. Song, and Y. Bian, "Using deep learning for detecting Bot cloud," *Journal of Communications*, vol. 37, no. 11, pp. 114–128, 2016.

[59] W. Jung, H. yang, M. Zhao, L. Sun, and G. Zhou, "IoT botnet detection via power consumption modeling," *Smart Health Smart Health*, vol. 15, Article ID 100103, 2020.

[60] R. Vinayakumar, P. Poornachandran, and K. P. Soman, "Scalable framework for cyber threat situational awareness based on domain name systems data analysis," in *Big Data in Engineering Applications*Springer, Berlin, Germany, 2018.

[61] Z. Feng, C. Shuo, and W. Xiaochuan, "Classification for DGA-based malicious domain names with deep learning architectures," in *Proceedings of the 2017 Second International Conference on Applied Mathematics and Information Technology*, Shanghai, China, December 2017.

[62] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of recurrent neural networks for botnet detection behavior," in *Biennial Congress of Argentina (ARGENCON)* Springer, Berlin, Germany, 2016.

[63] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, December 2018.

[64] L. Fernández Maimó, Á. L. Perales Gómez, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, "A self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[65] Y. Zhang and Z. O. U. Fu-Tai, "Detection method of malicious domain name based on knowledge map," *Communications Technology*, vol. 53, no. 1, pp. 168–173, 2020.
[66] D. Wu, B. Fang, X. Cui, and Q. Liu, "Bot Catcher: botnet detection system based on deep learning," *Journal of Communications*, vol. 39, no. 8, pp. 18–28, 2018.
[67] W. Niu, T. Jiang, and X. Zhang, "Fast-flux botnet detection method based on the temporal and spatial characteristics of traffic," *Journal of Electronics and Information*, vol. 42, no. 8, pp. 1872–1880, 2020.
[68] C. Yin, *Research on Network Anomaly Detection Technology Based on Deep Learning*, University of Information Engineering, Strategic Support Forces, Zhengzhou, China, 2018.
[69] K. Zhao, Glansheng, F. Qin, and X. Hong, "Deep model for DGA botnet detection based on word-hashing," *Journal of Southeast University (Natural Science Edition)*, vol. 47, no. S1, pp. 30–33, 2017.
[70] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the internet of things networks of smart cities," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4436–4456, 2020.
[71] X. Pei, S. Tian, L. Yu et al., "A two-stream network based on capsule networks and sliced recurrent neural networks for DGA botnet detection," *Journal of Network and Systems Management*, vol. 28, pp. 1694–1721, 2020.
[72] A. Almomani, "Fast-flux hunter: a system for filtering online fast-flux botnet," *Neural Computing and Applications*, vol. 29, pp. 483–493, 2018.
[73] M. Alauthman, N. Aslam, M. Alkasassbeh, S. Khan, A. AL-qerem, and K.-K. Raymond Choo, "An efficient reinforcement learning-based Botnet detection approach," *Journal of Network and Computer Applications*, vol. 52, 2019.
[74] Y. M. Mahardhika, A. Sudarsono, and A. R. Barakbah, "Botnet detection using on-line clustering with pursuit reinforcement competitive learning (PRCL)," *EMITTER International Journal of Engineering Technology*, vol. 6, no. 1, pp. 1–21, 2018.
[75] D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui, "Evading machine learning botnet detection models via deep reinforcement learning," in *Proceedigs of the ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.
[76] F. Haddadi, D. T. Phan, and A. N. Zincir-Heywood, "How to choose from different botnet detection system," in *Proceedings of the Network Operations and Management Symposium (NOMS)*, pp. 1079–1084, Istanbul, Turkey, April 2016.
[77] Does Alexa have a list of its top-ranked websites.
[78] OSINT Feeds from Bambenek Consulting," Bambenek Consulting.
[79] Lab, accessed: 2019-07–20. http://https//b.360.com/dga/.
[80] A. Abakumov: http://https//githu%20b.com/andre%20waeva/DGA.
[81] D. O. G. Szab and S. Malomsok, *ISOT Botnet Dataset*, University of Victoria, Victoria, Canada, 2010, http://www.uvic.ca/engineering/ece/isot/datasets/.
[82] E. Biglar, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Proceedings of the 2014 IEEE Conference on IEEE 2014*, Toronto, Canada, May 2014.
[83] H.-T. Nguyen, Q.-D. Ngo, D.-H. Nguyen et al., "PSI-rooted subgraph: a novel feature for iot botnet detection using classifier algorithms," *ICT Express*, vol. 42, 2020.
[84] Z. Yao, Y. Xie, Y. Fang et al., "BotGraph: large scale spamming botnet detection , NSDI '09," in *Proceedings of the 6th USENIX symposium on networked systems design and implementation*, Boston, MA, USA, April 2009.
[85] S. Chowdhury, M. Khanzadeh, R. Akula et al., "Botnet detection using graph-based feature clustering," *Journal of Big Data*, vol. 414 pages, 2017.
[86] J. François, S. Wang, R. State, and T. Engel, "BotTrack: tracking botnets using netflow and pagerank," in *NETWORKING 2011. NETWORKING 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds., Springer, Berlin, Heidelberg, 2011.
[87] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," 1998.
[88] Z. Xiong, *Research on Botnet Traffic Detection Methods for Fast-Flux and Domain-Flux*, University of Electronic Science and Technology, Chengdu, China, 2019.
[89] D. Zhuang and J. M. Chang, "PeerHunter: detecting peer-to-peer botnets through community behavior analysis," in *Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing*, pp. 493–500, Taipei, China, September 2017.
[90] Q. Yan, Y. Zheng, T. Jiang, W. Lou, and Y. T. Hou, "PeerClean: unveiling peer-to-peer botnets through dynamic group behavior analysis," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 316–324, Kowloon, Hong Kong, April 2015.
[91] J. Wang and I. C. Paschalidis, "Botnet detection based on anomaly and community detection," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 392–404, 2017.
[92] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: a novel honenypot for revealing current IoT threats," *Journal of Information Processing Systems*, vol. 24, pp. 522–533, 2016.
[93] VirusShare, "Because sharing is caring," 2019, https://virusshare.com/.
[94] Dataset, "The CAIDA UCSD DDoS Attack 2007," 2013, http://www.caida.org/data/passive/ddos-20070804%20dataset.%20xml.
[95] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 2809–2825, 2020.
[96] M. Habib, I. Aljarah, H. Faris, and S. Mirjalili, "Multi-objective particle swarm optimization for botnet detection in internet of things," in *Evolutionary Machine Learning Techniques*, S. Mirjalili, H. Faris, and I. Aljarah, Eds., Springer, Berlin, Germany, 2020.
[97] M. Moodi, M. Ghazvini, H. Moodi et al., "A smart adaptive particle swarm optimization–support vector machine: android botnet detection application," *The Journal of Supercomputing*, vol. 76, pp. 9854–9881, 2020.
[98] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 2809–2825, 2020.
[99] N. Moustafa, "The Bot-IoT dataset," *IEEE Dataport*, vol. 5, 2019.
[100] M. Moodi and M. Ghazvini, "A new method for assigning appropriate labels to create a 28 standard android botnet dataset (28-SABD)," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 4579–4593, 2018.

[101] Y. Meidan, M. Bohadana, Y. Mathov et al., "N-baiot: network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computer*, vol. 13, no. 9, pp. 12–22, 2018.

[102] S.-Y. Huang, C.-H. Mao, and H.-M. Lee, "Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 101–111, Beijing China, April 2010.

[103] S. Garg, M. Guizani, S. Guo, and C. Verikoukis, "Guest editorial special section on AI-driven developments in 5G-envisioned industrial automation: big data perspective," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1291–1295, 2020.

[104] X. Wang, Q. Yang, and X. Jin, "Periodic communication detection algorithm of botnet based on quantum computing," *Journal of Quantum Electronics*, vol. 33, no. 2, pp. 182–187, 2016.

[105] M. Albanese, S. Jajodia, and S. Venkatesan, "Defending from stealthy botnets using moving target defenses," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 92–97, 2018.

[106] J. Sonchack, J. M. Smith, A. J. Aviv, and E. Keller, *Enabling Practical Software-Defined Networking Security Applications With Ofx*, In NDSS, West Bengal India, 2016.

[107] Z. Zha, A. Wang, Y. Guo, D. Montgomery, and S. Chen, "BotSifter: an SDN-based online bot detection framework in data centers," in *Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 142–150, Washington DC, DC, USA, November 2019.

[108] X. Cheng, *Research and Implementation of Botnet Detection Method under Software Defined Network*, Wuhan University, Wuhan, China, 2017.

[109] G. Sagirlar, B. Carminati, and E. Ferrari, "Autobotcatcher: blockchain-based p2p botnet detection for the internet of things," in *Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pp. 1–8, Philadelphia, PA, USA, July 2018.

[110] W. She, Q. Liu, Z. Tian, J. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38947–38956, 2019.

[111] G. Spathoulas, N. Giachoudis, G.-P. Damiris, and G. Theodoridis, "Collaborative blockchain-based detection of distributed denial of service attacks based on internet of things botnets," *Future Internet*, vol. 11, p. 226, 2019.

[112] B. Wu, Q. Li, K. Xu, R. Li, and Z. Liu, "SmartRetro: blockchain-based incentives for distributed IoT retrospective detection," in *Proceedings of the 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 308–316, Chengdu, China, October 2018.

[113] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *Journal of Computer Networks and Communications*, vol. 2020, Article ID 9024726, 16 pages, 2020.

[114] R. U. Khan, R. Kumar, M. Alazab, and X. Zhang, "A hybrid technique to detect botnets, based on P2P traffic similarity," in *Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC)*, pp. 136–142, Melbourne, Australia, May 2019.

[115] K. Li, *Research on Botnet Countermeasures Based on Behavioral Analysis*, Beijing University of Posts and Telecommunications, Beijing, China, 2017.

[116] T. Ghosh, E. El-Sheikh, and W. Jammal, "A multi-stage detection technique for DNS-tunneled botnets," *Canadian Art Therapy Association*, vol. 58, pp. 137–143, 2019.

[117] A. Karim, R. B. Salleh, M. Shiraz et al., "Botnet detection techniques: review, future trends, and issues," *Journal of Zhejiang University-Science C.*vol. 15, pp. 943–983, 2014.

[118] M. Szymczyk, "Detecting botnets in computer networks using multi-agent technology," in *Proceedings of the IEEE 4th International Conference on Dependability of Computer Systems*, pp. 192–201, Brunow, Poland, July 2009.

[119] K. Sinha, V. Arun, and B. Julian, "Tracking temporal evolution of network activity for botnet detection," 2013, https://arxiv.org/abs/1908.03443.

[120] P. Tiago, "Peixoto. the graph-tool python library. figshare," 2014.