IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 December 2024, accepted 2 January 2025, date of publication 7 January 2025, date of current version 21 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526711

RESEARCH ARTICLE

Deep Learning-Based Intrusion Detection System for Detecting IoT Botnet Attacks: A Review

TAMARA AL-SHURBAJI[1], MOHAMMED ANBAR[1], (Member, IEEE), SELVAKUMAR MANICKAM[1], (Member, IEEE), IZNAN H HASBULLAH[1], NADIA ALFRIEHAT[1], BASIM AHMAD ALABSI[2], AHMAD REDA ALZIGHAIBI[3], AND HASAN HASHIM[3]

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Minden Heights, Pinang 11800, Malaysia
[2]Applied College, Najran University, Najran 61441, Saudi Arabia
[3]Department of Information Systems, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia

Corresponding author: Mohammed Anbar (anbar@usm.my)

ABSTRACT The proliferation of Internet of Things (IoT) devices has brought about an increased threat of botnet attacks, necessitating robust security measures. In response to this evolving landscape, deep learning (DL)-based intrusion detection systems (IDS) have emerged as a promising approach for detecting and mitigating botnet activities in IoT environments. Therefore, this paper thoroughly reviews existing literature on botnet detection in the IoT using DL-based IDS. It consolidates and analyzes a wide range of research papers, highlighting key findings, methodologies, advancements, shortcomings, and challenges in the field. Additionally, we performed a qualitative comparison with existing surveys using author-defined metrics to underscore the uniqueness of this survey. We also discuss challenges, limitations, and future research directions, emphasizing the distinctive contributions of our review. Ultimately, this survey serves as a guideline for future researchers, contributing to the advancement of botnet detection methods in IoT environments and enhancing security against botnet threats.

INDEX TERMS Intrusion detection system (IDS), botnet, deep learning, Internet of Things (IoT), IoT Botnet, neural networks.

I. INTRODUCTION

In 2016, several prominent companies and government institutions in Europe and America fell victim to the Mirai botnets [1], which primarily conducted Distributed Denial of Service (DDoS) attacks. These botnets exploited vulnerabilities in IoT devices to launch massive traffic floods, overwhelming the targeted systems and causing significant disruptions. While DDoS attacks are the traditional objective of botnets, recent trends have shown a diversification in botnet activities, including data theft, spam distribution, and in some cases, ransomware attacks. The primary goal remains to exploit the compromised devices for various malicious activities.

Cybersecurity analysts employ various techniques to detect these malicious networks, including intercepting packets, analyzing botnet structures, assessing network traffic, and employing signature detection [1], [2]. However, botnet structures constantly evolve, with attackers even utilizing local network connections to evade detection. The Internet of Things (IoT) faces persistent threats from botnets partly due to the absence of standards and the complexities of designing IoT protocols and sensors. Security experts face significant challenges when investigating and mitigating these security incidents [3]. Detecting botnet attacks in IoT networks is critical due to their potential harm to the entire system. While Deep Learning (DL) models have demonstrated remarkable performance in identifying botnet attacks in IoT networks, existing studies have primarily focused on using a single DL model [4], [5], [6]. This approach may not be effective in adapting to dynamic network conditions and the

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

ever-evolving nature of botnets. Moreover, relying solely on a single DL model may struggle with data heterogeneity, class imbalance, and achieving a balanced trade-off among multiple objectives.

There is a compelling need to explore the potential benefits of ensemble learning models for botnet detection in IoT networks, aiming to address the limitations. However, a significant gap exists in the literature regarding using ensemble learning models for botnet detection within IoT networks. Consequently, an investigation into the comparative performance of ensemble learning models versus single DL models is warranted. Such an inquiry holds promise for enhancing the effectiveness and robustness of botnet detection systems in IoT networks.

This review aims to comprehensively examine the current state of DL-based intrusion detection systems (IDS) for botnet detection in IoT environments. It consolidates and analyzes a wide range of research papers, highlighting key findings, methodologies, advancements, shortcomings, and challenges in the field. By focusing explicitly on the application of DL techniques for botnet detection, this review delves into the intricacies of various DL models such as CNNs, RNNs, and GANs in IoT environments.

In this context, we outline the main contributions of this paper as follows.

- Comprehensive review: The paper thoroughly reviews existing literature on botnet detection in the IoT using DL-based IDS. It consolidates and analyzes a wide range of research papers, highlighting the key findings, methodologies, advancements, shortcomings, and challenges in the field.
- Specialized focus: The paper emphasizes the specific application of DL techniques for botnet detection in the IoT. By narrowing the scope, it digs deeper into the intricacies and nuances of DL models such as CNNs, RNNs, and GANs in IoT environments.
- The review paper distinguishes itself by having a specific domain focus on the application that utilizes DL techniques to detect botnets on the IoT.
- The proposed review paper discusses different DL models, including CNNs, RNNs, and GANs, utilized for botnet detection.
- It examines the effectiveness and applicability of different DL models utilized for botnet detection within IoT environments and the coverage compared with existing surveys and reviews.
- By addressing challenges posed by IoT networks, the proposed review offers valuable insights for researchers and practitioners aiming to develop robust and efficient DL-based IDSs for botnet detection in IoT.
- Future research directions: The paper identifies key areas for future research in botnet detection in IoT using DL. It suggests avenues for further exploration, such as hybrid ensemble methods, dynamic ensemble feature selection techniques, multi-objective optimization approaches, and domain-specific feature selection methods. These future research directions guide researchers and practitioners in advancing the field and addressing emerging challenges.

This review paper is organized as follows: Section II introduces the research background, detailing the IoT concepts, applications, privacy, security, vulnerabilities, IDSs, and DL. Section III thoroughly analyzes related studies, exploring their methodologies, techniques, and findings to understand the existing field's status. Section IV offers a critical review. Section V focuses on potential future research directions in botnet detection in IoT using DL. It pinpoints areas needing more exploration and emphasizes new trends and challenges, guiding researchers and practitioners. Section VI compares related reviews in the field qualitatively, underscoring this paper's distinctive contributions and insights and affirming its originality and worth. Section VII concludes the paper, summarizing its significant findings, contributions, and implications.

II. BACKGROUND

This section provides an overview of IoT technology and discusses the security challenges faced by IoT. Additionally, it presents an overview of the DL-based Intrusion Detection System (IDS) used for detecting cyber-attacks in IoT networks.

A. BASIC CONCEPTS OF IoT

Despite its widespread use, there is no standardized definition of 'IoT' [7]. Although various definitions have been proposed in the literature by researchers [8], [9], [10], the overarching goal of IoT remains the same: to collect and share data through networks of uniquely identifiable endpoints or "things." This paper adopts the following definition of IoT: "a network of devices embedded with electronics, software, sensors, and actuators capable of exchanging information through communication networks, such as the Internet" [11]. The architecture of IoT consists of multiple layers of technology that work together to achieve its goals. Therefore, to fully understand IoT, it is essential to review its architecture. According to [12](p. 355), there is currently no standardization for IoT systems' technical specifications and reference architectures. Typically, IoT communication architectures allow devices to connect to the internet and communicate autonomously. Although there is no agreed-upon standard for IoT architecture, researchers, authors, and practitioners have developed five architectural models, all of which share similar components. Generally, an IoT system is composed of three layers: (i) a physical perception layer, (ii) a network layer, and (iii) an application layer. Reference [7] describe an IoT technology stack consisting of three core layers: (i) the device layer, (ii) the connectivity layer, and (iii) the IoT cloud layer. Reference [13] Also, IoT has three layers: (i) the device layer, (ii) the connection layer, and (iii) the application layer. Sensors collect and analyze data; cloud-based applications are crucial for interpreting and transmitting data from multiple sensors. presents a simplified

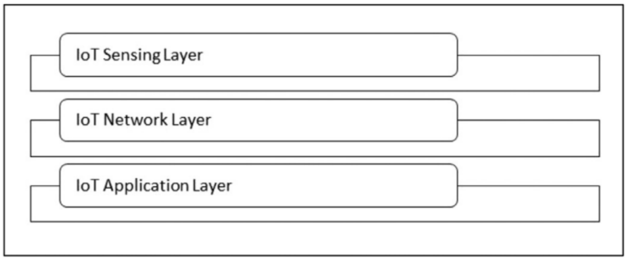

FIGURE 1. Simplified IoT architecture, source: [14].

IoT architectural model. Although understanding IoT architecture is essential for development teams, researchers and practitioners are likely more interested in IoT applications.

IoT serves as a network infrastructure that facilitates the connection of various devices through sensory, communication, networking, and information-processing technologies [15]. Among the key technologies employed in IoT, Radio-Frequency Identification (RFID) plays a significant role. As discussed by [16], RFID enables the wireless transmission of identification information from microchips to a reader, allowing objects equipped with RFID tags to be automatically identified, tracked, and monitored. From the 1980s onwards, RFID technology has been widely applied in various sectors, including logistics, pharmaceutical production, retail, and supply chain management [17]. Another pivotal technology for IoT is Wireless Sensor Networks (WSNs), which utilize interconnected intelligent sensors to gather data and monitor the environment. WSNs have diverse applications in areas such as environmental monitoring, healthcare monitoring, industrial monitoring, traffic monitoring, and more [18]. The progress of IoT has been significantly aided by advancements in RFID and WSN technology. Other technologies such as barcodes, smartphones, social networks, and cloud computing are also employed to support IoT [19], [20], as shown in Figure 2.

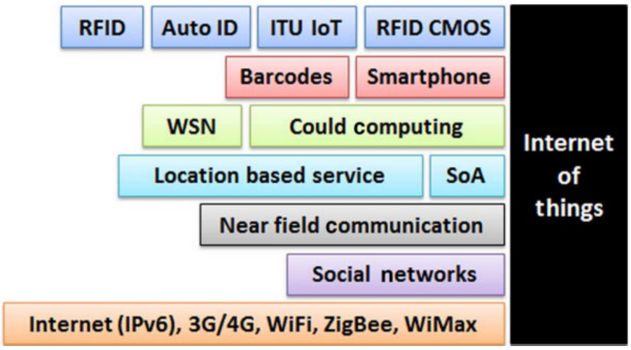

FIGURE 2. Technologies associated with IoT, source: [21].

The logistics, manufacturing, retail, and pharmaceutical industries have embraced IoT in recent years. The advancements in wireless communication, smartphone technology, and sensor networks have facilitated the integration of many smart objects into the IoT ecosystem. These IoT-related technologies have transformed how businesses operate and profoundly impacted the development of new Information and Communication Technology (ICT) and enterprise system technologies, as shown in Figure 3.

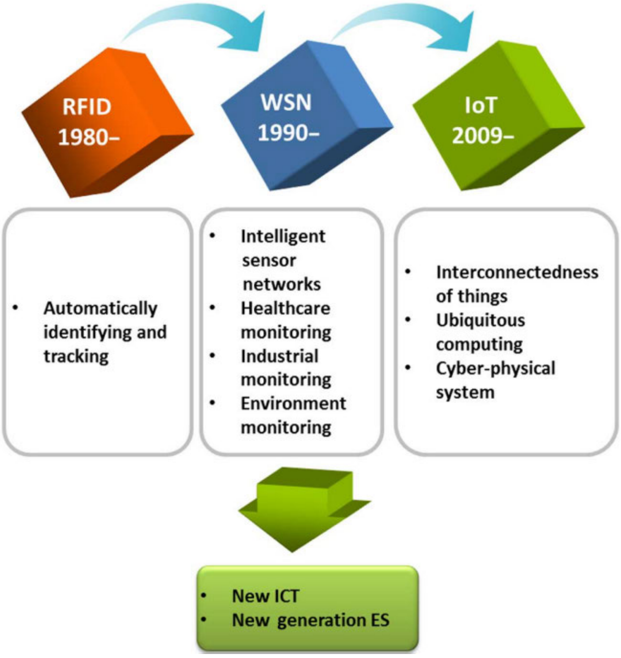

FIGURE 3. IoT-related technology, source: [21].

- Service-Oriented Architecture (SOA) for IoT:

IoT aims to establish connections between diverse devices and systems using networks. Service-oriented architecture (SOA) can be a pivotal technology for integrating heterogeneous systems or devices [22]. SOA has proven effective in various research domains, including cloud computing, WSNs, and vehicular networks [23], [24]. Various approaches have been proposed to develop multi-layer SOA frameworks tailored to the IoT, considering selected technologies, business requirements, and technical specifications. The International Telecommunication Union recommends an IoT architecture composed of five layers: (i) sensing, (ii) accessing, (iii) networking, (iv) middleware, and (v) application layers. Additionally, alternative architectures have been proposed, such as three-layered models that consist of (i) a perception layer, (ii) a network layer, and (iii) a service layer or application layer. Another three-layered model encompasses (i) an application layer, (ii) a network layer, and (iii) a sensing layer [21]. Figure 4 illustrates an SOA comprising four layers, showcasing the interconnectedness of each layer. The architectural design of IoT encompasses various factors, including architecture styles, networking and communication protocols, integration of smart objects, development of web services and applications, consideration of business models and processes, cooperative data processing, and ensuring security measures. Additionally, an IoT architecture should possess extensibility, scalability, modularity, and interoperability to accommodate diverse and heterogeneous devices effectively. It is crucial to employ an adaptive architecture that enables dynamic interactions among connected entities to accommodate IoT devices' mobility and real-time interaction requirements. Given IoT's decentralized and heterogeneous nature, architecture must offer efficient event-driven capabilities to support seamless communication. SOA is widely recognized as an effective approach to achieving interoperability between diverse and heterogeneous devices in multiple contexts [25], [26].

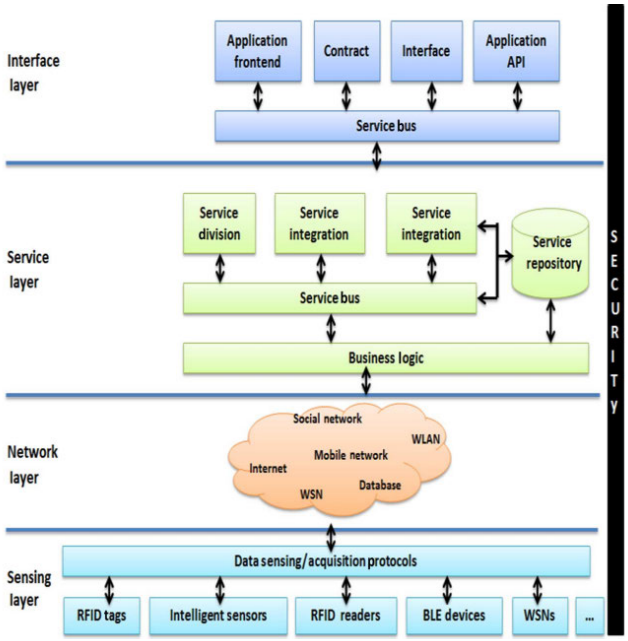

FIGURE 4. SOA for IoT, source: [21].

- Sensing layer:

The IoT's sensing layer comprises a globally interconnected network of physical devices that allow remote control and connectivity. As more devices incorporate RFID or smart sensors, linking objects becomes simpler. Wireless intelligent systems equipped with tags or sensors automatically detect and exchange information among various devices within the sensing layer. These technological advancements significantly enhance the IoT's ability to identify and perceive objects and the environment. Each device or service is assigned a unique Universal Unique Identifier (UUID) for intelligent service deployment across industry sectors. UUIDs enable effortless recognition and retrieval of devices, making them essential for successful service deployment in a vast network like the IoT [27].

- Networking layer:

The networking layer is critical in connecting and sharing information between interconnected devices. It also can aggregate data from various IT infrastructures, including transportation, healthcare, and power grids. Under the SOA-IoT paradigm, services offered by devices are usually deployed in a heterogeneous network, and all relevant devices are brought into the service internet. Quality-of-service (QoS) management and control might be necessary to satisfy user and application requirements. In a dynamic network, it is imperative to discover and map devices automatically. Devices must be assigned roles that enable them to deploy, manage, and schedule their behavior and change to other roles as necessary to accomplish tasks collaboratively. The designers responsible for the networking layer in IoT systems face various challenges that necessitate careful consideration. These challenges include incorporating network management technologies suitable for heterogeneous networks (such as fixed, wireless, and mobile networks), optimizing energy efficiency, meeting QoS requirements, implementing efficient service discovery and retrieval mechanisms, managing data and signal processing, and ensuring robust security and privacy measures [27], [28].

- Service layer:

Middleware technology plays a crucial role in the service layer of IoT by providing functionalities to integrate services and applications seamlessly. This cost-efficient platform also enables the reuse of hardware and software platforms. Developing service specifications for middleware is a primary activity in the service layer, and various organizations are currently undertaking it. A well-designed service layer can identify common application requirements and provide APIs and protocols to support necessary services, applications, and user needs. This layer handles service-oriented issues such as information exchange and storage, data management, search engines, and communication. One of the essential components of the service layer is service discovery, which efficiently locates objects capable of offering the required services and information [21].

- Interface layer:

Within the realm of IoT, devices sourced from different manufacturers or vendors often adhere to distinct protocols or standards, leading to potential complications in terms of their interoperability. These challenges can manifest in information exchange, communication, and collaborative event processing. Moreover, the escalating quantity of devices involved in IoT introduces difficulties in dynamically establishing connections, facilitating communication, terminating connections, and overall device operation. Implementing an interface layer becomes essential to mitigate these complexities and streamline the management and interconnection of IoT devices. An Interface Profile (IFP) serves as a subset of service standards that facilitates seamless interaction between applications on a network. A reliable and effective IFP can be realized by leveraging Universal Plug and Play (UPnP), a protocol that defines a standardized approach for interconnecting services offered by diverse devices. Interface profiles are crucial in describing the specifications and requirements for the interaction between applications and services. Services on the service layer are directly deployed on constrained network infrastructures, enabling applications to discover and utilize new services as they connect to the network. The SOCRADES Integration Architecture (SIA) has become a proposed solution to foster efficient interaction between applications and services. While the service layer has traditionally provided a universal Application Programming Interface (API) for applications, recent research in the context of Service-Oriented Architecture for IoT (SOA-IoT) has revealed that the Service Provisioning Process (SPP) can also facilitate the interaction between applications and services. The SPP commences by sending a service request in a generic Web Services Description Language (WSDL) format, utilizing a ''type of queries'' approach. Subsequently, a ''candidate search'' mechanism is employed to identify potential services. The services are then ranked based on the ''Application context'' and QoS information. Finally, an ''On-Demand service provisioning'' mechanism is utilized to identify the most suitable service instance that aligns with the application's specific requirements [21], [29]. Finally, a ''Process Evaluation'' is used to evaluate the process.

B. APPLICATIONS OF IoT

As stated in [9], the IoT can be used in personal and large-scale business environments. While devices and networks enable physical connections, IoT applications offer reliable interactions between devices and humans. [30] IoT applications are classified into four domains: transportation and logistics, healthcare, smart environment, and personal and social [9], [31] identified manufacturing, retail trade, information services, finance, and insurance as the most valuable industries in terms of IoT adoption. A study involving 500 senior executives leading IoT initiatives revealed that energy, financial services, healthcare, and manufacturing were the most essential areas in their organizations [11]. [32] observed that smart homes, wearables, and smart cities dominated searches, tweets, and written content. Reference [33] classified the top 10 enterprise IoT projects into connected industry, smart city, and smart energy categories. Researchers have identified significant application areas in IoT, as summarized in Table 1 [9], [11], [30], [32], [33].

TABLE 1. Examples of IoT applications.

Reference	IoT Applications
[30]	• Transportation and logistics • Healthcare • Smart environment (home, office, plant) • Personal and social
[9]	• Manufacturing • Retail trade • Information services • Finance and insurance
[11]	• Financial services • Healthcare • Manufacturing • Retail • Information technology • Telecom • Transport
[32]	• Smart home • Wearables •Smart city •Smart grid •Industrial internet •Connected car •Connected health •Smart retail •Smart supply chain •Smart farming
[33]	•Connected industry •Wearables •smart city •Smart energy •Connected car •Connected car •Smart agriculture

C. PRIVACY AND SECURITY IN IoT

Every technological advancement has challenges, and the IoT is no different. IoT solutions comprise various technologies, leading to a complex and dynamic environment. The most significant obstacles to developing IoT capabilities include investment, security concerns, cooperation between departments, integration of different data, and a shortage of skilled personnel [11], Effective communication is crucial for maintaining data privacy and confidentiality within various IoT architectures. Regulations governing data collection, storage, and exchange should prioritize safeguarding users' records to uphold data privacy standards. Implementing secure key management practices and leveraging physically unclonable functions can significantly enhance security measures.

As IoT becomes increasingly integrated into daily life, the adoption of IoT-based devices is rising. Predictions suggest that 70% of devices will be IoT-based, with Cisco estimating a staggering 14.4 trillion devices in use by 2025. Machine-to-machine (M2M) traffic is projected to constitute up to 45% of Internet traffic by 2022.

In the healthcare sector, IoT-based applications are expected to contribute \$1.1 trillion to \$2.5 trillion in annual growth by 2025, with a global impact estimated at \$2.7 trillion to \$6.2 trillion. By 2025, an astounding 75 billion IoT devices will operate worldwide. However, the growth of IoT devices also attracts malicious actors seeking to exploit the technology. Symantec reported a 300%increase in cyber-attacks in 2019 compared to the previous year, with approximately 3 billion attacks recorded [34]. Figure 5 shows the common types of IoT-based network security in different aspects.

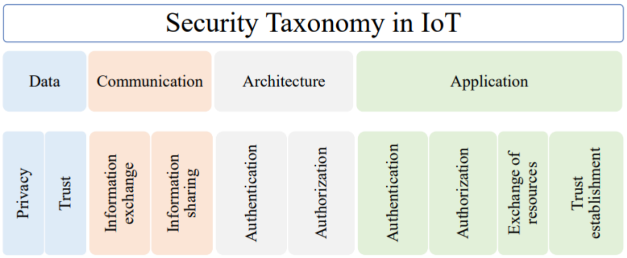

FIGURE 5. Taxonomy of IoT security, source: [34].

The expanding number and diverse nature of IoT devices contribute to a broadening attack surface, which is further compounded by factors such as population, heterogeneity, diversity, interoperability, portability, mobility, location, topology, and distribution of devices, controllers, connectivity, consumers, and services [35], [36].

Various enablers, including networks, protocols, and entities such as devices, methods, and information, influence the attack surface of IoT networks. It is determined by the interconnectivity of system components and the permissions granted to devices for system access. Attack surfaces manifest in different components of an IoT architecture, including administrative and device/cloud web interfaces, update mechanisms, mobile applications, physical interfaces, device firmware, and device memory. These attack surfaces are potential entry points for attackers, allowing them to exploit vulnerabilities and gain unauthorized access to a system. Consequently, attackers can manipulate or compromise sensitive information [37].

Each attack surface encompasses specific elements and device functionalities with inherent security vulnerabilities. As a result, comprehensive protection measures are essential. Despite traditional security approaches, IoT nodes with limited resources remain susceptible to attacks. Notably, the Mirai botnet and its derivatives exemplify threats – hijacking IoT devices to launch destructive DDoS attacks [31], [38]. Therefore, developing effective security solutions is crucial to address these challenges [39].

D. IoT VULNERABILITIES

The rise of IoT devices brings exciting experiences for consumers but also introduces security threats. Cybercriminals can exploit large quantities of data in this interconnected world, leading to data breaches if adequate security measures are not in place, which could expose sensitive information [34]. IoT devices lack built-in security features to protect against threats due to their low cost, limited power, and minimal computing capacity. Moreover, the scale and diversity of networks pose additional risks to these devices. Users' lack of security awareness and third-party apps further contribute to risks. Additionally, physical security is a concern since IoT components are accessible to users and malicious actors. These smart devices remain vulnerable for several reasons [40].

- Limited computing capabilities and hardware limitations: IoT devices are designed for specific applications, often requiring minimal processing power, leaving little space for integrating security and data protection measures.
- Heterogeneous transmission technology: These devices communicate with various devices and often use different communication technologies, making establishing uniform protection measures and protocols challenging.
- Vulnerable device components: Millions of smart devices can be compromised by insecure or outdated fundamental elements.
- Lack of security awareness among users: Due to a lack of knowledge regarding security measures, users may expose smart devices to risk zones and attack possibilities. Third-party apps can also introduce risks to IoT devices.
- Weak physical security: Unlike data centers for Internet services, IoT components are accessible to users and malicious actors.

Reference [41] considered the level of vulnerabilities and impacts to create the pyramid of threat factors shown in Figure 6. The topmost elements of the pyramid are more vulnerable but are less likely to have a significant impact. In contrast, the lower-level elements have a higher likelihood of causing significant impact. However, they are less susceptible, implying cyber threats usually target the lower levels of the stack to gain greater control and opportunities.

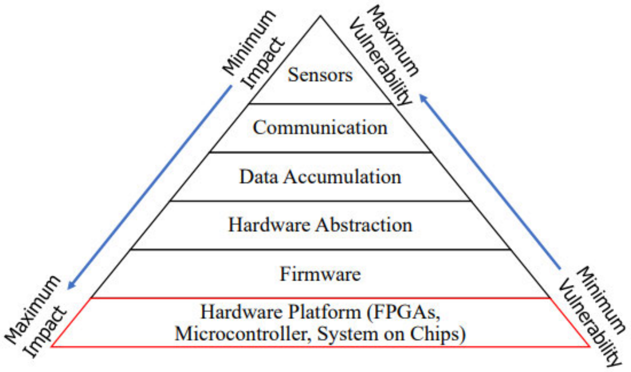

FIGURE 6. Pyramid of IoT devices vulnerability, source: [34].

E. INTRUSION DETECTION SYSTEM

IDS is a software or device that employs various detection techniques to identify attacks on a system and notify the system administrator. It can be a standalone device that monitors an individual system or a network system that conducts local analysis to detect potential attacks. Furthermore, IDSs provide three critical security services: (i) ensuring data confidentiality by verifying the secure storage of data within the system, (ii) ensuring data availability by checking if authorized users can access the data, and (iii) ensuring data integrity by verifying the accuracy and consistency of the data with other system data [42], [43].

1) TYPES OF IDS

Standalone IDSs include both network-based IDS and host-based IDS. In order to enhance the performance of IDS in large-scale IT ecosystems, multiple detectors are employed to correlate alerts and share information, forming what is known as Collaborative IDSs (CIDSs). CIDSs can be implemented using three network architectures: centralized, hierarchical, and distributed [44], as illustrated in Figure 7. In a centralized CIDS, multiple IDSs monitor the network, where each IDS connects to a single analysis unit to exchange data. In contrast, hierarchical and decentralized CIDSs also utilize multiple IDSs, but the analysis units are linked in a hierarchical configuration to oversee different network points. This configuration helps address the problem of a single point of failure. Finally, in a distributed CIDS, a peer-to-peer network architecture is adopted, where each participant has its own analysis unit and shares information with others in a distributed manner [45].

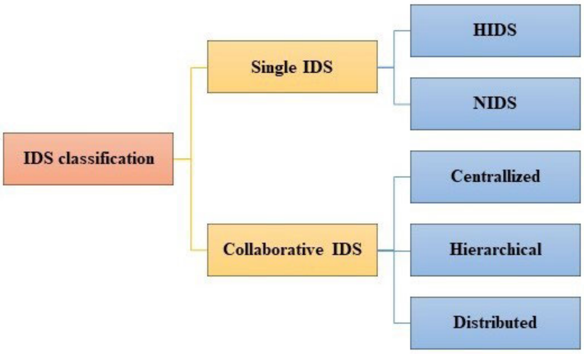

FIGURE 7. IDS categorization-based location.

2) IDS APPROACHES

The two primary approaches for IDS are the signature and anomaly approaches. The signature approach identifies attacks by matching predefined signatures or patterns in a database. It is effective in detecting known attacks, but it may not be able to detect new attacks that have no existing signatures. On the other hand, the anomaly approach monitors the behavior of the system to identify unknown attacks. It detects abnormal activities and alerts the network administrator. While the anomaly approach can detect unknown attacks, it may also generate false positive alarms. Both approaches utilize multiple techniques, as depicted in Figure 8.

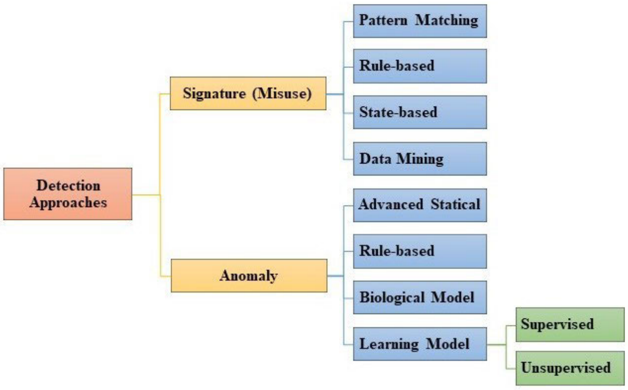

FIGURE 8. Detection approaches.

- **Pattern Matching** is a technique used in IDS to compare incoming strings with strings stored in the system's database to detect malicious attacks. When a matching pattern is found, the system generates an alarm to notify the administrator of a possible attack. There are two types of pattern-matching algorithms: single and multiple. Single pattern-matching algorithms are simpler as they search for one pattern at a time. In contrast, multiple pattern-matching algorithms search for all patterns simultaneously, which requires more time and resources. The Boyer-Moore algorithm is a well-known single pattern-matching algorithm commonly used in IDS. It compares strings from the rightmost character and has been proven to detect patterns effectively. Although the Boyer-Moore algorithm excels in search operations, it may face limitations regarding feature scalability. On the other hand, the Aho-Corasick and Wu-Manber algorithms are multiple pattern-matching algorithms that can search for multiple patterns simultaneously. However, the Aho-Corasick algorithm requires more memory than the Wu-Manber algorithm. Selecting a pattern-matching algorithm involves balancing search speed and memory usage. To enhance the detection performance of IDSs, researchers have proposed optimization techniques and the development of new algorithms [46].
- **The Rule-Based technique** is utilized in both the signature and anomaly approaches. In signature detection, malicious attacks are detected by diagnosing packets using predefined rules in the system. Meanwhile, in anomaly detection, the system's behaviour is diagnosed to detect differences between normal and abnormal behaviour based on predefined rules, such as programmers' sequence of system calls. Updating the rules in both detection methods is essential to increase the network's security. The signature approach makes updating the rules simple, easy, and automatic, whereas the anomaly detection approach requires more time to record new training rules, making updating the rules more complex [47].
- **The State-Based approach** employs signature detection to represent attack scenarios and consists of two main components: state and arc. The state represents a user or process, while the arc represents an action. An attack is identified when the user or process reaches the final state. The Unix State Transition Analysis Tool pioneered the state transition analysis technique, primarily used for host-based intrusion detection. It operates as a rule-based expert system that examines the audit traces of multi-user computer systems to identify known attacks. However, the tool has limitations, including challenges in extending or adapting its features to different operating systems [48].
- **Data Mining techniques** can enhance the signature detection approach by discovering new patterns for IDS and addressing its main disadvantage. While data mining is commonly used in the signature-based approach, there is also substantial research on applying it to

anomaly detection. However, data mining relies on various machine learning techniques, including rule-based methods, classification, and clustering, to gather knowledge for network intrusion detection [49].

- **Statistical-based IDS** Statistical-based IDS utilizes two profiles: one for monitoring current network traffic and the other for statistical training. When an event occurs, the system compares the two behaviors to assess it. If the anomaly score surpasses a predefined threshold, the IDS generates an alarm, indicating a potential intrusion [50]. Model-based statistics often employ multivariate statistical techniques, such as the chi-square statistic, Canberra technique, and Hotelling's T-squared distribution. These techniques help detect outliers in the dataset by analyzing behavioral patterns. Each element in the dataset possesses specific features and a local outlier factor, which can be utilized to identify abnormal behavior [51].
- **Biological Models:** Previous research has drawn parallels between the human immune system and computer network security, highlighting their similarities in complex network structures and the common objective of protecting nodes from malicious attacks. Both systems also employ security policies and maintain various levels of security. The human immune system utilizes natural selection phenomena to establish policies that fulfill disposability, correction, integrity, and accountability requirements. Similarly, computer networks establish rules to defend against attacks and detect unauthorized actions that violate specific security levels. In recent years, researchers have applied algorithms inspired by biological processes, such as genetic algorithms and artificial neural networks, to enhance the performance of intrusion detection within the anomaly detection approach [52].
- **Learning models:** have significantly enhanced the effectiveness of anomaly detection techniques. Anomaly detection can be categorized into two types: supervised and unsupervised. In supervised anomaly detection, the model is trained using labeled datasets that distinguish between normal and abnormal behavior. Support vector machines and k-nearest neighbor algorithms are examples of supervised anomaly detection techniques. On the other hand, unsupervised anomaly detection relies on unlabelled datasets and utilizes various techniques to differentiate between normal and abnormal behavior. One commonly used technique is clustering, which is applied in intrusion detection to identify outliers displaying anomalous behavior. The k-means clustering algorithm is the most popular among clustering algorithms and has been widely employed in intrusion detection [53].

DL has become a prominent area of machine learning in the realm of IDSs, as it employs artificial neural networks that emulate the structure and operation of the human brain to identify patterns in data. DL has demonstrated remarkable effectiveness in detecting and categorizing network intrusions by extracting high-level features from raw data. A significant advantage of DL-based IDSs is their adaptability to dynamic network environments and emerging types of attacks. Considering the burgeoning network traffic and sophisticated threats, DL-based IDSs are increasingly essential for safeguarding network security. Compared to machine learning-based techniques, DL techniques are more effective when dealing with large datasets. Consequently, DL has emerged as the most used IDS in network security. However, while DL models have shown considerable promise in IoT botnet detection, they are not without limitations. These limitations include generalization issues, lack of diversity, and challenges in handling imbalanced datasets, as discussed in [5] and [54]. Such challenges can adversely impact the effectiveness and reliability of IDS implementations, emphasizing the need for further optimization and complementary approaches. Ensemble learning methods, for instance, have demonstrated superior performance in addressing these challenges and enhancing detection robustness in specific contexts [55]. While DL remains a cornerstone in IDS development due to its ability to process large-scale data and adapt to sophisticated threats, comparative studies underline the importance of evaluating its performance alongside alternative methods to ensure comprehensive and resilient network security solutions. The following subsection introduces DL models in detail.

F. DEEP LEARNING

DL is a type of machine learning that uses artificial neural networks to learn from data [56]. Neural networks are inspired by the structure and function of the human brain, and they can learn complex patterns from large amounts of data. DL is a machine learning methodology with several features, including the ability to represent data effectively by transforming it into features that can be utilized to develop superior approaches for managing large amounts of data, resulting in a significant enhancement of classification performance and helping to overcome the constraints of shallow learning models. When analyzing network data and identifying intrusions, DL is more efficient and capable of detecting intrusions faster than other methods [57], [58], [59]. Figure 9. illustrates the performance of DL in contrast to machine learning.

DL is a specialized branch of Neural Networks (NNs) that falls under the broader umbrella of ML. DL has garnered significant attention due to its remarkable capabilities: it can automatically learn features from raw data, achieve outstanding outcomes, and operate comprehensively and efficiently. Additionally, DL architectures incorporate multiple hidden layers, as depicted in Figure 10, making them a preferred choice for handling vast amounts of data [60], [61], [62].

DL networks use two distinct learning methodologies: generative and discriminative learning. Identifying correlations among data to detect patterns in unsupervised learning is

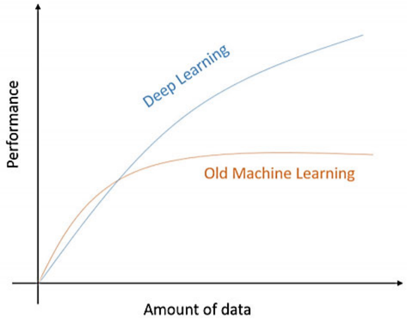

FIGURE 9. Performance of DL vs. ML [60].

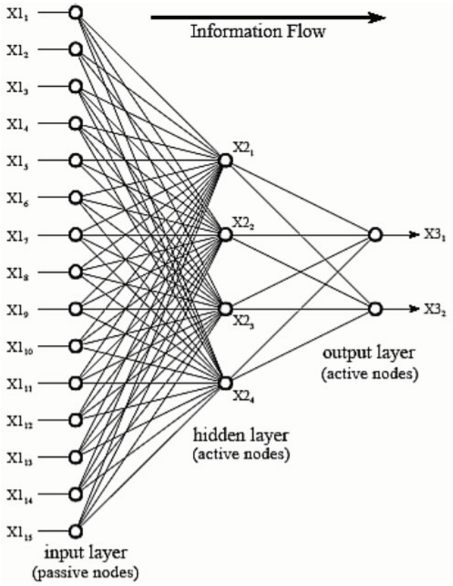

FIGURE 10. DL structure [61].

known as generative learning in deep networks. On the other hand, deep neural networks primarily employ supervised learning for classification purposes, while unsupervised learning is the focus of discriminative learning. For instance, as per [63], the Convolutional Neural Network (CNN) is a discriminative model.

1) CONVOLUTIONAL NEURAL NETWORK

CNNs find extensive applications in computer vision and image processing, including tasks like image classification, segmentation, face recognition, and object detection. Specifically designed for 2D images, the CNN architecture extracts pixel information during training and automatically learns relevant features. Figure 11 illustrates the fundamental components of CNNs, which typically include an input layer, convolution layers, pooling layers, one or more fully connected layers, and an output layer [64], [65].

In recent years, researchers have explored various innovative ideas for enhancing the performance of CNNs, such as parameter optimization and regularization techniques.

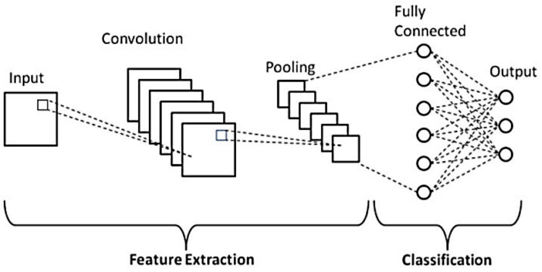

FIGURE 11. Architecture of CNN [66].

Furthermore, CNNs have been applied to numerous domains, including IDSs, where they have been shown to reduce the number of parameters required and streamline the training process once the intrusion type has been identified. This approach has been used in various domains [67].

2) RECURRENT NEURAL NETWORK

The Recurrent Neural Network (RNN) comprises three types of units: (i) input units, (ii) hidden units, and (iii) output units. The information flow in RNN is unidirectional, passing through the input, hidden, and output units. The hidden units in the RNN model play a crucial role, serving as memory units that retain information processed at time t. This mechanism is depicted in Figure 12 and has been studied by [68] and [69].

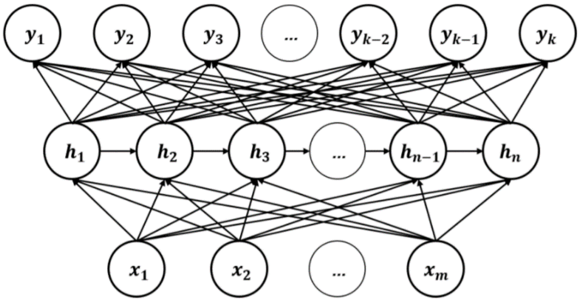

FIGURE 12. RNN architecture [63].

RNN is a well-established type of recurrent neural network that can be trained using supervised or unsupervised learning approaches. However, RNN has some limitations, such as the issue of vanishing gradients, which can impede further neural network training. RNN has been extended with various learning models, such as the Long Short-Term Memory (LSTM) network, to overcome these limitations. LSTM effectively resolves the vanishing gradient problem in RNN [70], [71].

3) LONG-SHORT TERM MEMORY (LSTM)

RNNs have feedback loops in their recurrent layer, allowing them to store information in their 'memory' over time. LSTM networks are a type of RNN that uses specialized units to address the vanishing gradient problem. These units include a 'memory cell' storing data for extended periods. Three gates control the functioning of LSTM units: the 'Forget Gate,' 'Input Gate,' and 'Output Gate.' These gates work together to regulate the information flow in an LSTM unit, as depicted in Figure 13. The Forget Gate determines which information from the previous state cell should be remembered and discarded, the Input Gate controls the information that enters the cell state, and the Output Gate controls the output [72].

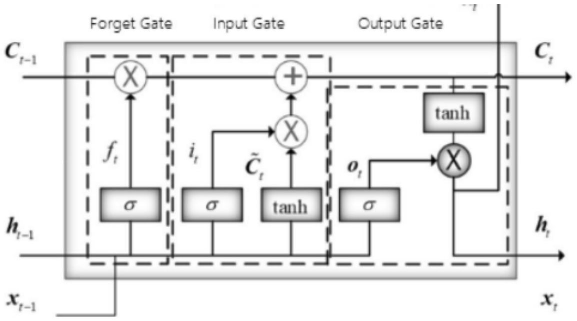

FIGURE 13. LSTM architecture, source: [72].

LSTM networks offer unique advantages in handling and analyzing sequential data, setting them apart from traditional networks. As a result, LSTM has found wide applications in various domains, including time-series prediction, time-series anomaly detection, natural language processing, question-answering chatbots, machine translation, speech recognition, and more. Given the abundance of sequential data in cybersecurity, such as network traffic flows and time-dependent malicious activities, LSTM models can also be effectively utilized in cybersecurity. Numerous security solutions leveraging LSTM models have been investigated in diverse areas, including intrusion detection [73], identification and categorization of malicious apps [74], phishing detection [75], and time-based botnet detection [76]. While the key advantage of recurrent networks over traditional ones lies in their ability to model sequential data, training them may demand substantial resources and time. Consequently, developing an effective LSTM-RNN network can enhance security models, particularly in detecting threats characterized by temporal dynamic behaviors.

4) GENERATIVE ADVERSARIAL NETWORKS (GAN)

In 2014, Ian Goodfellow and his colleagues introduced the Generative Adversarial Network (GAN), a DL model that can synthesize new images. GAN comprises two primary neural network models: (i) generative, which produces fake samples indistinguishable from the original samples, and (ii) discriminative, distinguishing between the original and fake samples. The objective of the discriminative model is to differentiate between the real and fake samples. The architecture of GAN is illustrated in Figure 14 and has been studied by [77], [78], [79].

GANs have been applied in image generation, natural language processing, time-series synthesis, and other fields. Researchers have also developed several methods to enhance GAN training, such as modifying the architecture, loss function, game theory, multi-agent, and gradient-based approaches. Figure 15 provides a taxonomy of different types of GAN variants. These advances in GAN technology have been discussed in [81] study. The standard GAN and its variants are popular models that have been used in several fields, as they can train data without the need for annotations. GAN is versatile and can be adapted for various applications based on unsupervised or semi-supervised learning, such as image classification and synthesis. Furthermore, GANs can be utilized to generate additional training samples, and they have achieved state-of-the-art performance on tasks such as pose and gaze estimation [82].

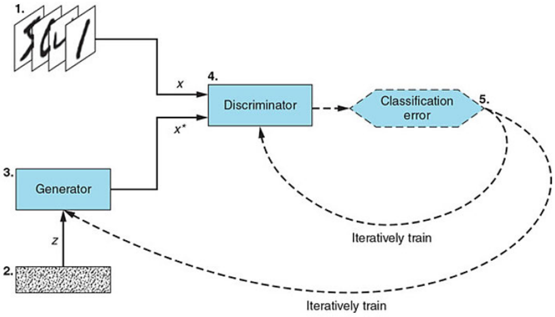

FIGURE 14. GAN architecture [80].

5) COMPARISON OF DEEP LEARNING MODELS FOR IoT INTRUSION DETECTION

In the previous sections, we discussed the various DL models, including CNNs, RNNs, LSTM networks, and Generative GANs. These models are widely used for various tasks in machine learning and have shown significant potential in IDS, particularly in IoT environments. A comparison of these models is provided below, summarizing their key characteristics, strengths, weaknesses, and applications, as presented in Table 2. This comparison will help clarify the distinctions between each model and outline their unique features, making it easier to understand how they contribute to improving IoT-based intrusion detection systems. The models discussed are essential for efficiently handling the challenges associated with IoT data, particularly the high volume, complexity, and dynamic nature of IoT environments.

III. RELATED WORKS

This section thoroughly examines existing literature and related works relevant to the research topic. Its primary objective is to provide an overview of the current knowledge in the field, pinpoint the research gaps that require attention, and critically analyze the limitations of existing approaches. The literature review is further divided into subsections, as shown in Figure 16. 3.1 Single detector-based IDS and 3.2 Ensemble detector-based IDS provide an in-depth analysis of the existing literature on the research topic, with a particular focus on the single detector-based IDS

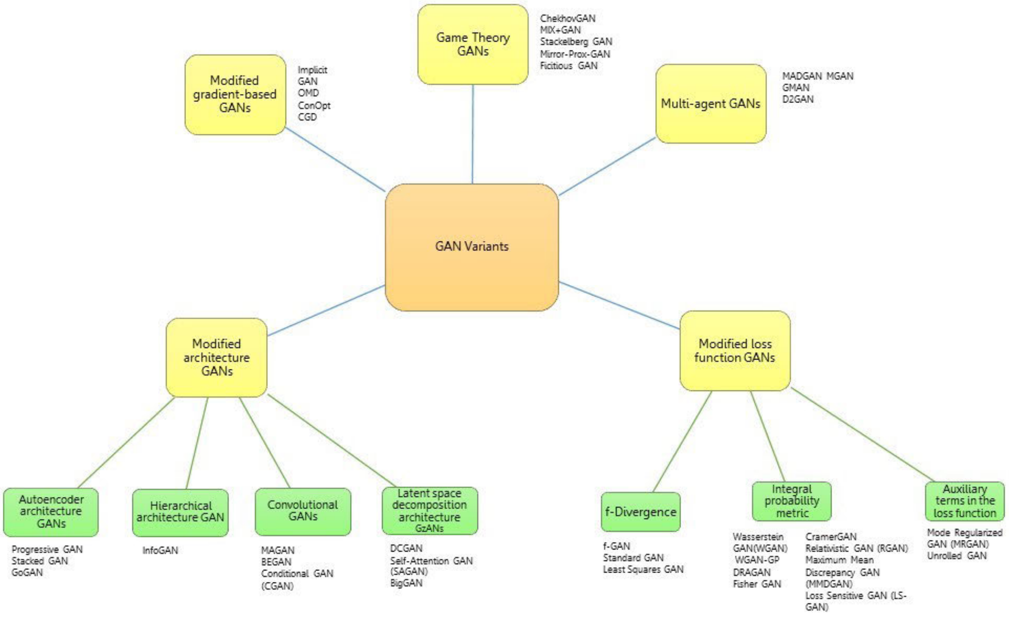

FIGURE 15. GANs variants [81].

TABLE 2. Comparison of deep learning models for IoT intrusion detection.

Model	Primary Use	Key Characteristics	Strengths	Weaknesses	Common Applications
CNN	Image and spatial data analysis	Uses convolutional layers to extract hierarchical features from spatial data	Exceptional at image classification, automatic feature extraction	Not suitable for sequential data (e.g., time-series)	Image classification, object detection, video analysis, IoT IDS.
RNN	Sequence prediction, time-series analysis	Uses feedback loops, recurrent connections for learning temporal dependencies	Effective for sequential data (text, time-series) and temporal patterns	Struggles with long-term dependencies (vanishing gradient problem)	Speech recognition, language translation, IoT traffic prediction.
LSTM	Long-range sequence modeling, time-series prediction	Special type of RNN with memory cells to combat vanishing gradients	Overcomes RNN limitations in handling long-term dependencies	Computationally expensive and slower to train	NLP, time-series analysis, IDS for IoT.
GAN	Data generation, anomaly detection	Composed of two networks: generator (creates data) and discriminator (evaluates)	Powerful for generating synthetic data, unsupervised learning	Training instability, mode collapse, difficult to tune	Image synthesis, anomaly detection, IDS in IoT, synthetic data generation.

and ensemble detector-based IDS approaches that have been proposed. Additionally, a dedicated subsection titled ‘‘Discussion on Limitations of DL Models’’ highlights critical challenges in existing DL-based IDS methodologies, including issues such as computational complexity, mode collapse, data heterogeneity, and interpretability. This critical discussion aims to set the stage for exploring innovative solutions to address these limitations. Lastly, Tables 3 and 4 summarize the related works.

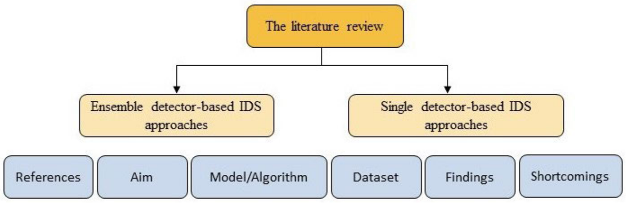

FIGURE 16. Taxonomy for related work - existing work.

To evaluate the performance of single DL models and ensemble models in IDS for IoT environments, we propose the following framework and criteria for comparison:

- Model Selection and Preparation: Single DL Models: Select representative single DL models such as CNNs, RNNs, and GANs. Ensemble Models: Choose various ensemble methods, including bagging (e.g., Random Forests), boosting (e.g., XGBoost), and hybrid ensembles that combine different base learners.
- Datasets: Utilize benchmark datasets relevant to IoT botnet detection, such as the N-BaIoT dataset, Bot-IoT

dataset, and CICIDS2017 dataset. These datasets should include a mix of normal and malicious traffic to ensure a comprehensive evaluation.

- Evaluation Metrics: Accuracy: Measure the overall correctness of the model in classifying benign and malicious traffic. Precision, Recall, and F-Score: Evaluate the model's performance in detecting true positives and minimizing false positives. Area Under the Receiver Operating Characteristic Curve (AUC-ROC): Assess the model's ability to discriminate between classes. Computation Time: Measure the training and inference times to evaluate the efficiency of the models.
- Experimental Setup: Training and Validation: Use cross-validation techniques, such as k-fold cross-validation, to ensure the robustness and generalizability of the results. Hyperparameter Tuning: Optimize hyperparameters for both single DL models and ensemble models to achieve the best possible performance.
- Comparative Analysis: Performance Comparison: Compare the single DL models and ensemble models based on the evaluation metrics. Highlight the strengths and weaknesses of each approach. Scalability and Resource Utilization: Assess the models' scalability and resource utilization, particularly in resource-constrained IoT environments. Robustness and Generalization: Evaluate the robustness of the models against various attack scenarios and their ability to generalize to new, unseen data.
- Interpretation and Insights: Feature Importance: Analyze the importance of different features in the detection process to understand the decision-making of the models. Model Interpretability: Discuss the interpretability of the models, emphasizing techniques like SHAP, LIME, and attention mechanisms for better transparency.

A. SINGLE DETECTOR-BASED IDS

A single detector-based IDS entails using a single detection method to detect botnets in IoT. This subsection will explore the strengths and weaknesses of various single detector-based IDS approaches, their detection capabilities, and limitations. The rise of IoT botnets, which exploit vulnerabilities in IoT devices, poses a significant threat, resulting in massive traffic to attack target nodes. The increased use of IoT devices in the smart health area has also increased the risk. Although researchers have proposed several botnet detection systems, applying them to resource-constrained IoT devices is challenging, and early detection of botnets is difficult due to slight traffic differences. However, IoT botnets produce recognizable power consumption patterns, which can be used to identify malicious behavior in IoT devices. A DL model based on CNN architecture has been proposed to tackle this issue, consisting of a data processing module and an 8-layer CNN [54]. The power consumption data collected is segmented and normalized to enhance the accuracy of the CNN model. The 8-layer CNN is responsible for classifying the processed data into four categories, specifically focusing on the botnet class. Various tests were conducted to evaluate the model's performance, including self-evaluation, cross-device evaluation, leave-one-device-out, and leave-one-botnet-out tests. The tests were performed on three commonly used IoT devices: Security Cameras, routers, and Voice Assistants. The self-tests achieved an impressive classification accuracy of up to 96.5%, while the cross-evaluation tests achieved an accuracy of approximately 90%. The leave-one-out tests also demonstrated a detection accuracy of over 90% for botnets. The article proposed by [4] addresses the challenges associated with implementing an IoT smart environment, highlighting the importance of protecting it from various attacks, including IoT Botnet attacks. To effectively detect these attacks, the paper proposes a detection approach that uses the interpolation reasoning method. The approach was evaluated using an open-source benchmark dataset of IoT Botnet attacks and achieved a high detection rate of 96.4%. The method is considered a competitive alternative to other approaches and effectively reduces false positives while being able to detect IoT Botnet attacks even with a limited rule base. As aforementioned, botnets are now a significant concern in cybersecurity, and conventional detection techniques like reverse engineering are ineffective in identifying botnets that use covert technologies such as fast flux. Available detection methods for fast-flux botnets are active or passive but have limitations. Reference [5] presents a novel approach for detecting fast-flux botnets using a combination of CNNs and RNN models to address these issues. Their method leverages the spatiotemporal features of network traffic to identify fast-flux botnets in both spatial and temporal dimensions. To evaluate the effectiveness of their approach, the researchers conducted experiments using the CTU-13 and ISOT public datasets. The results demonstrate that their proposed method achieves impressive performance metrics. Specifically, it achieves an accuracy rate of 98.3%, a recall rate of 96.7%, and an accuracy of 97.5%. It is important to note that the accuracy rate (98.3%) refers to the overall performance of the proposed method across all evaluation scenarios, while accuracy (97.5%) pertains to a specific aspect of the evaluation, such as performance on a particular dataset or configuration. These results surpass the performance of existing methods for fast-flux botnet detection. Reference [6] examine the susceptibility of IoT applications in smart cities to Advanced Persistent Threats (APT) posed by botnets. The paper proposes a botnet detection system that employs a two-level DL framework to address this concern. This framework focuses on semantically differentiating between botnet activities and legitimate behaviors at the application layer of the Domain Name System (DNS) services. The first level of the proposed framework utilizes Siamese networks to estimate similarity measures of DNS queries. These measures are then compared against a predefined threshold to identify potential botnet activities. The second level of the framework employs a DL-based Domain Generation Algorithm (DGA) to classify domain names as normal or abnormal. This approach utilizes DL architectures and is designed to be scalable on a commodity hardware server. The proposed framework was evaluated using two datasetsand compared with recent DL models, the NSRL Dataset, which provides reliable and verified data for cybersecurity systems, and the AmritaDGA Dataset, which specializes in analyzing DGA behavior for botnet detection. The experimental results show significant improvements in F1-score, detection speed, and false alarm rate, demonstrating the system's effectiveness, although specific numerical values for these metrics were not provided. However, the study identifies limitations, such as the lack of integration of contextual data like HTTP headers and passive DNS, which could enhance accuracy. Future work should focus on incorporating these elements and exploring DGA malware families to improve the robustness and performance of the detection system in real-world scenarios. In addition, [83] address the serious threat posed by botnets to Internet-connected computers, particularly the use of DGAs by advanced bots to generate random domain names for malware communication with Command and Control (C&C) servers. In response to this challenge, the paper introduces a Two-Stream network-based DL framework named TS-ASRCaps, which incorporates multimodal information to capture the characteristics of DGAs. The proposed framework leverages an Attention Sliced Recurrent Neural Network (ATTSRNN) to extract underlying semantics from the data and a Capsule Network (CapsNet) with dynamic routing to model high-level visual information. By combining these two components, TS-ASRCaps effectively analyzes and understands the complex patterns and structures inherent in DGAs. The multimodal approach of TS-ASRCaps, which combines semantic and visual information, enhances the detection and characterization of DGAs. The framework was evaluated using four benchmark datasets: OSINT, Lab360, AR, and XJU. These datasets provide comprehensive insights into DGA behavior, enabling robust evaluation of classification models. The experimental results, as detailed in Tables 3, 4, 5, and 6 of the study, demonstrate significant improvements in F1-score (ranging from 0.08% to 0.82%) and classification accuracy compared to six state-of-the-art models, including Endgame, CMU, NYU, Invincea, MIT, and LSTM-MI. These results validate the superior performance of TS-ASRCaps in both binary and multiclass domain name classification tasks. Additionally, the combination of semantic and visual information contributed to its enhanced capability to detect botnet-related activities and distinguish them from legitimate traffic. Reference [84] present a new IDS based on reinforcement learning to identify infected hosts within P2P botnets, including new bots with previously unknown behaviors and payloads. The proposed IDS includes a traffic reduction method to manage the high volume of network traffic and an early detection mechanism during the propagation phase, specifically during the peer discovery and secondary update stages. The researchers minimize false positives by adaptively setting a set of features representing the host traffic to distinguish between P2P bot-infected hosts and legitimate network hosts. In summary, the proposed system effectively detects and identifies P2P botnets while minimizing false positives. Also, [85] examines the severe threat botnets pose to Internet security and highlights the vulnerability of DL-based detection models to adversarial attacks. Recent research in adversarial DL has revealed that attackers can exploit vulnerabilities in detection models by crafting specific samples that contain nearly imperceptible input perturbations. The research introduces a Deep Reinforcement Learning (DRL) framework to address this issue and generate adversarial traffic flows capable of deceiving detection models [85]. The proposed framework automates adding perturbations to samples by leveraging a reinforcement learning agent. This agent continuously updates the adversarial samples by incorporating feedback from the target model and employing a sequence of actions. These actions alter the temporal and spatial characteristics of the traffic flows while ensuring their original functionality and executability. Experimental results demonstrate significant improvements in the evasion rates of adversarial botnet flows when employing the proposed DRL framework. By leveraging reinforcement learning and incorporating perturbations into traffic flows, the framework effectively enhances the ability to deceive detection models. It highlights the importance of developing robust defenses against adversarial attacks in the context of botnet detection.

Building on the findings of previous studies, Single DL models, while powerful, often face significant limitations when applied to IDS in IoT environments. These limitations include generalization issues, lack of diversity in detecting various attack types, and vulnerability to adversarial attacks. For instance,in, a CNN-based DL model achieved high accuracy in self-tests but faced generalization issues during cross-evaluation tests, indicating a need for improved adaptability to diverse data. Moreover, Niu et al. in [5] combined CNN and RNN models for fast-flux botnet detection and achieved impressive performance metrics. However, they noted adaptability issues and a lack of diversity in detecting evolving attack patterns, highlighting the need for more robust solutions. Additionally, Owen et al. in [2] discussed the vulnerability of DL-based detection models to adversarial attacks, where attackers can craft specific samples to deceive the models. This vulnerability further emphasizes the limitations of relying solely on single DL models. Ensemble learning models address these limitations by combining multiple models to leverage their individual strengths, resulting in more robust and accurate detection systems. For example, the BoostedEnsML model employs boosted machine learning classifiers, achieving superior accuracy, precision, recall, and F-score compared to existing ensemble models by using stacking and majority voting techniques. Another study introduced DeL-IoT, a deep ensemble learning framework that combines deep and stacked autoencoders with ensemble learning techniques, significantly improving detection performance even when dealing with imbalanced datasets. These examples demonstrate that ensemble learning models can effectively address the limitations of single DL models, providing enhanced robustness, adaptability, and accuracy in detecting diverse and sophisticated attacks in IoT environments.

Table 3 summarizes the related works for single detector-based IDS.

B. ENSEMBLE DETECTOR-BASED IDS

This subsection scrutinizes the ensemble detector-based IDS, which combines multiple detection methods to improve the detection rate and reduce false alarms. This subsection will explore the various ensemble-based IDS techniques, their effectiveness, and their limitations. Reference [86] discusses the cybersecurity challenges posed by the IoT, which is vulnerable to cyberattacks and can be exploited by botnets to launch devastating DDoS attacks against Internet services. Botnets are a significant security concern for the IoT because they can infect and control private network devices without the owners' knowledge, leading to various malicious activities. Despite the potential of ML technology in detecting botnets, previous studies have been either inaccurate or limited to specific types of botnets or devices. In this study, the authors propose an ensemble learning model that combines supervised, unsupervised, and regression learning methods to enhance botnet detection accuracy on the IoT while minimizing the required features. After conducting various experiments with different combinations of ML algorithms, the proposed model achieves high accuracy in detecting botnets on the IoT with just 20 features. Further, [87] address the security threats associated with the widespread adoption of IoT systems enabled by recent advancements in wireless communication. The authors emphasize the need for reliable IDS to detect cyberattacks and network intrusions in IoT environments. While ML algorithms have demonstrated potential in mitigating attacks on IoT devices, intruders' dynamics in IoT networks necessitate improved IDS models with higher detection rates and lower computational resource requirements. Ensemble methods have been proposed leveraging various ML classifiers such as decision trees and random forests. The authors present BoostedEnsML, an efficient IDS model that employs boosted ML classifiers to detect cyberattacks and network intrusions. They train six different ML classifiers, use the stacking method and majority voting approach to obtain an ensemble, and evaluate and test the IDS model on two datasets containing high-profile attacks. Data balancing with SMOTE and ADASYN techniques is performed, and a stratified K-fold is used to split the data into training, validation, and testing sets. Based on the best two models, BoostedEnsML is constructed using LightGBM and XGBoost. Experimental results demonstrate that BoostedEnsML surpasses existing ensemble models in accuracy, precision, recall, F-score, and area under the curve (AUC).

Also, [88] discuss the potential of attackers exploiting vulnerabilities in application protocols such as DNS, HTTP, and MQTT, leading to security breaches and data leakage in IoT services. The paper presents an ensemble intrusion detection technique that targets botnet attacks on these protocols to counter such threats. The technique described in the study involves generating new statistical flow features by examining the inherent properties of protocols. These novel features, including mean, variance, skewness, and kurtosis, were normalized and scaled, and features with zero variance were removed to enhance computational efficiency and classification accuracy. These features were utilized in an ensemble learning method called AdaBoost, which combines three machine learning techniques: Decision Tree, Naive Bayes, and Artificial Neural Network. The ensemble approach aims to accurately detect malicious events in network traffic while mitigating class imbalance by assigning higher weights to misclassified samples. The UNSW-NB15 and NIMS botnet datasets were employed, simulating data from IoT sensors. The selection of DNS, HTTP, and MQTT protocols was justified as these are frequently exploited in IoT-based attacks, making them high-priority targets. The data was divided into training and testing subsets to evaluate the method's performance. The experimental results demonstrate that the proposed features exhibit distinct characteristics related to normal and malicious activities, as indicated by correntropy and correlation coefficients. Moreover, the ensemble technique introduced in the study outperforms each classification technique employed in the framework and three other advanced techniques regarding detection and false positive rates. This suggests that combining multiple machine learning algorithms in the ensemble approach contributes to superior performance in identifying malicious events.

Reference [89] explore the use of honeypots in various computer security defense systems, which have proven effective in attracting botnet attacks and revealing the membership and behavior of attackers. However, botnet creators must find ways to avoid these honeypot traps. The article employs machine learning techniques to aid in detecting and preventing botnet attacks. The Ensemble Classifier Algorithm with Stacking Process (ECASP) is proposed to select optimal features that will serve as input for machine learning classifiers to determine botnet detection performance. The proposed method achieves an accuracy rate of 94.08%, a sensitivity rate of 86.5%, a specificity rate of 85.68%, and an F-measure of 78.24%. In addition, [90] study the security challenges associated with IoT devices, including their inherent vulnerability due to insecure design, implementation, and configuration. These challenges are compounded by the increasingly sophisticated tactics employed by attackers and the heterogeneity of IoT data. To tackle the challenges mentioned above, the authors present DeL-IoT, a deep ensemble learning framework designed for anomaly detection and prediction in IoT systems using SDN. DeL-IoT comprises three main modules: anomaly detection,

TABLE 3. Summary of related works- single-based detector.

Ref	Aim	Model / Algorithm	Dataset	Findings	Shortcomings
[54]	CNN-based deep learning model was suggested, comprising a data processing module and an 8-layer CNN.	CNN	Collected dataset (real dataset)	The self-tests achieved classification accuracy of up to 96.5%, while the cross-evaluation tests scored approximately 90% accuracy. The leave-one-out tests also yielded greater than 90% accuracy in botnet detection.	Generalization issue. Performance improvement is required.
[4]	It proposes a detection approach that uses the interpolation reasoning method.	Interpolation reasoning method	Bot-IoT dataset	It achieves a detection rate of 96.4%.	Lack of adaptability to dynamic scenarios.
[5]	It proposes a fast-flux botnet detection method that utilizes spatiotemporal features of network traffic and combines CNN and RNN models.	CNN and RNN	CTU-13 and ISOT	Accuracy rate of 98.3%, a recall rate of 96.7%, and overall accuracy of 97.5%.	Adaptability issues. Lack of diversity. Performance improvement is required.
[6]	It proposes a botnet detection system based on a two-level deep learning framework that semantically discriminates botnets and legitimate behaviors at the application layer of DNS services.	DGA	NSRL & AmritaDGA	The experimental results show noticeable improvements in terms of F1-score, speed of detection, and false alarm rate	Requires contextual data like HTTP headers, passive DNS. Gap in analyzing DGA malware families.
[83]	It presents a Two-Stream network-based deep learning framework called TS-ASRCaps that incorporates multimodal information to reflect the characteristics of DGAs.	DGA, ATTSRNN, and CapsNet	OSINT, Lab360, AR and XJU	The result demonstrate that the multimodel-based model surpasses the other models in terms of performance; F1-score improvement: 0.08%-0.82%.	Performance improvement is required. Stability issues.
[84]	It presents a new detection system based on reinforcement learning that aims to identify infected hosts within P2P botnets	Classification and Regression Tree (CART), NN and reinforcement learning	ISOT and ISCX	The experimental results show noticeable performance.	The size and dimensionality of the dataset, as the number of the packets that require scanning is significant
[85]	It proposes a DRL framework that generates adversarial traffic flows to deceive the detection model by automatically adding perturbations to samples.	DRL	IOST 2010	The results show improvements of evasions and mutations.	Performance improvement is required.

intelligent flow management, and device status forecasting. The framework leverages deep and stacked autoencoders to extract informative features, which are then utilized to build an ensemble learning model. By employing this approach, DeL-IoT can efficiently detect anomalies, dynamically manage flows, and provide short- and long-term forecasts of device status to enable proactive actions. The proposed framework is thoroughly tested and validated using testbed and benchmark datasets. Experimental results demonstrate that the deep feature extraction combined with the deep ensemble learning model outperforms a single model by approximately 3%, even when dealing with imbalanced datasets as small as 1%. DeL-IoT presents an effective IoT anomaly detection and prediction solution, offering improved performance through deep feature extraction and ensemble learning techniques. The framework's capabilities are demonstrated through extensive testing, confirming its potential for enhancing IoT system security and management. Reference [91] explore the benefits of integrating fog computing with IoT to facilitate swift detection of attacks, as the distance between IoT devices and fog devices is relatively shorter than between IoT devices and the cloud. However, due to resource constraints such as processing power and memory, fog devices may not be capable of detecting attacks in real-time using machine learning. The paper introduces an approach that distributes the machine-learning model selection task to the cloud and a real-time prediction task to fog nodes to overcome this obstacle. The approach constructs an ensemble machine-learning model in the cloud based on historical data and employs fog nodes to detect real-time attacks. The efficacy of the proposed approach is tested on the NSL-KDD dataset, and the experimental results demonstrate its efficiency in terms of various performance measures, including execution time, precision, recall, accuracy, and ROC curve. Reference [92] argued that smart cities' connectivity and intelligence features allow connected vehicles to collaborate and perform complex tasks that they cannot achieve individually. However, this connectivity also poses cybersecurity risks to connected vehicles, as cybercriminals use various techniques such as botnets, phishing, zero-days, and rootkits to disrupt vehicle communication. Botnets are a significant threat due to their ability to launch DDoS attacks using compromised devices. Therefore, early detection of botnet attacks is crucial for cybersecurity analysts. However, current research lacks precision in identifying botnet attacks in their early stages. In their study, they put forth a novel approach that utilizes machine learning algorithms to accurately detect botnet attacks early on by analyzing common network traffic patterns and temporal characteristics. The approach explores the effectiveness of decision trees, probabilistic neural networks, sequential minimal optimization, and Adaboost classifiers while examining the significance of temporal features in botnet detection. By employing this approach, the researchers achieve a commendable true positive rate, indicating the approach's ability to identify botnet attacks accurately. The results obtained through their experiments showcase the approach's efficiency, surpassing the performance of existing studies in the field. Reference [93] focus on the seriousness of cyber threats posed by the growth of botnets in the past decade. Due to their complex attack behaviors and communication patterns, the detection of botnets is challenging, and researchers have used machine learning techniques to improve detection rates. This paper proposes an ensemble classification framework incorporating noise filtering to enhance detection performance. Experimental results demonstrate that this framework outperforms other ensemble classification models in terms of accuracy and false alarm rate reduction.

Due to the limited computing, storage, and communication capabilities of endpoint devices in IoT infrastructures, they are vulnerable to various cyber-attacks, including Darknet and blackholes attacks [94]. Such attacks are relatively new and have targeted numerous IoT communication services. To address this issue, [95] developed, investigated, and assessed the performance of machine-learning-based Darknet Traffic Detection Systems (DTDS) in IoT networks. They employed six supervised machine-learning techniques and evaluated the implemented DTDS models using the CIC-Darknet-2020 dataset, covering four distinct classes of IoT communication traffic. The analysis revealed that the bagging ensemble techniques yielded better accuracy and lower error rates than supervised learning techniques. The proposed model achieves high classification accuracy with a low inferencing overhead of 9.09 seconds. It also compared their BAG-DT-DTDS with other existing DTDS models and demonstrated that its best results improved by 1.9% over the former state-of-the-art models.

Botnet attacks are a significant threat to IoT systems as they can manipulate devices to carry out malicious activities on a large scale. Various IDSs based on ML and DL have been proposed to detect such attacks. However, optimizing IDSs for IoT networks is challenging due to limited battery power and constrained resources. To tackle this concern, [96] present a novel approach that enhances the detection of IoT botnet attacks through aggregated mutual information-based feature selection using machine learning techniques. The proposed method utilizes the N-BaIoT benchmark dataset, which comprises genuine traffic data from nine commercial IoT devices, to train and evaluate the botnet attack detection system. Within the feature selection stage, Mutual Information, Principal Component Analysis, and ANOVA f-test are employed at a granular level to identify the most relevant features. For the classification step, the approach employs various individual and ensemble classifiers such as Random Forest, XGBoost, Gaussian Naïve Bayes, KNN, Logistic Regression, and SVM. By integrating these classifiers, the system aims to improve the accuracy and robustness of the botnet attack detection mechanism. Through their experiments and evaluations, [96] demonstrate the effectiveness of their approach in detecting IoT botnet attacks. Utilizing feature selection techniques and diverse classifiers contributes to achieving superior performance compared to existing methods. Reference [97] introduce a novel host-based IDS (HIDS) that leverages the C5 and the One-Class SVM classifier to achieve high detection accuracy and low false alarm rates for detecting well-known intrusions and zero-day attacks. The HIDS combines the strengths of signature-based IDS (SIDS) and anomaly-based IDS (AIDS). To evaluate the effectiveness of the HIDS, the researchers employed the Bot-IoT dataset, consisting of both normal IoT network traffic and different attack types. The experimental results demonstrate that the proposed HIDS outperforms both SIDS and AIDS techniques, achieving a higher detection rate and lower false positive rate. To further explore the potential of ensemble learning methods in IDS development, [55] investigated the application of automated machine learning (AutoML) to enhance the performance of ensemble-based techniques. Their study focused on comparing various machine learning algorithms, including Random Forest (RF), Naïve Bayes (NB), Multilayer Perceptron (MLP), and Sequential Minimal Optimization (SMO), using AutoML tools like Auto-WEKA and RapidMiner. By leveraging the NSL-KDD dataset, they demonstrated that RF, when optimized with Auto-WEKA, achieved a remarkable accuracy of 99.98%, significantly outperforming other models. The findings highlight the advantages of ensemble learning in achieving superior accuracy and reducing manual hyperparameter tuning. However, the study noted certain limitations, including the dependency on dataset characteristics and challenges in scaling AutoML solutions to more complex and dynamic IoT environments. This underscores the need for further exploration of ensemble learning techniques tailored specifically for real-world IoT applications.

Reference [98] introduced a novel deep learning approach named DeBot for detecting bots in industrial network environments. DeBot leverages a unique Cascade Forward Back Propagation Neural Network (CFBPNN) that employs a Correlation-based FS method to identify a critical subset of features. It further utilizes a time series-based Nonlinear Auto-regressive Network with eXogenous inputs (NARX) to assess and predict the influential factors on the outcome variable, understanding the behavioral trends. This model is pioneering in integrating optimal feature selection with a cascading deep learning framework for botnet detection in Industrial IoTs. They evaluated the proposed on five renowned bot datasets and comparing the performance of CFBPNN against other neural network models. The findings reveal that DeBot's CFBPNN achieves highest accuracy across all datasets with a subset evaluation, alongside optimal F1-scores and precision.

Besides, [99] focus on creating an ML-based IDS for IoT applications, called Ens-IDS. They utilized the Bot-IoT dataset to enhance attack detection in IoT networks, combining real and simulated IoT network traffics with various attack types. They generated two databases, with the second one being reduced in size, and addressed the issue of imbalance in the third database. The implementation involved applying five ML algorithms— DTs, ensemble bag, K-nearest neighbor, linear discriminant, and support vector machine—which achieved high performance scores. This study assesses classifier differences using key metrics such as accuracy, error rate, recall, specificity, precision, and f-measure. Research on IoT using the Bot-IoT dataset is not common in existing literature. The contribution is significant, offering a novel AI system based on ML to safeguard IoT networks and detect attacks, particularly DoS attacks. The Bot-IoT dataset was pivotal in evaluating the proposed approach, and the results demonstrate a notable enhancement in detection capabilities compared to current methods.

As steted by [100], many ML and DL models struggle with misclassification of malicious traffic, often due to poor feature selection. A key unresolved challenge is identifying effective features for precise detection of malevolent traffic in IoT networks. To tackle this issue, they introduced a new framework. Initially, they proposed a novel feature selection metric called CorrAUC, followed by the creation of a CorrAUC-based feature selection algorithm. This algorithm employs a wrapper technique for accurate feature filtering, selecting the most effective features for the chosen ML algorithm, using the Area Under the Curve (AUC) metric. Additionally, they utilized an integrated TOPSIS and Shannon entropy method within a bijective soft set framework to validate the chosen features for identifying malicious traffic in IoT networks. The approach is tested using the Bot-IoT dataset and four distinct ML algorithms. The results from the experiments indicate that the proposed method is effective, consistently achieving over high accuracy on average. Extending the analysis to feature selection methods, Current feature selection methods often rely on simple or multi-objective functions that may not fully capture the complexity of IoT network data. One such study [96] an aggregated mutual information-based feature selection approach using machine learning methods to improve the detection of IoT botnet attacks. They employed Mutual Information, Principal Component Analysis, and ANOVA f-test during the feature selection stage and various individual and ensemble classifiers for classification. Despite achieving superior performance compared to existing methods, the study highlights that using a single objective function can only optimize one criterion at a time, which may not reflect the complexity of the problem. Another study [86] discusses the use of ensemble learning techniques for detecting botnets in IoT. The proposed ensemble learning model combines supervised, unsupervised, and regression learning methods to enhance detection accuracy while minimizing the number of features. However, the study points out mode collapse issues, lack of diversity, and the need for performance improvement, emphasizing that current feature selection models might not effectively detect evolving botnets' sophisticated behaviors. These examples illustrate the limitations of current feature selection methods in handling the dynamic and complex nature of IoT environments, reinforcing the need for more advanced and comprehensive approaches.

Table 4 summarizes the related works for Ensemble-based detector.

TABLE 4. Summary of related works- ensemble-based detector.

Ref	Aim	Model / Algorithm	Dataset	Findings	Shortcomings
[86]	It proposes an ensemble learning model that combines supervised, unsupervised, and regression learning methods to enhance botnet detection accuracy on the IoT while minimizing the number of features.	ANN and DT	The dataset gathered data from three sources which are:Kaggle,Github, and Cert	This study Achieves 98% accuracy with 20 features selected; significant improvement in detecting botnets on the IoT.	Mode collapse issues. Lack of diversity. Performance improvement is required.
[87]	It presents BoostedEnsML, IDS model that employs boosted ML classifiers to detect cyberattacks and network intrusions	Boosted ML classifiers (LightGBM and XGBoost)	CIC0IDS2017 & CSE-CICIDS2018	Experimental results demonstrate that BoostedEnsML surpasses existing ensemble models with a detection accuracy of 97.3%, precision of 96.8%, and recall of 95.5%.	Lack of significant features selected to detect botnet efficiently.
[88]	It presents an ensemble IDS that targets botnet attacks on these protocols.	Decision Tree, Naive Bayes, and Artificial Neural Network	UNSW-NB15 & NIMS	The experimental findings indicate that the proposed features possess potential normal and malicious. Accuracy of 93.2%, improved detection rate, and efficient attack classification.	The proposed IDS result in significant overhead which degrads its performance.
[89]	It employs machine learning techniques to aid in detection and prevention.	ML & ECASP	Cyber Clean Center (CCC) dataset, which is publicly available, is used containing C08, C09, C10, and C13 datasets	The proposed method achieves an accuracy rate of 94.08%, sensitivity rate of 86.5%, specificity rate of 85.68%, and F-measure of 78.24%.	Performance improvement is required.

C. DISCUSSION ON LIMITATIONS OF DL MODELS

While DL models, such as CNNs, RNNs, and GANs, have demonstrated significant promise in IDS for IoT botnet detection, they are not without limitations. These limitations can adversely affect the effectiveness and reliability of IDS implementations.

- Mode Collapse in GANs: GANs are particularly prone to a phenomenon known as mode collapse, where the generator produces a limited diversity of outputs, neglecting significant portions of the data distribution. This issue arises because the generator learns to produce only a few types of outputs that can easily fool the

TABLE 4. *(Continued.)* Summary of related works- ensemble-based detector.

[90]	It proposes DeL-IoT, a deep ensemble learning framework for IoT anomaly detection and prediction using SDN.	Deep and stacked autoencoders	Testbed and benchmark datasets	Experimental results indicate that the deep feature extraction with a deep ensemble learning model achieves approximately 3% better performance compared to a single model, even when handling a 1% imbalanced dataset. The study reports a detection accuracy of 97.2% alongside a 2.5% reduction in false positives, demonstrating its effectiveness in improving classification performance.	Performance improvement is required.
[91]	It introduces an approach that distributes the machine learning model selection task to the cloud and real-time prediction task to fog nodes.	naïve Bayes, KNN, and decision trees, stacking, bagging	NSL-KDD	the experimental results demonstrate its efficiency in terms of various performance measures, including execution time, precision, recall, accuracy, and ROC curve. Specifically, the study achieved 95.6% precision, 94.3% recall, and demonstrated low execution time.	Performance improvement is required.
[92]	It proposes an approach that uses machine learning algorithms to accurately detect botnet attacks at an early stage using typical network traffic and temporal features.	Decision tree, probabilistic neural network, sequential minimal optimization, and Adaboost	The datasets collected from the PROMISE data repository	The study achieves a high true positive rate and an accuracy of 92.8%, demonstrating its efficiency and effectiveness compared to existing methods.	Performance improvement is required.

discriminator, leading to a lack of variety in the generated samples. Mode collapse reduces the effectiveness of GANs in IDS by limiting their ability to generate diverse adversarial samples needed for robust training. This can compromise the IDS's ability to generalize across different types of botnet attacks, potentially missing new or evolving threats. Addressing mode collapse often requires careful tuning of the GAN architecture, loss functions, and training processes, but these solutions can be complex and computationally intensive.

- Lack of Diversity: In addition to mode collapse, DL models often struggle with data heterogeneity.

TABLE 4. ***(Continued.)*** **Summary of related works- ensemble-based detector.**

[93]	It proposes an ensemble classification framework that incorporates noise filtering to enhance detection performance.	T algorithm is SVM or NB.	CTU-13 dataset	Experimental results demonstrate that this framework achieves 96.5% accuracy and reduces false alarms by 2.1%, outperforming other ensemble classification models.	Performance improvement is required.
[95]	It proposes machine-learning-based Darknet Traffic Detection Systems in IoT networks.	bagging decision tree ensembles (BAG-DT), AdaBoost decision tree ensembles (ADA-DT), RUSBoosted decision tree ensembles (RUS-DT), optimizable decision tree (O-DT), optimizable k-nearest neighbor (O-KNN), and optimizable discriminant (O-DSC).	CIC-Darknet-2020	A comparison of BAG-DT-DTDS with other existing models demonstrates its superiority, achieving a 98.9% precision rate, which is a 1.9% improvement over former state-of-the-art models.	Performance improvement is required.
[96]	It proposes an aggregated mutual information-based feature selection approach using machine learning methods to improve the detection of IoT botnet attacks.	Mutual Information, Principal Component Analysis, and ANOVA f-test, Random Forest, XGBoost, Gaussian Naïve Bayes, k-Nearest Neighbor, Logistic Regression, and Support Vector Machine.	N-BaIoT	The experimental results demonstrated the efficiency and effectiveness of the proposed approach, achieving 99.9% precision, recall, and F1-score for the selected aggregation methods.	The use of a single objective function can only optimize one criterion at a time, which may not reflect the complexity of the problem . Performance improvement is required.

TABLE 4. *(Continued.)* **Summary of related works- ensemble-based detector.**

[97]	It introduces a new HIDS that leverages both C5 classifier and One Class Support Vector Machine classifier.	C5 and SVM	Bot-IoT	The experimental results demonstrate that the proposed HIDS outperforms both SIDS and AIDS techniques, achieving 98.5% accuracy with a 1.8% false alarm rate.	The utilization of machine learning has led to restricted handling of traffic volume. Performance improvement is required.
[55]	To evaluate the performance of automated ensemble learning methods for IDS development.	Random Forest (RF) optimized with Auto-WEKA	The NSL-KDD datase	RF achieved 99.98% accuracy, demonstrating the effectiveness of AutoML in optimizing models.	Dependency on dataset characteristics; challenges in scaling AutoML to dynamic IoT scenarios.
[99]	To propose an ML-based IDS for IoT applications, called Ens-IDS.	DT, ensemble bag, KNN, linear discriminant, and SVM	Simulated IoT network traffics	Results demonstrate a notable enhancement in detection capabilities compared to current methods, achieving 95.4% precision and 93.6% recall.	Limited dataset utilization. Algorithmic bias and overfitting.
[98]	To introduce a novel deep learning approach named DeBot for detecting bots in industrial network environments	CFBPNN, CFS & NARX	NF-UNSW-NB15, NF-ToN-IoT, NF-BoT-IoT, NF-CSE-CIC-IDS2018, and ToN-IoT-Windows	The findings reveal that DeBot's CFBPNN achieves the highest accuracy across all datasets with subset evaluation, achieving 100% accuracy in select cases and an average F1-score of 0.99.	Optimizing only a single criterion at a time using a single objective function may not adequately capture the complexity of the issue, indicating a need for enhancement in performance.
[100]	To introduce a feature selection metric called CorrAUC, followed by the creation of a CorrAUC-based feature selection algorithm.	wrapper technique, TOPSIS and Shannon entropy	Bot-IoT	The proposed method is effective, consistently achieving over 95% accuracy in detecting malicious traffic.	Performance improvement is required.

IoT networks generate vast amounts of diverse data, including different types of traffic and various device behaviors. A single DL model might not effectively capture all these variations, leading to suboptimal performance. This lack of diversity in the training data can result in a higher rate of false positives or negatives, thereby affecting the IDS's reliability. For instance, if the training data does not include examples of certain types of botnet attacks, the IDS may fail to detect those attacks in real-world scenarios. Enhancing data diversity through comprehensive data collection and augmentation techniques is crucial but can be challenging to implement effectively.

- Training Instability: Training instability is another significant challenge, especially with GANs. The adversarial nature of GANs means that the training process is a dynamic game between the generator and the discriminator, which can lead to instability and make it difficult to achieve convergence. This instability affects the model's ability to learn effectively, leading to potential gaps in the IDS's coverage of botnet behaviors. Techniques such as spectral normalization, improved optimization algorithms, and stabilization strategies can help mitigate these issues but often require sophisticated adjustments and additional computational resources.
- Computational Complexity: DL models, particularly deep architectures like GANs and large CNNs, are computationally intensive. Training these models requires significant processing power, memory, and time. This complexity can be a barrier to deploying DL-based IDS on resource-constrained IoT devices, which often have limited computational capabilities. Solutions such as model compression, pruning, and the use of lightweight architectures can help, but these approaches might compromise the model's accuracy and detection capabilities.
- Interpretability and Explainability: DL models are often criticized for their lack of interpretability. Understanding the decision-making process of a DL-based IDS can be challenging, making it difficult for security analysts to trust and verify the model's predictions. This black-box nature hinders the ability to explain why certain traffic patterns are flagged as malicious, complicating the process of refining and improving the IDS. Developing methods for interpretable machine learning, such as attention mechanisms and explainable AI techniques, is essential to address this issue.
- Data Labeling and Quality: Effective training of DL models requires large amounts of labeled data, which can be difficult and expensive to obtain. The quality of the labeled data is also crucial, as noisy or incorrect labels can degrade the model's performance. In the context of IDS for IoT, collecting and accurately labeling diverse and representative datasets is a significant challenge. Approaches such as semi-supervised learning, transfer learning, and the use of synthetic data generation can help alleviate some of these challenges but come with their own set of complexities.
- Impact on IDS Performance: These limitations can significantly impact the performance of DL-based IDS. Mode collapse and lack of diversity reduce the robustness of the IDS, making it less effective against varied and sophisticated botnet attacks. Training instability can lead to incomplete learning, further compromising the detection capabilities of the IDS. Computational complexity can limit the applicability of these models in real-world IoT environments, where resources are constrained. The lack of interpretability and the challenges in data labeling further exacerbate these issues, making it difficult to achieve and maintain high-performance IDS.

Addressing these limitations requires a multifaceted approach, including the development of more advanced training techniques, the collection of diverse and high-quality datasets, the implementation of computationally efficient model architectures, and the incorporation of interpretability methods. By tackling these challenges, researchers and practitioners can enhance the effectiveness and reliability of DL-based IDS for securing IoT environments against botnet attacks.

IV. CRITICAL REVIEW

IoT's vulnerability to botnets is mainly due to the absence of standardized protocols and the intricacies of designing IoT sensors and protocols. Consequently, security experts encounter significant challenges when investigating and addressing security incidents within IoT networks. Detecting botnet attacks in these networks is paramount due to the potential harm they can inflict on the system. Although DL models have demonstrated remarkable performance in identifying botnet attacks in IoT, current studies have predominantly focused on employing a single DL model. However, this singular approach may not effectively adjust to changing network conditions and the evolving characteristics of botnets. Additionally, it may encounter difficulties handling diverse data and class imbalances, resulting in suboptimal trade-offs among multiple objectives, such as maximizing detection accuracy, minimizing false alarms, reducing computation time, and ensuring resilience against various attack scenarios. Addressing these limitations necessitates thoroughly exploring the potential benefits ensemble learning models can offer to detect botnets in IoT networks. However, the current literature lacks adequate research on applying ensemble learning models for botnet detection in IoT networks. Therefore, it is imperative to investigate and assess the advantages that ensemble learning models could bring to botnet detection and to conduct a comparative analysis of their performance against single DL models. Such research endeavors have the potential to significantly enhance the efficiency and robustness of botnet detection systems in IoT networks. The existing literature on DL-based IDS for botnet detection in IoT can be divided into two categories, as the summary of the related works in Table 3 and Table 4 illustrates: (i) single-based IDS [4], [5], [6]; and (ii) ensemble detector-based IDS [86], [87], [89]. Despite the abundance of IDS proposed in the literature, including those by [4] and [89], there is room for improvement in their detection accuracy for several reasons:

1) Reliance on features extracted based on simple or simple multi-objective functions for feature subset selection.
2) Dependency on a single feature selection model to choose the feature subset.
3) A single DL model instead of an ensemble might not effectively detect evolving botnets' sophisticated and complex behaviors.
4) Inability to detect adversarial attacks that can alter botnets' behaviors, potentially allowing evasive actions.

Although DL has proven its superiority over traditional methods, especially in handling large datasets, it still faces challenges such as mode collapse, lack of diversity, and training instability, which are commonly encountered in traditional GANs [101], [102], [103]. These issues can significantly compromise the performance of DL-based IDS.

Moreover, practical challenges such as computational efficiency and real-time detection remain critical in IoT environments. Studies such as [87, 89] emphasize the necessity of balancing computational overhead and detection accuracy, particularly in resource-constrained settings. Simpler models, such as Decision Trees and Random Forests, are advantageous for initial analysis due to their efficiency, but their limitations in detecting complex attack patterns highlight the need for complementary approaches. The integration of lightweight models for rapid detection with more robust DL models for detailed analysis offers a promising direction. These trade-offs and their implications for real-time detection will be explored in the following section.

A. REAL-TIME DETECTION CHALLENGES IN DL MODELS

Real-time detection is crucial for promptly identifying and mitigating botnet attacks in IoT environments. Despite the advancements in DL-based IDS, several challenges hinder the achievement of real-time detection, which necessitate a variety of approaches to address these limitations effectively. As we have outlined the key obstacles in real-time detection, we will now explore the various existing methods that have been developed to tackle these challenges. The following section will detail these approaches, demonstrating how they contribute towards overcoming the complexities of real-time detection in IoT environments.

- Computational Complexity: DL models, particularly deep architectures like CNNs and GANs, are computationally intensive. The high computational requirements for training and inference can lead to delays, making real-time detection challenging, especially in resource-constrained IoT devices. The complexity arises from the need to process large amounts of data through multiple layers of neural networks, each requiring significant computational resources. Implementing these complex models on devices with limited processing power, memory, and energy resources remains a significant challenge. Solutions such as edge computing, where data processing occurs closer to the source, can help mitigate some of these issues by distributing the computational load.
- Latency Issues: Latency is a critical factor in real-time detection. The time taken for data to be processed and analyzed by the DL models can result in delayed responses to attacks. This delay can be detrimental in a real-time scenario where immediate action is required to mitigate threats. Techniques such as model compression, pruning, and the use of lightweight architectures have been explored to reduce latency but often at the cost of model accuracy and effectiveness. For instance, pruning involves removing less important neurons or weights in the network to speed up computation, but it can lead to a loss of valuable information, reducing the model's ability to detect complex attack patterns.
- Online Learning and Adaptation: Real-time detection requires models that can continuously learn and adapt to new data. Traditional DL models are typically trained offline on static datasets and may not effectively adapt to evolving threats in real-time. Online learning techniques, which allow models to update and refine themselves as new data arrives, are essential for maintaining the efficacy of IDS in dynamic environments. However, implementing online learning in DL models is complex and can introduce additional computational overhead. This overhead comes from the need to continuously integrate new data, retrain models, and adjust parameters, all of which require significant computational resources and can slow down the detection process.
- Data Stream Management: Managing and processing data streams in real-time is another significant challenge. IoT networks generate massive amounts of data continuously, and the IDS must efficiently handle and analyze these data streams to detect anomalies promptly. Ensuring that the DL models can process high-throughput data streams without bottlenecks is critical for effective real-time detection. This involves not only fast data processing but also efficient data storage and retrieval mechanisms. Stream processing frameworks like Apache Kafka and Apache Flink are often used to manage real-time data, but integrating these with DL models can be challenging and requires careful tuning and optimization.
- Trade-off Between Accuracy and Speed: Achieving a balance between detection accuracy and processing speed is a persistent challenge. High accuracy often requires more complex models and extensive computations, which can slow down the detection process. Conversely, simpler models that offer faster processing

times may not provide the desired level of accuracy and robustness in detecting sophisticated attacks. This trade-off necessitates the exploration of hybrid approaches that can optimize both aspects. For example, a layered approach could be used where a lightweight model performs initial detection, and more complex models are employed for detailed analysis of suspected threats. This can help achieve a balance between speed and accuracy, ensuring that real-time requirements are met without compromising the detection capabilities.

1) EXISTING REAL-TIME DETECTION METHODS:

Several approaches have been proposed to address these challenges:

- Edge and Fog Computing: Fog computing reduces latency by processing data closer to the source rather than in a centralized cloud. In this approach, computational tasks are distributed between the cloud and fog nodes, with real-time prediction tasks handled by the latter to enhance detection speed. This reduces the amount of data that needs to be sent to the cloud for processing, thereby decreasing latency and improving response times.
- Ensemble Learning Methods: Ensemble learning methods, which combine multiple classifiers, have shown promise in improving detection accuracy and robustness. Techniques such as stacking, boosting, and bagging can be used to aggregate the strengths of different models, providing a more comprehensive detection system. However, these methods also face challenges related to computational efficiency and real-time applicability. Managing multiple models and combining their outputs in real-time can be resource-intensive and requires efficient parallel processing capabilities.
- Incremental Learning: Incremental learning methods allow DL models to update themselves with new data without the need for complete retraining. This approach helps maintain the model's relevance over time as new attack patterns emerge. However, ensuring that the model does not forget previously learned information (catastrophic forgetting) while integrating new data is a complex challenge that requires sophisticated algorithms and careful balancing of old and new data.
- Model Optimization Techniques: Techniques such as quantization, where the precision of the model parameters is reduced, and knowledge distillation, where a smaller model learns from a larger, more complex model, can help reduce the computational demands of DL models. These techniques aim to create more efficient models that can perform real-time detection without significant loss of accuracy.
- Stream Processing Frameworks: Integrating DL models with stream processing frameworks like Apache Kafka and Apache Flink can enhance real-time data handling capabilities. These frameworks provide robust tools for managing high-throughput data streams, ensuring that data is processed and analyzed in real-time. However, the integration process can be complex and requires careful tuning to ensure seamless operation.
- Hybrid Approaches: Hybrid approaches that combine traditional machine learning methods with DL models can leverage the strengths of both. For example, initial anomaly detection can be performed using lightweight traditional methods, followed by detailed analysis using DL models. This can help balance the trade-offs between speed and accuracy, ensuring that real-time detection requirements are met.

V. FUTURE DIRECTIONS

Despite the extensive research conducted on botnet detection in the IoT, the prior section has provided a thorough review of the existing techniques, shedding light on their respective merits and limitations. However, it is vital to recognize that several crucial considerations in future research endeavors must be addressed. These considerations include:

- Overcoming the shortage of labeled training data is a crucial obstacle in botnet detection, and future research can focus on exploring specialized data augmentation techniques designed specifically for IoT environments. Techniques like GANs or synthetic data generation can help augment the limited labeled data, resulting in more robust and accurate DL models. GANs are a powerful tool for data augmentation, particularly in IoT environments where labeled data is often scarce. GANs consist of two neural networks—a generator and a discriminator—that are trained simultaneously through adversarial processes. The generator creates synthetic data samples, while the discriminator evaluates their authenticity. This process helps generate high-quality, realistic data that can be used to augment the training datasets for DL models, improving their performance and robustness. GANs can generate synthetic data that closely mimics real-world IoT data. For example, in the context of network traffic, GANs can create additional traffic patterns that include both normal and malicious activities. This augmented dataset can be used to train DL models, enhancing their ability to detect diverse and previously unseen attack patterns. Since IoT datasets often suffer from class imbalance, where benign data significantly outweighs malicious data, thus GANs can help balance these datasets by generating more samples of the underrepresented class. This approach improves the model's ability to detect rare attack types and reduces the likelihood of bias towards the majority class. Also, GANs can also be used to generate adversarial examples that expose the weaknesses of DL models. By training models with these adversarial examples, researchers can improve the robustness of IDS against sophisticated attacks designed to evade detection. These applications illustrate the practical benefits of using GANs for data augmentation in IoT environments, providing a robust framework for enhancing DL-based IDS. By generating synthetic data, balancing datasets, and enhancing model robustness, GANs offer a promising approach to overcoming the challenges associated with limited and imbalanced IoT data.

- Leveraging transfer learning can improve the performance of botnet detection models in IoT. Researchers can pre-train DL models on more extensive and diverse datasets, such as general network traffic or related domains, and then fine-tune them on specific IoT datasets. This approach allows the transfer of learned representations and enhances detection performance. Transfer learning has shown great potential in enhancing the performance of DL models by leveraging pre-trained models on extensive datasets and fine-tuning them on specific tasks with limited data. In the context of IoT environments, transfer learning can help address the challenge of scarce labeled data by transferring knowledge from related domains. Current research in this area includes works such as Abdelhamid et al., 2024; Nandanwar and Katarya, 2024)], which investigates the use of transfer learning to improve the detection of botnet attacks in IoT networks. The study highlights how pre-trained models on general network traffic can be fine-tuned with specific IoT data to enhance detection accuracy. However, the research also points out limitations, such as the need for domain-specific adjustments and the potential for overfitting when transferring knowledge from significantly different domains. These examples demonstrate the potential benefits of transfer learning in IoT environments, as well as the current limitations that future research needs to address. By highlighting these studies, we provide a clearer understanding of the existing landscape and underscore the importance of further exploring transfer learning techniques to improve IDS performance in IoT networks.
- Real-time detection is essential in IoT environments to promptly identify and mitigate botnet attacks. Future research can explore online learning techniques that enable continuous learning and adaptation of IDSs as new data arrives. This approach enhances the system's ability to detect emerging botnet behaviors and adapt to evolving attack strategies in real-time.
- DL models often lack transparency and interpretability, making understanding the reasoning behind their decisions difficult. Future research should focus on developing techniques that provide explainable and interpretable botnet detection models. By incorporating domain knowledge and designing model architectures that offer insights into decision-making, stakeholders can better understand the detection process and build trust in the system. Specific techniques and frameworks for explainable AI (XAI) in the context of DL-based IDS for botnet detection can be used. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) help in understanding feature importance and how individual predictions are made. Visualization methods like activation maps, heatmaps, and saliency maps can highlight the regions of input data that strongly influence the model's decisions. Additionally, attention mechanisms in RNNs and Transformers, as well as autoencoders and generative models, enhance the interpretability by showing the model's focus and learned data distributions. Model-specific interpretability can also be achieved through rule extraction and surrogate models, where simpler models approximate the behavior of complex DL models to provide understandable decision logic. Frameworks such as IBM's AI Explainability 360 and Microsoft's InterpretML offer comprehensive tools for implementing various XAI techniques. These methods collectively aim to build trust in IDS by providing transparency in the decision-making processes of DL models, ultimately improving their reliability and effectiveness in detecting botnet attacks.
- Implementing complex DL models on resource-constrained IoT devices presents challenges due to their limited computational resources. Future research should investigate techniques to optimize DL models for such devices, including model compression, quantization, and lightweight architectures, for efficient and practical deployment of botnet detection systems on a wide range of IoT devices. Specific techniques such as pruning, quantization, and knowledge distillation. Pruning reduces the model's complexity by eliminating less significant weights and neurons, while quantization lowers the precision of model parameters, thereby decreasing memory footprint and computational demands. Knowledge distillation trains a smaller model to mimic the performance of a larger one, maintaining accuracy with fewer resources. Additionally, lightweight architectures like MobileNet and SqueezeNet, designed for efficiency, are well-suited for IoT devices with constrained resources. Furthermore, edge and fog computing paradigms offer practical solutions by distributing computational tasks. Edge computing processes data closer to the source, reducing latency and bandwidth requirements, while fog computing creates a hierarchical architecture that balances the computational load between cloud servers, fog nodes, and edge devices. Existing research, such as the DeL-IoT framework, leverages deep ensemble learning for efficient anomaly detection in IoT systems, and ensemble machine learning models distribute tasks between cloud and fog nodes for real-time detection. These approaches demonstrate the potential for improving IoT system security and management while addressing resource constraints.
- Hybrid Ensemble Approaches: Exploring hybrid ensemble methods that integrate multiple feature selection algorithms can yield more robust and dependable

outcomes. By leveraging the strengths of diverse techniques, such as filter, wrapper, and embedded methods, researchers have the potential to devise ensemble strategies that enhance the effectiveness of feature selection and elevate the overall performance of IDSs. Hybrid ensemble methods combine multiple learning algorithms to leverage their individual strengths, resulting in more robust and accurate detection systems. Several studies have demonstrated the effectiveness of such approaches in various domains. For instance, the BoostedEnsML model employs boosted machine learning classifiers, specifically LightGBM and XGBoost, to detect cyberattacks and network intrusions. This approach involves training multiple classifiers and combining their outputs using stacking and majority voting techniques. The ensemble model was evaluated on the CICIDS2017 and CSE-CICIDS2018 datasets, achieving superior accuracy, precision, recall, and F-score compared to existing ensemble models. Another example is the DeL-IoT framework, which utilizes deep ensemble learning for anomaly detection in IoT systems. This framework combines deep and stacked autoencoders with ensemble learning techniques to enhance performance, even when dealing with imbalanced datasets. The experimental results indicated a performance improvement of approximately 3% over single models. These examples highlight the potential benefits of hybrid ensemble methods in enhancing the robustness and accuracy of DL-based IDS, particularly in dynamic and resource-constrained IoT environments.

- Dynamic Feature Selection Ensembles: The investigation of dynamic feature selection techniques within ensemble frameworks can be highly advantageous. These methods can adaptively select pertinent features based on evolving data characteristics. By continuously monitoring the importance and relevance of features over time, these techniques enable efficient feature selection in dynamic IoT environments where network traffic patterns may change.
- Multi-Objective Ensemble Feature Selection: Future studies should consider integrating multiple objectives within ensemble feature selection approaches, which include optimizing for high detection accuracy and other evaluation metrics like computational efficiency, scalability, and interpretability. Utilizing multi-objective optimization algorithms aids in striking a balance between these objectives and guiding the selection of an optimal feature subset.
- Domain-Specific Feature Selection: Given IoT networks' varied domains and application scenarios, developing feature selection methods specific to each domain is crucial. Researchers should formulate domain-specific feature selection techniques considering different IoT domains' unique characteristics and requirements. This tailored approach will result in more accurate and customized feature selection strategies for specific IoT applications, including healthcare, smart cities, and industrial control systems.

VI. QUALITATIVE COMPARISON WITH RELATED REVIEWS:

This section compares this review paper with the existing reviews on botnet detection in the IoT using DL-based IDS to emphasize its distinctive contributions and novel insights. Numerous literature reviews have been conducted on the topic of botnet detection in the IoT [104], [105], [106], [107], [108], focusing on various aspects such as conventional detection methods, machine learning approaches, and anomaly detection techniques. While these reviews have offered valuable summaries of the existing research, this review paper sets itself apart by explicitly concentrating on applying DL techniques for botnet detection in the IoT, allowing deeper focus into the intricacies and nuances of DL-based approaches, resulting in a more comprehensive analysis of their strengths, limitations, and future research directions. Moreover, the proposed review paper distinguishes itself by not only discussing different DL models utilized for botnet detection, including CNNs, RNNs, and GANs but also by examining their effectiveness and applicability within the context of IoT environments and the coverage compared with existing surveys and reviews, as illustrated in Table 5. It highlights the unique challenges IoT networks pose, such as the dynamic nature of IoT devices, limited computational resources, and the heterogeneity of IoT data. By addressing these challenges, the proposed review offers valuable insights for researchers and practitioners aiming to develop robust and efficient DL-based IDSs for botnet detection in IoT. Furthermore, in addition to its specialized focus and comprehensive analysis, the proposed review paper provides a forward-looking perspective by identifying critical areas for future research. It emphasizes exploring hybrid ensemble methods, dynamic ensemble feature selection techniques, multi-objective optimization approaches, and domain-specific feature selection methods, which are critical for advancing the field of botnet detection in IoT and tackling the evolving challenges presented by sophisticated botnet attacks.

This section provides a detailed evaluation of the IoT vulnerabilities, deep learning methodologies, and comparative analyses of IoT-specific attacks presented in existing reviews to further enrich the discussion. Table 5 systematically highlights the distinctions between the proposed review and prior works. It emphasizes critical gaps, particularly in IoT vulnerabilities, deep learning methodologies, and recent studies covered. Notably, prior reviews [106], [107], [108] lack comprehensive coverage of deep learning frameworks and fail to holistically address IoT-specific attacks. For instance, while [107], [108] demonstrate comprehensive insights into recent studies, their partial coverage of IoT-specific challenges limits their applicability to dynamic IoT environments.

TABLE 5. Coverage comparison between the proposed review and state-of-the-art reviews.

Review Ref	IoT vulnerabilities	Deep learning	Taxonom of DL	Recent Studies Covered	IoT-Specific Attacks Covered
[106]	✓	X	✓	Partially	Partial
[107]	✓	X	✓	Comprehensive	Comprehensive
[108]	✓	X	✓	Comprehensive	Partial
[104]	X	X	X	Partially	Limited
[105]	✓	X	X	Partially	Comprehensive
Proposed review	✓	✓	✓	Comprehensive	Comprehensive

Conversely, the proposed review stands out by offering comprehensive coverage across all dimensions, including taxonomy of deep learning models, recent advancements, and IoT-specific attack scenarios. This integrated perspective provides a clearer understanding of the limitations in existing works and addresses key IoT-specific challenges.

Table 5 further illustrates how prior reviews often focus on isolated aspects of IoT security (e.g., [104] lacks IoT vulnerability considerations and deep learning techniques) or present only partial coverage of recent advancements [105], [106]. In contrast, the proposed review bridges these gaps by leveraging a multidimensional approach that incorporates a broader evaluation of IoT-specific challenges and novel solutions. The gaps highlighted in Table 5 further emphasize the need for advanced methods like hybrid ensemble techniques to overcome the challenges of IoT-specific attacks and enhance the robustness of DL-based IDSs.

Hybrid ensemble methods combine multiple learning algorithms to leverage their individual strengths, resulting in more robust and accurate detection systems. These approaches can effectively address limitations such as mode collapse, lack of diversity, and training instability in DL models. For instance, the BoostedEnsML model employs boosted machine learning classifiers, specifically LightGBM and XGBoost, to detect cyberattacks and network intrusions. By using stacking and majority voting techniques, this ensemble model outperforms existing ensemble models in accuracy, precision, recall, and F-score. Another notable example is the DeL-IoT framework, which leverages deep ensemble learning for anomaly detection and prediction in IoT systems. Combining deep and stacked autoencoders with ensemble learning techniques, DeL-IoT enhances performance even when dealing with imbalanced datasets. This approach significantly improves detection accuracy and robustness compared to single DL models, addressing challenges like data heterogeneity and class imbalance. These examples illustrate the potential of hybrid ensemble methods to enhance the effectiveness and reliability of DL-based IDS in dynamic IoT environments.

Table 5 not only illustrates the distinctions between the proposed review and prior works but also emphasizes how the comprehensive coverage of IoT vulnerabilities, deep learning methodologies, and IoT-specific attacks provides a broader and more actionable understanding of botnet detection in IoT environments. This systematic evaluation highlights the strengths of the proposed review in addressing key gaps and sets a strong foundation for future advancements in DL-based IDSs tailored to the unique challenges of IoT networks.

VII. CONCLUSION

This paper provides a comprehensive review of the utilization of DL-based IDSs for botnet detection in the IoT. The proliferation of IoT devices and the increasing sophistication of botnet attacks underscore the urgency for advanced detection techniques. With its ability to learn intricate patterns and representations from data, DL emerges as a promising solution to counter the evolving botnet threat in IoT environments effectively. The review highlights the successful applications of various DL techniques in botnet detection, including CNNs, RNNs, and GANs. These techniques enable extracting high-level features and identifying anomalous behavior patterns, enhancing IDS accuracy and efficiency. Several initiatives are currently underway to develop standardized datasets and evaluation metrics for IoT security and IDS. For instance, the Bot-IoT dataset is widely used in research for evaluating the performance of IDS in detecting various types of IoT-related cyberattacks. This dataset provides comprehensive labeled data, simulating different attack scenarios, which is crucial for training and testing IDS models. Another notable effort is the CICIDS2017 dataset, which includes diverse network traffic data to benchmark IDS performance across various attack vectors. Additionally, organizations such as the National Institute of Standards and Technology and the European Union Agency for Cybersecurity are actively working on establishing guidelines and best practices for cybersecurity, including the development of standardized metrics for evaluating IDS effectiveness. These efforts collectively aim to enhance the comparability and robustness of IDS research, fostering collaboration and innovation in the field.

REFERENCES

[1] C. Wei, G. Xie, and Z. Diao, "A lightweight deep learning framework for botnet detecting at the IoT edge," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103195, doi: 10.1016/j.cose.2023.103195.

[2] H. Owen, J. Zarrin, and S. M. Pour, "A survey on botnets, issues, threats, methods, detection and prevention," *J. Cybersecurity Privacy*, vol. 2, no. 1, pp. 74–88, Feb. 2022, doi: 10.3390/jcp2010006.

[3] E. Alomari, S. Manickam, B. B. Gupta, P. Singh, and M. Anbar, "Design, deployment and use of HTTP-based botnet (HBB) testbed," in *Proc. 16th Int. Conf. Adv. Commun. Technol.*, Feb. 2014, pp. 1265–1269, doi: 10.1109/ICACT.2014.6779162.

[4] M. Almseidin and M. Alkasassbeh, "An accurate detection approach for IoT botnet attacks using interpolation reasoning method," *Information*, vol. 13, no. 6, p. 300, Jun. 2022, doi: 10.3390/info13060300.

[5] W. Niu, T. Jiang, X. Zhang, J. Xie, J. Zhang, and Z. Zhao, "Fast-flux botnet detection method based on spatiotemporal feature of network traffic," *J. Electron. Inf.*, vol. 42, no. 8, pp. 1872–1880, Aug. 2020.

[6] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. 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 Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4436–4456, Jul. 2020, doi: 10.1109/TIA.2020.2971952.

[7] F. Wortmann and K. Flüchter, "Internet of Things: Technology and value added," *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 221–224, Mar. 2015, doi: 10.1007/s12599-015-0383-3.

[8] D. Lund, C. MacGillivray, V. Turner, and M. Morales, *Worldwide and Regional Internet of Things (IoT) 2014-2020 Forecast: A Virtuous Circle of Proven Value and Demand*, document IDC f248451, May 2014. [Online]. Available: www.idc.com

[9] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015, doi: 10.1016/j.bushor.2015.03.008.

[10] X. Huang, P. Craig, H. Lin, and Z. Yan, "SecIoT: A security framework for the Internet of Things," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3083–3094, May 2015, doi: 10.1002/sec.1259.

[11] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, and I. Traore, "A survey on IoT-enabled smart grids: Emerging, applications, challenges, and outlook," *Energies*, vol. 15, no. 19, p. 6984, Sep. 2022, doi: 10.3390/en15196984.

[12] R. Nicolescu, M. Huth, P. Radanliev, and D. De Roure, "Mapping the values of IoT," *J. Inf. Technol.*, vol. 33, no. 4, pp. 345–360, Dec. 2018, doi: 10.1057/s41265-018-0054-1.

[13] S. Bandyopadhyay, P. Balamuralidhar, and A. Pal, "Interoperation among IoT standards," *J. ICT Standardization*, vol. 1, no. 2, pp. 253–270, 2013, doi: 10.13052/jicts2245-800x.12a9.

[14] J. H. Nord, A. Koohang, and J. Paliszkiewicz, "The Internet of Things: Review and theoretical framework," *Expert Syst. Appl.*, vol. 133, pp. 97–108, Nov. 2019, doi: 10.1016/j.eswa.2019.05.014.

[15] N. Sami, T. Mufti, S. S. Sohail, J. Siddiqui, and D. Kumar, *Future Internet Things (IoT) From Cloud Perspective: Aspects, Appl. Challenges*. Cham, Switzerland: Springer, 2020, pp. 515–532, doi: 10.1007/978-3-030-37468-6_27.

[16] L. Huang, X. Yuan, J. Zhang, N. Zhang, J. Li, and L. Wang, "Research on Internet of Things technology and its application in building smart communities," *J. Phys., Conf. Ser.*, vol. 1550, no. 2, May 2020, Art. no. 022029, doi: 10.1088/1742-6596/1550/2/022029.

[17] E. Elbasani, P. Siriporn, and J. S. Choi, "A survey on RFID in Industry 4.0," in *Internet Things for Ind. 4.0: Design, Challenges Solutions*. Springer, Dec. 2020, pp. 1–16. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-32530-5_1

[18] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining Anal.*, vol. 4, no. 1, pp. 25–32, Mar. 2021, doi: 10.26599/BDMA.2020.9020018.

[19] D. Uckelmann, M. Harrison, and F. Michahelles, *Architecting Internet Things*. Cham, Switzerland: Springer, 2011.

[20] D. Jiang, "The construction of smart city information system based on the Internet of Things and cloud computing," *Comput. Commun.*, vol. 150, pp. 158–166, Jan. 2020, doi: 10.1016/j.comcom.2019.10.035.

[21] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

[22] R. Kohar, "Iot systems based on soa services: Methodologies, challenges and future directions," in *Proc. 4th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2020, doi: 10.1109/iccmc48092.2020.iccmc-000103.

[23] B. Costa, P. F. Pires, and F. C. Delicato, "Towards the adoption of OMG standards in the development of SOA-based IoT systems," *J. Syst. Softw.*, vol. 169, Nov. 2020, Art. no. 110720, doi: 10.1016/j.jss.2020.110720.

[24] S. K. Mishra and A. Sarkar, "Service-oriented architecture for Internet of Things: A semantic approach," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8765–8776, Nov. 2022, doi: 10.1016/j.jksuci.2021.09.024.

[25] N. Shahid and S. Aneja, "Internet of Things: Vision, application areas and research challenges," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 583–587, doi: 10.1109/I-SMAC.2017.8058246.

[26] A. E. Varjovi and S. Babaie, "Green Internet of Things (GIoT): Vision, applications and research challenges," *Sustain. Comput., Informat. Syst.*, vol. 28, Dec. 2020, Art. no. 100448, doi: 10.1016/j.suscom.2020.100448.

[27] S. Li, L. D. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2014.

[28] M. K. Hooshmand, M. D. Huchaiah, A. R. Alzighaibi, H. Hashim, E.-S. Atlam, and I. Gad, "Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI)," *Alexandria Eng. J.*, vol. 94, pp. 120–130, May 2024, doi: 10.1016/j.aej.2024.03.041.

[29] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, and S. Zhang, "Recommendations on the Internet of Things: Requirements, challenges, and directions," *IEEE Internet Comput.*, vol. 23, no. 3, pp. 46–54, May 2019, doi: 10.1109/MIC.2019.2909607.

[30] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, doi: 10.1016/j.comnet.2010.05.010.

[31] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. A. Bahashwan, I. H. Hasbullah, M. A. Aladaileh, and G. A. Mukhaini, "A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100741, doi: 10.1016/j.iot.2023.100741.

[32] K. L. Lueth, *The 10 Most Popular Internet of Things Applications Right Now*. Hamburg, Germany: IoT Analytics, 2015.

[33] J. Bartje. (2016). *The Top 10 IoT Application Areas–Based on Real IoT Projects*. [Online]. Available: https://iot-analytics.com/top-10-iot-project-application-areas-q3-2016/

[34] P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, "Internet of Things: Security and solutions survey," *Sensors*, vol. 22, no. 19, p. 7433, Sep. 2022, doi: 10.3390/s22197433.

[35] H. Tschofenig and E. Baccelli, "Cyberphysical security for the masses: A survey of the Internet protocol suite for Internet of Things security," *IEEE Secur. Privacy*, vol. 17, no. 5, pp. 47–57, Sep. 2019, doi: 10.1109/MSEC.2019.2923973.

[36] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of Things security: Requirements, challenges, and solutions," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100129, doi: 10.1016/j.iot.2019.100129.

[37] M. Elattar, A. Younes, I. Gad, and I. Elkabani, "Explainable AI model for PDFMal detection based on gradient boosting model," *Neural Comput. Appl.*, vol. 36, no. 34, pp. 21607–21622, Sep. 2024, doi: 10.1007/s00521-024-10314-y.

[38] S. D. A. Rihan, M. Anbar, and B. A. Alabsi, "Approach for detecting attacks on IoT networks based on ensemble feature selection and deep learning models," *Sensors*, vol. 23, no. 17, p. 7342, Aug. 2023, doi: 10.3390/s23177342.

[39] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, pp. 1–27, 2021.

[40] T. Micro, "Smart yet flawed: IoT device vulnerabilities explained," Secur. News, Trend Micro Inc., Irving, TX, USA, Tech. Rep., 2020. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/smart-yet-flawed-iot-device-vulnerabilities-explained

[41] V. Venugopalan and C. D. Patterson, "Surveying the hardware trojan threat landscape for the Internet-of-Things," *J. Hardw. Syst. Secur.*, vol. 2, no. 2, pp. 131–141, Apr. 2018, doi: 10.1007/s41635-018-0037-2.

[42] G. Agrawal, "A survey on attacks and approaches of intrusion detection systems," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 8, pp. 499–504, Aug. 2017, doi: 10.26483/ijarcs.v8i8.4771.

[43] Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization," *Expert Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117597, doi: 10.1016/j.eswa.2022.117597.

[44] S. Tug, W. Meng, and Y. Wang, "CBSigIDS: Towards collaborative blockchained signature-based intrusion detection," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1228–1235, doi: 10.1109/Cybermatics_2018.2018.00217.

[45] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, *Towards Blockchain-Based Collaborative Intrusion Detection Systems*. Cham, Switzerland: Springer, 2018, pp. 107–118, doi: 10.1007/978-3-319-99843-5_10.

[46] S. Antonatos, K. G. Anagnostakis, and E. P. Markatos, "Generating realistic workloads for network intrusion detection systems," in *Proc. 4th Int. Workshop Softw. Perform.*, Jan. 2004, pp. 207–215, doi: 10.1145/974044.974078.

[47] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput.*, vol. 23, no. 2, pp. 1397–1418, Oct. 2019, doi: 10.1007/s10586-019-03008-x.

[48] S. T. Eckmann, G. Vigna, and R. A. Kemmerer, "STATL: An attack language for state-based intrusion detection," *J. Comput. Secur.*, vol. 10, nos. 1–2, pp. 71–103, Jan. 2002, doi: 10.3233/jcs-2002-101-204.

[49] R. Sahani, Shatabdinalini, C. Rout, J. C. Badajena, A. K. Jena, and H. Das, *Classification of Intrusion Detection Using Data Mining Techniques*. Singapore: Springer, 2018, pp. 753–764, doi: 10.1007/978-981-10-7871-2_72.

[50] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, Feb. 2009, doi: 10.1016/j.cose.2008.08.003.

[51] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. IEEE Found. New Directions Data Mining Workshop*, Piscataway, NJ, USA, Jan. 2003, pp. 172–179.

[52] P. Saurabh and B. Verma, "Immunity inspired cooperative agent based security system," *Int. Arab J. Inf. Technol.*, vol. 15, pp. 289–295, Jan. 2018.

[53] M. Jha and R. Acharya, "An immune inspired unsupervised intrusion detection system for detection of novel attacks," in *Proc. IEEE Conf. Intell. Secur. Informat. (ISI)*, Sep. 2016, pp. 292–297, doi: 10.1109/ISI.2016.7745493.

[54] W. Jung, H. Zhao, M. Sun, and G. Zhou, "IoT botnet detection via power consumption modeling," *Smart Health*, vol. 15, Mar. 2020, Art. no. 100103, doi: 10.1016/j.smhl.2019.100103.

[55] V. W. Samawi, S. A. Yousif, and N. M. G. Al-Saidi, "Intrusion detection system: An automatic machine learning algorithms using Auto- WEKA," in *Proc. IEEE 13th Control Syst. Graduate Res. Colloq. (ICSGRC)*, Jul. 2022, pp. 42–46.

[56] I. Gad, A. E. Hassanien, A. Darwish, and M. Tang, *A Hybrid Quantum Deep Learning Approach Based on Intelligent Optimization to Predict the Broiler Energies*. Singapore: Springer, 2022, pp. 693–704, doi: 10.1007/978-981-16-8656-6_61.

[57] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. S1, pp. 949–961, Jan. 2019.

[58] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "$Deep-full-range$: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019, doi: 10.1109/ACCESS.2019.2908225.

[59] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors*, vol. 23, no. 9, p. 4441, May 2023, doi: 10.3390/s23094441.

[60] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.

[61] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2016, pp. 581–585, doi: 10.1109/ICCSN.2016.7586590.

[62] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019, doi: 10.3390/app9204396.

[63] T. Le, M. Vo, B. Vo, E. Hwang, S. Rho, and S. Baik, "Improving electric energy consumption prediction using CNN and bi-LSTM," *Appl. Sci.*, vol. 9, no. 20, p. 4237, Oct. 2019, doi: 10.3390/app9204237.

[64] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/s10462-020-09825-6.

[65] Y. Peng, M. Liao, H. Deng, L. Ao, Y. Song, W. Huang, and J. Hua, "CNN–SVM: A classification method for fruit fly image with the complex background," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 5, no. 2, pp. 181–185, May 2020, doi: 10.1049/iet-cps.2019.0069.

[66] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Appl. Sci.*, vol. 9, no. 21, p. 4500, Oct. 2019, doi: 10.3390/app9214500.

[67] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 113, pp. 418–427, Dec. 2020, doi: 10.1016/j.future.2020.07.042.

[68] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[69] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018, doi: 10.1109/ACCESS.2018.2867564.

[70] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6, doi: 10.1109/ITSC.2017.8317872.

[71] B. Lee, S. Amaresh, C. Green, and D. Engels, "Comparative study of deep learning models for network intrusion detection," *SMU Data Sci. Rev.*, vol. 1, no. 1, p. 8, 2018.

[72] I. H. Sarker, "Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective," *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 154, 2021.

[73] G. Kim, "Lstm-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, *arXiv:1611.01726*

[74] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Deep Android malware detection and classification," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1677–1683, doi: 10.1109/ICACCI.2017.8126084.

[75] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *J. Enterprise Inf. Manage.*, vol. 36, no. 3, pp. 747–766, Jun. 2020, doi: 10.1108/jeim-01-2020-0036.

[76] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A LSTM based framework for handling multiclass imbalance in DGA botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, Jan. 2018.

[77] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov, "Good semi-supervised learning that requires a bad GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2017.

[78] H. Aghakhani, A. Machiry, S. Nilizadeh, C. Kruegel, and G. Vigna, "Detecting deceptive reviews using generative adversarial networks," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 89–95, doi: 10.1109/SPW.2018.00022.

[79] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[80] V. Bok, *GANs in Action*. Shelter Island, NY, USA: Manning Publications, 2019.

[81] M. Wiatrak, S. V. Albrecht, and A. Nystrom, "Stabilizing generative adversarial networks: A survey," 2019, *arXiv:1910.00927*.

[82] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.

[83] X. Pei, S. Tian, L. Yu, H. Wang, and Y. Peng, "A two-stream network based on capsule networks and sliced recurrent neural networks for DGA botnet detection," *J. Netw. Syst. Manage.*, vol. 28, no. 4, pp. 1694–1721, Jul. 2020, doi: 10.1007/s10922-020-09554-9.

[84] M. Alauthman, N. Aslam, M. Al-Kasassbeh, S. Khan, A. Al-Qerem, and K.-K. R. Choo, "An efficient reinforcement learning-based botnet detection approach," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102479, doi: 10.1016/j.jnca.2019.102479.

[85] D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui, "Evading machine learning botnet detection models via deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761337.

[86] A. Rezaei, "Using ensemble learning technique for detecting botnet on IoT," *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 148, 2021.

[87] O. D. Okey, S. S. Maidin, P. Adasme, R. L. Rosa, M. Saadi, D. C. Melgarejo, and D. Z. Rodríguez, "BoostedEnML: Efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning," *Sensors*, vol. 22, no. 19, p. 7409, Sep. 2022, doi: 10.3390/s22197409.

[88] N. Moustafa, B. Turnbull, and K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[89] S. Srinivasan and P. Deepalakshmi, "Enhancing the security in cyber-world by detecting the botnets using ensemble classification based machine learning," *Meas., Sensors*, vol. 25, Feb. 2023, Art. no. 100624, doi: 10.1016/j.measen.2022.100624.

[90] E. Tsogbaatar, M. H. Bhuyan, Y. Taenaka, D. Fall, K. Gonchigsumlaa, E. Elmroth, and Y. Kadobayashi, "DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100391, doi: 10.1016/j.iot.2021.100391.

[91] V. Tomer and S. Sharma, "Detecting IoT attacks using an ensemble machine learning model," *Future Internet*, vol. 14, no. 4, p. 102, Mar. 2022.

[92] A. R. Javed, Z. Jalil, S. A. Moqurrab, S. Abbas, and X. Liu, "Ensemble AdaBoost classifier for accurate and fast detection of botnet attacks in connected vehicles," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 10, p. 4088, Oct. 2022.

[93] T.-J. Liu, T.-S. Lin, and C.-W. Chen, "An ensemble machine learning botnet detection framework based on noise filtering," *J. Internet Technol.*, vol. 22, no. 6, pp. 1347–1357, Nov. 2021.

[94] K. M. Yogesh, T. Stephan, M. B. Bharath, I. Gad, S. Arpitha, and M. M. Prakash, "Characterization of darknet traffic detection using time domain features," in *Proc. Int. Conf. Comput. Vis. Internet Things (ICCVIoT'T)*, Nov. 2023, pp. 233–237, doi: 10.1049/icp.2023.2881.

[95] Q. Abu Al-Haija, M. Krichen, and W. Abu Elhaija, "Machine-learning-based darknet traffic detection system for IoT applications," *Electronics*, vol. 11, no. 4, p. 556, Feb. 2022, doi: 10.3390/electronics11040556.

[96] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An aggregated mutual information based feature selection with machine learning methods for enhancing IoT botnet attack detection," *Sensors*, vol. 22, no. 1, p. 185, Dec. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/185

[97] A. Khraisat, "A novel ensemble of hybrid intrusion detection system for detecting Internet of Things attacks," *Electronics*, vol. 8, no. 11, p. 1210, 2019.

[98] P. L. S. Jayalaxmi, G. Kumar, R. Saha, M. Conti, T.-H. Kim, and R. Thomas, "DeBot: A deep learning-based model for bot detection in industrial Internet-of-things," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108214, doi: 10.1016/j.compeleceng.2022.108214.

[99] S. Alosaimi and S. M. Almutairi, "An intrusion detection system using BoT-IoT," *Appl. Sci.*, vol. 13, no. 9, p. 5427, Apr. 2023, doi: 10.3390/app13095427.

[100] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, Mar. 2021, doi: 10.1109/JIOT.2020.3002255.

[101] Z. Ding, S. Jiang, and J. Zhao, "Take a close look at mode collapse and vanishing gradient in GAN," in *Proc. IEEE 2nd Int. Conf. Electron. Technol., Commun. Inf. (ICETCI)*, May 2022, pp. 597–602, doi: 10.1109/ICETCI55101.2022.9832406.

[102] N. Granot, B. Feinstein, A. Shocher, S. Bagon, and M. Irani, "Drop the GAN: In defense of patches nearest neighbors as single image generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13450–13459, doi: 10.1109/CVPR52688.2022.01310.

[103] M. M. Saad, M. H. Rehmani, and R. O'Reilly, "Addressing the intra-class mode collapse problem using adaptive input image normalization in GAN-based X-ray images," in *Proc. 44th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2022, pp. 2049–2052, doi: 10.1109/EMBC48229.2022.9871260.

[104] W. S. Hamza, H. M. Ibrahim, M. A. Shyaa, and J. J. Stephan, "IoT botnet detection: Challenges and issues," *Test Eng. Manag*, vol. 83, pp. 15092–15097, Jan. 2020.

[105] I. Ali, A. I. A. Ahmed, A. Almogren, M. A. Raza, S. A. Shah, A. Khan, and A. Gani, "Systematic literature review on IoT-based botnet attack," *IEEE Access*, vol. 8, pp. 212220–212232, 2020, doi: 10.1109/ACCESS.2020.3039985.

[106] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, "Machine learning-based botnet detection in software-defined network: A systematic review," *Symmetry*, vol. 13, no. 5, p. 866, May 2021, doi: 10.3390/sym13050866.

[107] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of Things botnet detection approaches: Analysis and recommendations for future research," *Appl. Sci.*, vol. 11, no. 12, p. 5713, Jun. 2021, doi: 10.3390/app11125713.

[108] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo, "Survey on botnet detection techniques: Classification, methods, and evaluation," *Math. Problems Eng.*, vol. 2021, pp. 1–24, Apr. 2021, doi: 10.1155/2021/6640499.

TAMARA AL-SHURBAJI received the B.Sc. degree in computer science from Hashemite University, in 2007, and the M.S. degree in computer science from Amman Arab University, in 2018. She is currently pursuing the Ph.D. degree with the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM). She was a Researcher of image processing areas and in the process to enhance some algorithms. Her current research interests include deep learning, IPv6 security, intrusion detection system (IDS), and the Internet of Things.

MOHAMMED ANBAR (Member, IEEE) received the Ph.D. degree in advanced computer network from Universiti Sains Malaysia (USM). He is currently a Senior Lecturer with the National Advanced IPv6 Centre (NAv6), USM. His current research interests include malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the Internet of Things (IoT), and IPv6 security.

SELVAKUMAR MANICKAM (Member, IEEE) is currently the Director of the National Advanced IPv6 Centre and an Associate Professor specializing in cybersecurity, the Internet of Things, industry 4.0, cloud computing, big data, and machine learning. He has authored and co-authored more than 220 papers in journals, conference proceedings, and book reviews. He has graduated 18 Ph.D. students in addition to master's and bachelor's students. He has given several keynote speeches and dozens of invited lectures and workshops at conferences, international universities, and industry. He has given talks and training on internet security, the Internet of Things, industry 4.0, IPv6, machine learning, software development, and embedded and OS kernel technologies at various organizations and seminars. He also lectures in various computer science and IT courses, including developing new courseware in tandem with current technology trends. He is involved in various organizations and forums locally and globally. Previously, he was with Intel Corporation, and a few start-ups working in related areas before moving to academia. While building his profile academically, he is still very involved in industrial projects involving industrial communication protocol, robotic process automation, machine learning, and data analytics using open source platforms. He also has experience in the building IoT embedded, server, mobile, and web-based applications.

IZNAN H HASBULLAH received the Bachelor of Science degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently pursuing the M.Sc. degree in advanced network security. He has experience working as a Software Developer, a Research and Development Consultant, and a Network Security Auditor, prior to joining the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, in 2010, as a Research Officer. His research interests include unified communication, telematics, network security, network protocols, and next generation network.

NADIA ALFRIEHAT received the B.S. degree in computer science from Jerash University, and the M.Sc. degree in computer science from Amman Arab University, in 2018. She is currently pursuing the Ph.D. degree with the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM). Her research interests include computer networks, network security, intrusion detection systems (IDS), artificial intelligence (AI), and the Internet of Things (IoT).

BASIM AHMAD ALABSI received the B.Sc. degree in computer science from Al-Azhar University, Palestine, in 2000, the M.Sc. degree in computer science from Amman Arab University, Jordan, in 2005, and the Ph.D. degree in internet infrastructure security from Universiti Sains Malaysia (USM), in 2020. He is currently an Assistant Professor with Najran University. His current research interests include the Internet of Things (IoT), routing protocol for low-power and lossy networks (RPL) security, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), and IPv6 security.

AHMAD REDA ALZIGHAIBI received the B.Sc., M.Sc. and Ph.D. degrees in computer information systems from the University of Canberra, Canberra, Australia, in 2009, 2012, and 2017, respectively. He is currently the Head of the Information Systems Department and the Vice Dean of the Quality and Development, College of Computer Science and Engineering, Yanbu, Saudi Arabia. His research interests include artificial intelligence and cyber security, big data studies, the Internet of Things (IoT), and natural language processing. He is a member of the Computer Algorithm Series of the IEEE Computer Society Press (CAS).

HASAN HASHIM received the master's degree in management information systems from the Graduate School of Computer and Information Sciences, Nova South Eastern University, USA, in 2007. In 2014, he engaged in research and earned the Ph.D. degree in information systems from the Faculty of Science, Information School, U.K. He currently holds positions as the Vice Dean of Academic Affairs and the Head of the Computer Science Department, College of Computer Science and Engineering, Yanbu. His research interests include diversity of information systems knowledge, such as infrastructure in e-government and information systems domains, integrated electronic systems, smart cities, artificial intelligence, and role of information systems.