

Review article



A quantitative and deployment-aware survey of AI-based techniques for IoT botnet detection and classification

Sara Afzal ^{a,b,*}, Abraham Gebrehiwot ^a, Filippo Maria Lauria ^a,
Farrukh Aslam Khan ^c

^a Institute of Informatics and Telematics (IIT), National Research Council (CNR), Pisa, 56124, Italy

^b IMT School for Advanced Studies, Lucca, 55100, Italy

^c College of Engineering and Advanced Computing, Alfaisal University, Riyadh, 11533, Saudi Arabia

ARTICLE INFO

Keywords:

IoT botnets
Botnet detection
Botnet classification
Artificial intelligence
Deployment feasibility
Cross-dataset generalization
Adversarial robustness

ABSTRACT

The rapid expansion of Internet of Things (IoT) systems has increased their exposure to botnet-based threats, while device heterogeneity, constrained resources, weak security configurations, and evolving attack behavior continue to complicate effective detection. Artificial intelligence (AI)-based approaches have consequently become central to IoT botnet detection and classification, spanning machine learning, deep learning, hybrid and ensemble models, graph-based methods, deep reinforcement learning, federated learning, and edge-oriented frameworks. This survey presents a quantitative and deployment-aware analysis of 43 empirical studies published between 2020 and August 2026. In addition to methodological synthesis, the reviewed literature is analyzed across primary technique category, classification depth, dataset usage and breadth, cross-dataset validation, deployment evidence, evaluation practices, explainability, zero-day evaluation, and adversarial robustness. Hybrid approaches form the largest technique category (32.6%), followed by deep learning (23.3%). Binary detection and attack-type classification are each addressed in 62.8% of the studies, whereas malware-family classification and lifecycle-stage modeling remain less common at 27.9% and 14.0%, respectively. Dataset dependence also remains substantial: N-BaIoT is used in 48.8% of the corpus, while only 32.6% of studies perform explicit cross-dataset validation. Moreover, only 14.0% provide strong deployment evidence, and explicit XAI, zero-day evaluation, and adversarial-robustness assessment appear in 9.3%, 18.6%, and 7.0% of studies, respectively. These findings indicate that future IoT botnet research should move beyond benchmark-centered accuracy toward generalizable, deployment-efficient, behavior-aware, interpretable, and adversarially robust detection systems.

1. Introduction

The Internet of Things (IoT) has transformed modern computing by enabling large numbers of connected devices to exchange data and provide intelligent services across domains [1,2] such as healthcare, smart cities, transportation, industrial automation, and consumer environments [3,4]. This extensive connectivity has also enlarged the attack surface available to cybercriminals. IoT devices often differ considerably in hardware capability, operating systems, communication protocols, firmware support, and security configuration, while many operate with constrained computational and memory resources [5]. Weak credentials, insecure remote-

* Corresponding author at: Institute of Informatics and Telematics (IIT), National Research Council (CNR), Pisa, 56124, Italy.

E-mail address: sara.afzal@iit.cnr.it; sara.afzal@imtlucca.it (S. Afzal).

management interfaces, delayed patching, and limited monitoring can therefore make these devices attractive targets for large-scale compromise.

IoT botnets consist of compromised connected devices that can be remotely coordinated to perform malicious activities such as distributed denial-of-service (DDoS) attacks, scanning, malware propagation, spam distribution, and data theft [6,7]. The Mirai botnet demonstrated the scale of this threat by exploiting weak authentication on Internet-connected devices and coordinating compromised systems for large-volume DDoS attacks [8]. Subsequent IoT botnets have employed increasingly diverse propagation, persistence, and command-and-control mechanisms, including encrypted communication, decentralized architectures, and adaptive behavior [9]. Fig. 1 illustrates representative attacks associated with IoT botnets, while Fig. 2 presents prominent botnets in the evolution of the threat landscape.

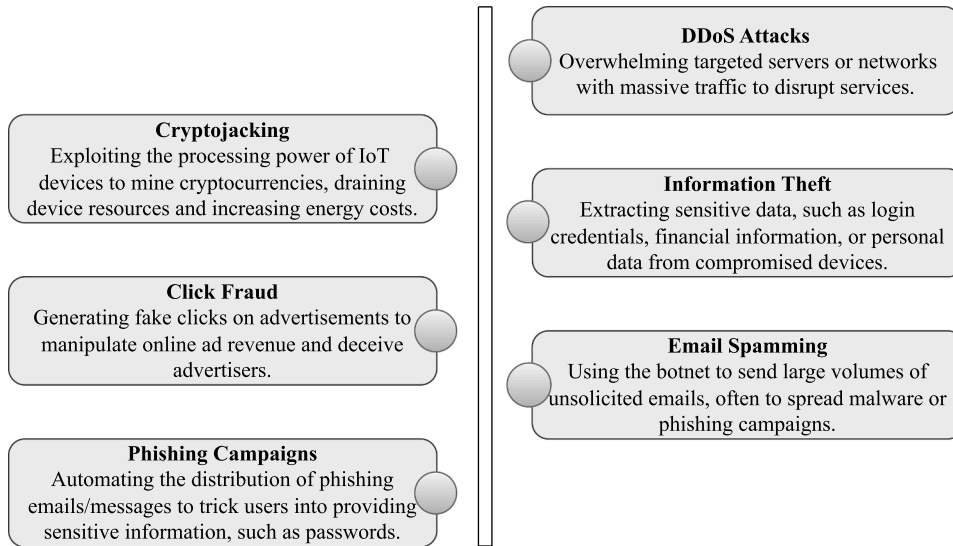


Fig. 1. Representative malicious activities associated with IoT botnets.

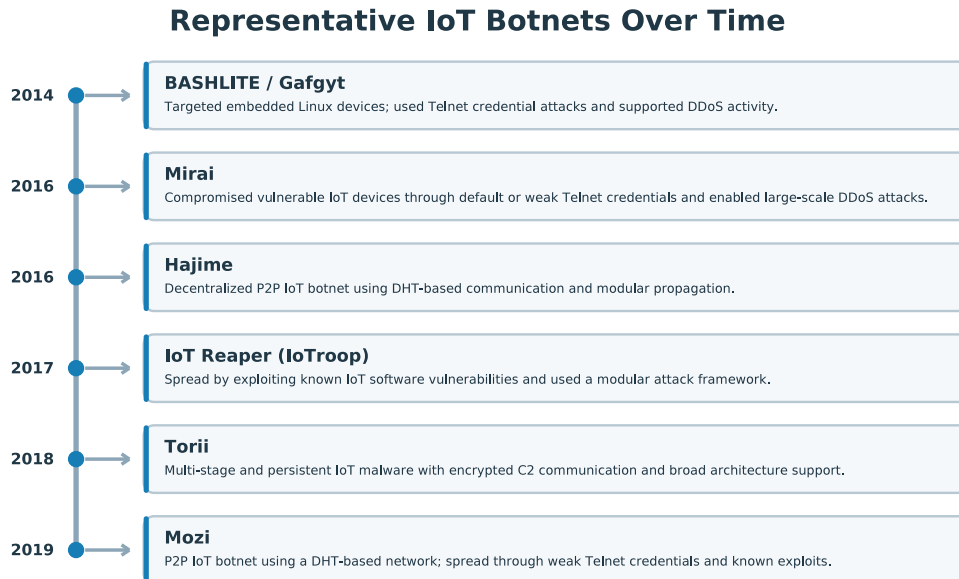


Fig. 2. Representative IoT botnets in the evolution of the threat landscape.

Traditional detection mechanisms, including signature-, rule-, anomaly-, behavior-, flow-, and DNS-based approaches, provide useful mechanisms for identifying malicious activity [10,11]. However, rapidly evolving malware, encrypted communication, heterogeneous device behavior, and changing traffic distributions can reduce the effectiveness of static detection rules. These challenges

have motivated extensive research on artificial intelligence (AI)-based detection, including classical machine learning (ML), deep learning (DL), hybrid and ensemble architectures, graph-based learning, deep reinforcement learning (DRL), and federated learning.

ML approaches commonly operate on engineered traffic or behavioral features, whereas DL architectures can learn more complex nonlinear, temporal, or latent representations. Graph-based models provide mechanisms for representing communication relationships and coordinated behavior, while federated approaches enable collaborative learning without directly centralizing raw traffic. Recent research also increasingly considers lightweight edge deployment, explainability, zero-day detection, and robustness against adversarial manipulation.

Despite substantial progress in predictive performance, several challenges remain. Evaluation is still concentrated around a relatively small set of public benchmarks, limiting the evidence available for cross-device and cross-domain generalization. Classification capability also varies considerably: many studies support binary or attack-type decisions, whereas malware-family identification and lifecycle-stage modeling remain less common. Practical deployment introduces additional concerns involving inference latency, memory, computational cost, energy consumption, communication overhead, and hardware constraints. Furthermore, explainability, zero-day evaluation, and adversarial robustness are not yet consistently incorporated into experimental assessment.

A growing body of survey literature has addressed IoT botnet detection from perspectives including ML/DL methods, lifecycle analysis, datasets, deployment constraints, adversarial behavior, and explainable AI. As discussed in Section 4, several recent reviews already provide substantial coverage of these topics. The remaining methodological need is therefore not simply broader topic inclusion, but a consistent framework that allows methodological choices, classification capability, dataset generalization, operational evidence, and trustworthiness to be examined across the same body of empirical studies.

Accordingly, this paper presents a quantitative and deployment-aware survey of **43 empirical studies** on AI-driven IoT botnet detection and classification published between 2020 and August 2026. The selected literature is analyzed using a structured coding framework covering primary AI technique, classification depth, dataset usage and breadth, explicit cross-dataset validation, deployment evidence, evaluation-metric reporting, explainability, zero-day evaluation, and adversarial robustness. This integrated perspective enables the survey to assess not only what techniques achieve strong predictive results, but also the extent to which those results are supported by broader validation and practical deployment evidence.

The overall organization and analytical scope of the survey are illustrated in Fig. 3, while commonly used abbreviations are listed in Table 1.

Organization and Analytical Scope of the Survey

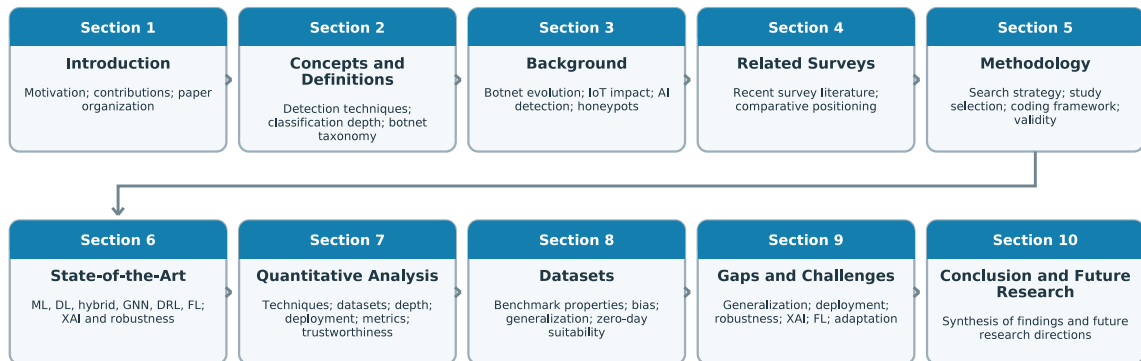


Fig. 3. Organization and analytical scope of the survey.

1.1. Motivation

The central motivation of this survey is the gap between strong benchmark performance and the broader requirements of dependable IoT botnet detection. High accuracy within a controlled dataset does not necessarily demonstrate transfer to different devices, malware families, traffic distributions, or deployment environments. Similarly, describing a model as lightweight, real-time, or edge-oriented does not by itself demonstrate practical feasibility without supporting hardware, timing, or resource measurements.

A meaningful assessment of IoT botnet detection therefore requires several dimensions to be considered together. The detection technique must be interpreted alongside the granularity of its output, the diversity of its evaluation data, the evidence of generalization, and the computational conditions under which it operates. Trustworthiness is also increasingly important because security systems may face previously unseen attacks, distribution shift, adversarial manipulation, and the need to provide interpretable decisions to analysts.

Table 1
Table of abbreviations.

Abbreviation	Full Form
AI	Artificial Intelligence
AUC	Area Under the Curve
C&C	Command and Control
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DGA	Domain Generation Algorithm
DL	Deep Learning
DNN	Deep Neural Network
DNS	Domain Name System
DRL	Deep Reinforcement Learning
FL	Federated Learning
FPR	False Positive Rate
GNN	Graph Neural Network
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IRC	Internet Relay Chat
LAN	Local Area Network
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NFV	Network Function Virtualization
P2P	Peer-to-Peer
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SDN	Software-Defined Networking
SLR	Systematic Literature Review
SSH	Secure Shell
SVM	Support Vector Machine
XAI	Explainable Artificial Intelligence

This survey addresses these requirements through a unified study-level analysis rather than treating technique, classification, dataset, deployment, and robustness characteristics as independent observations. The resulting synthesis is intended to clarify where current IoT botnet research is mature and where stronger evidence is still required for practical and trustworthy deployment.

1.2. Main contributions of the survey

The main contributions of this survey are summarized as follows:

- A structured review of **43 empirical studies** published between 2020 and August 2026, covering classical ML, DL, hybrid and ensemble methods, graph/GNN-based approaches, DRL, and federated learning for IoT botnet detection and classification.
- A quantitative study-level framework using mutually exclusive primary technique categories together with a non-cumulative classification-depth model covering binary detection, attack-type classification, malware-family classification, and lifecycle-stage modeling.
- A dataset-oriented analysis that distinguishes dataset popularity and dataset breadth from explicit cross-dataset validation, enabling a clearer assessment of benchmark dependence and generalization evidence.
- An evidence-based deployment assessment that evaluates reported hardware testing, inference and training characteristics, memory and model size, computational complexity, detection delay, scalability, energy consumption, and communication overhead.
- An integrated analysis of evaluation practices and trustworthiness, including predictive-metric reporting, explainability, zero-day evaluation, and empirical adversarial robustness, together with the resulting research gaps and future directions.

Organization of the Paper: The remainder of this paper is organized as follows. [Section 2](#) distinguishes botnet detection, classification objectives, and taxonomic categorization. [Section 3](#) summarizes the evolution of botnets, the impact of IoT, AI-based detection, and honeypot-assisted data collection. [Section 4](#) reviews and compares representative survey literature. [Section 5](#) describes the study-selection and coding methodology. [Section 6](#) provides a qualitative synthesis of state-of-the-art AI-based IoT botnet detection and classification approaches. [Section 7](#) presents the quantitative cross-study analysis. [Section 8](#) compares the principal datasets and their implications for model evaluation and generalization. [Section 9](#) synthesizes the major research gaps and open challenges. Finally, [Section 10](#) concludes the survey and outlines future research directions.

2. Detection, classification and categorization of botnets

A botnet is a collection of compromised Internet-connected devices that are remotely controlled by an attacker, commonly referred to as a botmaster, without the knowledge or authorization of the legitimate device owners. Compromised devices may include conventional computers, mobile systems, routers, cameras, sensors, and other Internet of Things (IoT) devices. Botnets can be used to conduct malicious activities such as distributed denial-of-service (DDoS) attacks, spam distribution, credential theft, data exfiltration, click fraud, scanning, and malware propagation [6,12].

In this survey, *detection*, *classification*, and *categorization* are treated as related but distinct concepts. **Botnet detection** concerns determining whether an observed device, flow, communication pattern, or traffic instance exhibits botnet-related activity. **Botnet classification** concerns assigning detected activity to a more specific operational label, such as an attack type, malware family, or botnet lifecycle stage. **Botnet categorization**, in contrast, refers to the conceptual organization of botnets according to properties such as command-and-control architecture, communication mechanism, malicious purpose, malware family, or target platform. This distinction is important because a taxonomy used to describe botnets does not necessarily correspond to the output classes predicted by a detection model.

Fig. 4 provides a conceptual overview of these relationships. Detection techniques describe *how evidence of botnet activity is identified*; classification objectives describe *what level of information the detector predicts*; and taxonomic categories describe *how botnets themselves can be characterized*.

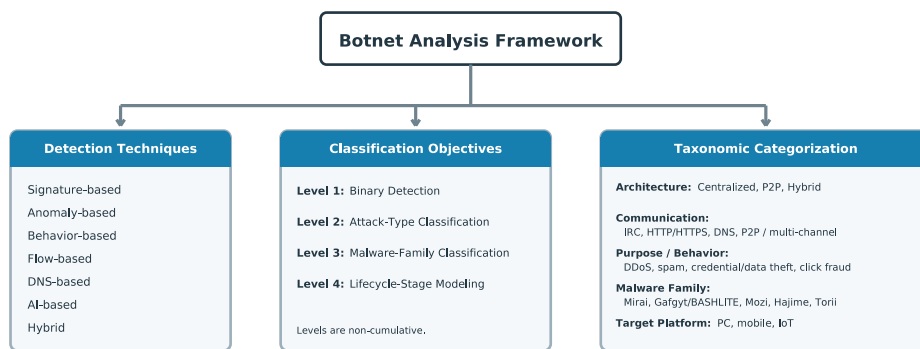


Fig. 4. Conceptual relationship between botnet detection techniques, classification objectives, and taxonomic categorization.

2.1. Detection techniques

Botnet detection can be performed using several complementary sources of evidence and decision mechanisms [13–15]. These techniques are not mutually exclusive. For example, an ML model may operate on flow-level features, while a behavior-based detector may additionally use graph or temporal representations. Hybrid systems may combine several such mechanisms.

1. **Signature-Based Detection:** Signature-based detection compares observed traffic, files, commands, or other artifacts against predefined indicators associated with known malware or botnet activity [16]. Its principal advantages are efficiency and low computational cost when signatures are available. However, it is inherently dependent on previously identified patterns and is therefore less effective against new variants, polymorphic malware, obfuscated behavior, and previously unseen botnets.
2. **Anomaly-Based Detection:** Anomaly-based detection constructs a representation of expected or normal behavior and identifies observations that deviate sufficiently from this baseline [17,18]. Because detection does not necessarily depend on a known attack signature, anomaly-based systems can identify previously unseen malicious activity. Their main challenge is that legitimate changes in device or network behavior may also appear anomalous, leading to false alarms if the normal profile is poorly defined or becomes outdated.
3. **Behavior-Based Detection:** Behavior-based detection explicitly examines actions or relationships associated with malicious operation, such as scanning, repeated connection attempts, propagation behavior, coordinated communication, command-and-control activity, or characteristic temporal sequences [19–21]. Unlike anomaly detection, which primarily identifies deviation from a baseline, behavior-based detection seeks evidence of specific behavioral patterns or interactions associated with malicious activity. The two approaches may nevertheless be combined within the same system.
4. **Flow-Based Detection:** Flow-based detection analyzes communication metadata such as source and destination addresses, ports, protocols, packet counts, byte counts, flow duration, timing, and related statistical characteristics [22–24]. Because payload inspection is not required, flow-based analysis is suitable for high-volume monitoring and remains applicable when application payloads are unavailable or encrypted. However, aggregation may remove information needed to identify stealthy, low-volume, or highly similar benign and malicious communication.

5. **DNS-Based Detection:** DNS-based techniques analyze domain-resolution behavior to identify suspicious communication patterns associated with botnet infrastructure [25,26]. Relevant indicators may include unusual query patterns, rapidly changing domains, repeated failed resolutions, or characteristics associated with domain-generation algorithms. Such methods can be computationally efficient, but their effectiveness decreases when botnets avoid DNS, use hard-coded addresses, rely on peer-to-peer communication, or obscure observable DNS activity.
6. **AI-Based Detection:** ML and DL provide learning mechanisms that can operate on flow, behavioral, temporal, graph, DNS, or other representations [27–29]. Classical ML models can provide efficient inference when informative features are available, whereas DL architectures can learn more complex nonlinear, spatial, temporal, or latent representations. Their computational requirements vary considerably; therefore, AI-based detection should not automatically be assumed to be either lightweight or resource intensive. Performance depends on the model architecture, feature-processing pipeline, dataset quality, hardware platform, and deployment setting.
7. **Hybrid Detection:** Hybrid approaches combine multiple detection mechanisms, feature representations, learning models, or optimization procedures [30–32]. Examples include anomaly detection followed by supervised classification, deep feature extraction combined with conventional ML, ensemble classifiers, or graph and temporal components integrated within a common framework. Hybridization can improve modeling capability but may also increase computational cost, preprocessing requirements, and implementation complexity.

Table 2 summarizes the main detection perspectives. The comparison emphasizes that the categories describe different forms of evidence or decision mechanisms and may therefore coexist within a single detection framework.

Table 2
Comparison of botnet detection techniques.

Technique	Primary Basis	Strengths	Limitations
Signature-Based	Known indicators, patterns, rules, or malware signatures	Fast and efficient for recognized threats; straightforward to operationalize	Limited ability to detect previously unseen, modified, or obfuscated botnets
Anomaly-Based	Deviation from a learned or statistically defined representation of normal behavior	Can identify unknown or previously unseen activity without an explicit attack signature	Sensitive to changes in legitimate behavior; poorly calibrated baselines may generate false alarms
Behavior-Based	Malicious actions, communication relationships, temporal sequences, or coordinated activity	Can represent operational botnet behavior and support earlier or more contextual detection	Requires suitable behavioral observations and may be difficult to distinguish from unusual benign activity
Flow-Based	Flow metadata and statistical characteristics	Does not require payload inspection; scalable for gateway and network-level monitoring	Aggregation may obscure stealthy or low-volume malicious behavior
DNS-Based	Domain-resolution and DNS communication patterns	Useful for detecting suspicious domain behavior and some forms of C&C infrastructure	Limited when DNS is bypassed, obscured, or not central to botnet communication
AI-Based	Data-driven learning using ML, DL, graph, temporal, or related models	Learns complex decision boundaries and can operate on several forms of network and behavioral evidence	Dependent on data quality and evaluation protocol; computational and interpretability costs vary by model
Hybrid	Combination of multiple models, representations, or detection mechanisms	Can exploit complementary evidence and improve detection capability	Additional processing stages can increase complexity, latency, and resource requirements

2.2. Classification objectives

Once botnet-related activity is detected, a model may provide different levels of classification detail. In this survey, classification depth is represented using four non-cumulative analytical levels. These levels describe the *prediction capability demonstrated by a detection system*, rather than the general taxonomy of the botnet itself.

1. **Level 1 — Binary Botnet Detection:** The model distinguishes botnet or malicious activity from benign activity. This is the coarsest classification objective and answers whether botnet-related behavior is present.
2. **Level 2 — Attack-Type Classification:** The model distinguishes among operational attack categories such as DDoS, DoS, scanning, reconnaissance, brute-force activity, or other explicitly defined attack types. This provides greater operational detail than a binary malicious/benign decision.
3. **Level 3 — Malware-Family Classification:** The model identifies the malware or botnet family responsible for the observed activity, for example Mirai, Gafgyt/BASHLITE, Mozi, Hajime, or Torii. Family-level classification can support attribution, threat intelligence, and family-specific mitigation.
4. **Level 4 — Lifecycle-Stage Modeling:** The model identifies or explicitly models stages of botnet operation, such as reconnaissance or scanning, exploitation, infection, propagation, command-and-control establishment, and attack execution. Lifecycle awareness can support earlier intervention by identifying malicious behavior before the final attack phase.

These levels are deliberately **non-cumulative**. A study that performs malware-family classification is not automatically assumed to perform attack-type classification, and a lifecycle-oriented study is not assumed to provide binary, attack-type, or malware-family

Table 3
Classification-depth framework for IoT botnet detection.

Level	Classification Objective	Typical Output	Operational Relevance
Level 1	Binary Detection	Benign vs. botnet/malicious	Provides basic identification of compromised or malicious activity.
Level 2	Attack-Type Classification	DDoS, DoS, scanning, reconnaissance, brute force, or other attack classes	Supports attack-specific response and mitigation.
Level 3	Malware-Family Classification	Mirai, Gafgyt/BASHLITE, Mozi, Hajime, Torii, or other families	Supports malware attribution, threat intelligence, and family-specific response.
Level 4	Lifecycle-Stage Modeling	Scanning, exploitation, infection, propagation, C&C, or attack execution	Supports behavioral understanding and potentially earlier intervention.

outputs unless those tasks are explicitly evaluated. Similarly, the terms *binary* and *multiclass* alone are insufficient to determine classification depth because a multiclass problem may represent attack types, malware families, device states, or another labeling scheme.

Table 3 summarizes the four classification objectives.

2.3. Taxonomic categorization of botnets

Botnets can also be categorized according to structural and operational characteristics. These categories provide a descriptive taxonomy and should not be confused with the classification-depth levels defined above. A single botnet may belong to several categories simultaneously; for example, an IoT botnet may use a decentralized architecture, communicate through web-based protocols, and primarily conduct DDoS attacks.

2.3.1. Architecture-based categorization

Botnet architecture describes how compromised devices receive commands and coordinate malicious activity.

- **Centralized Botnets:** Compromised devices communicate with one or more central command-and-control (C&C) servers. The structure simplifies coordination but creates infrastructure that defenders may identify and disrupt [33,34].
- **Peer-to-Peer Botnets:** Bots exchange information directly with peer devices rather than depending exclusively on a central controller. Decentralization reduces single points of failure and can make infrastructure discovery and takedown more difficult [35–38].
- **Hybrid Botnets:** Hybrid architectures combine centralized and decentralized control mechanisms, for example by maintaining central coordination while providing peer-based fallback communication [39,40]. Such designs can provide both administrative flexibility and resilience against infrastructure disruption.

2.3.2. Communication-based categorization

Botnets may also be categorized according to the protocols or channels used for command-and-control communication.

- **IRC-Based Botnets:** Historically, many botnets used Internet Relay Chat (IRC) channels for command distribution. Their recognizable communication structure made them comparatively amenable to signature and traffic-analysis techniques [41,42].
- **HTTP/HTTPS-Based Botnets:** Web protocols allow malicious communication to resemble ordinary application traffic and, in the case of HTTPS, conceal payload content through encryption [43,44].
- **DNS-Based Botnets:** DNS can be used for command discovery, domain rotation, or covert communication. Domain-generation mechanisms can further complicate infrastructure blocking and require specialized DNS-oriented analysis [45].
- **Peer-to-Peer and Multi-Channel Communication:** Some botnets distribute control across peer relationships or combine several communication mechanisms. Such designs reduce dependence on a single observable protocol and may increase resilience to disruption.

2.3.3. Purpose- and behavior-based categorization

The malicious objective of a botnet provides another useful categorization dimension.

- **DDoS Botnets:** These coordinate large numbers of compromised devices to overwhelm a target with traffic or requests [46]. Large-scale DDoS campaigns are particularly relevant to IoT botnets because widespread insecure devices provide substantial aggregate attack capacity.
- **Spam Botnets:** Compromised hosts may be used to distribute unsolicited messages, malicious links, or other unwanted content [47].
- **Credential-Theft and Data-Stealing Botnets:** These botnets collect credentials or other sensitive information for unauthorized access, financial gain, or subsequent attacks [48].

- **Click-Fraud Botnets:** Compromised systems generate fraudulent interactions with online advertisements or services to create artificial revenue or traffic [49].

The categories above are not necessarily exclusive because a single botnet may support several malicious functions or change its purpose according to commands issued by the operator.

2.3.4. Malware-family-based categorization

Botnets may be grouped according to malware lineage or family. Prominent families associated with IoT environments include Mirai, Gafgyt/BASHLITE, Mozi, Hajime, and Torii [50,51]. Family-based categorization is useful because related variants may share propagation methods, communication behavior, or attack functionality. However, continuously evolving variants can make strict family boundaries difficult to maintain.

2.3.5. Platform-based categorization

Botnets can additionally be categorized according to the types of devices they compromise.

- **PC Botnets:** Traditional botnets primarily compromise desktop and server-class systems [52].
- **Mobile Botnets:** Mobile botnets target smartphones or tablets, commonly through malicious applications, software vulnerabilities, or compromised services [53].
- **IoT Botnets:** IoT botnets compromise connected devices such as cameras, routers, smart-home equipment, sensors, and other embedded systems [54]. Their security challenges arise from large device populations, hardware and protocol heterogeneity, constrained resources, limited monitoring capabilities, and inconsistent patching or update mechanisms.

Table 4 summarizes these taxonomic dimensions and their relevance to botnet analysis.

Table 4

Taxonomic categorization of botnets.

Dimension	Categories / Examples	Analytical Relevance	Important Consideration
Architecture	Centralized, P2P, Hybrid	Describes command distribution, coordination structure, and resilience to infrastructure disruption	Architectures may evolve or combine several control mechanisms
Communication	IRC, HTTP/HTTPS, DNS, P2P, multi-channel	Characterizes how bots discover infrastructure and exchange commands or data	A single botnet may use several protocols or fallback communication mechanisms
Purpose / Behavior	DDoS, spam, credential theft, data exfiltration, click fraud	Describes the malicious objectives and operational behavior of the botnet	Individual botnets may perform several activities depending on botmaster commands
Malware Family	Mirai, Gafgyt/BASHLITE, Mozi, Hajime, Torii	Supports attribution and comparison of family-specific propagation and communication characteristics	Variants evolve rapidly and may share behaviors across family boundaries
Target Platform	PC, mobile, IoT	Identifies the device ecosystem being compromised and the corresponding security constraints	Modern botnets may target heterogeneous device classes simultaneously

Overall, detection technique, classification depth, and botnet taxonomy represent three complementary analytical perspectives. Detection techniques describe how malicious evidence is identified; classification depth specifies the granularity of the prediction produced by a model; and taxonomic categorization describes structural or operational characteristics of the botnet itself. Maintaining this distinction provides a consistent conceptual basis for the study-selection, coding, and quantitative analyses presented in the subsequent sections.

3. Background: evolution of botnets

Botnets have evolved from relatively simple networks of compromised computers controlled through centralized command channels into heterogeneous infrastructures spanning personal computers, mobile systems, cloud platforms, and Internet of Things (IoT) devices. This evolution has been accompanied by changes in propagation mechanisms, command-and-control (C&C) architectures, evasion strategies, and attack objectives. The growth of IoT has been particularly significant because large populations of continuously connected devices often combine heterogeneous hardware and software with limited security capabilities.

This section briefly outlines the historical evolution of botnets, the impact of IoT on the botnet threat landscape, the limitations of traditional detection mechanisms, the increasing role of AI-based detection, and the use of honeypots for collecting botnet-related data.

3.1. Botnets: from early threats to modern complexities

Botnets have existed in different forms for several decades and have progressively developed from centralized systems into more distributed and resilient infrastructures [55]. Early botnets frequently relied on Internet Relay Chat (IRC) channels for command distribution [56], while later generations supported increasingly diverse malicious activities, including DDoS attacks, spam distribution, credential theft, malware propagation, and information exfiltration [57,58].

Malware such as Zeus demonstrated the use of botnet infrastructures for credential theft and financial fraud, whereas Conficker demonstrated large-scale automated propagation and persistence [59,60]. Botnets subsequently adopted more resilient communication mechanisms, including P2P architectures, web-based C&C, DNS-based communication, and encrypted channels. These developments reduced dependence on easily identifiable central infrastructure and made malicious communication more difficult to distinguish from legitimate traffic.

Consequently, botnet detection increasingly requires analysis beyond known signatures, including traffic characteristics, communication relationships, temporal behavior, and other indicators of coordinated malicious activity.

3.2. The impact of IoT on botnet evolution

The expansion of IoT significantly increased both the scale and diversity of devices available for botnet compromise [61]. IoT environments include cameras, routers, sensors, smart-home systems, industrial equipment, and numerous other embedded devices. Fig. 5 illustrates representative IoT application domains.

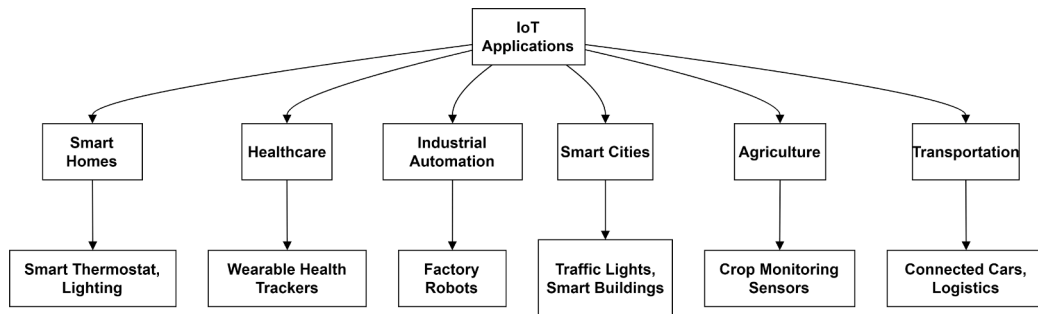


Fig. 5. Representative application domains of Internet of Things technologies.

IoT devices are attractive botnet targets because many remain continuously connected while differing substantially in hardware, operating systems, protocols, firmware, and security mechanisms. Weak credentials, insecure remote-management interfaces, unpatched vulnerabilities, and limited monitoring capabilities can further increase exposure to automated compromise [62].

Mirai demonstrated the consequences of these weaknesses in 2016 by automatically scanning for vulnerable IoT devices and compromising systems using default or weak Telnet credentials [8]. Infected devices were subsequently coordinated for large-scale DDoS attacks. BASHLITE/Gafgyt had already targeted embedded Linux devices before Mirai [63], while later malware such as Torii demonstrated more persistent and multi-stage infection behavior [64].

The heterogeneity of IoT environments also complicates detection. A model trained on a limited set of devices or malware families may learn dataset-specific characteristics rather than general botnet behavior, making transfer across devices and environments difficult [65,66].

3.3. Challenges in botnet detection: traditional and modern approaches

Traditional detection mechanisms rely primarily on signatures, anomaly detection, behavioral analysis, and combinations of these approaches. Signature-based detection identifies known malicious indicators and can be efficient against established threats [67]. However, its dependence on previously identified patterns limits its effectiveness against unseen or modified botnets.

Anomaly-based detection instead identifies deviations from normal network or device behavior [68]. This provides greater potential for detecting unknown activity, but legitimate behavioral changes may also appear anomalous and generate false alarms. As discussed in Section 2, anomaly-based and behavior-based detection are conceptually distinct: anomaly detection identifies deviation from an expected baseline, whereas behavior-based detection explicitly searches for malicious actions or relationships such as scanning, propagation, or C&C activity.

Modern botnets further complicate detection through encrypted communication, decentralized control, rapidly changing infrastructure, and traffic patterns designed to resemble legitimate behavior [69]. These limitations have encouraged the adoption of more adaptive, data-driven detection mechanisms.

3.4. The role of artificial intelligence in botnet detection

AI-based approaches learn patterns of malicious activity from network traffic, device behavior, temporal sequences, communication relationships, and other security data. Classical ML models commonly use features such as flow duration, packet and byte statistics, timing, protocol information, and communication frequency to differentiate benign from malicious observations [70,71].

DL models extend this capability by learning nonlinear and hierarchical representations from high-dimensional or sequential traffic data [72]. CNNs can capture local relationships in structured traffic representations, whereas RNNs, LSTMs, and related architectures can model temporal dependencies [73,74]. More recent research also employs graph-based learning, hybrid architectures, and federated learning, which are analyzed in detail in Section 6.

Despite strong predictive results, AI-based detection remains sensitive to dataset quality and distribution shift. Models evaluated within a single benchmark may not retain their performance when exposed to different devices, malware families, or traffic environments [66]. Computational cost, interpretability, concept drift, and adversarial robustness also remain important considerations for real-world deployment.

3.5. Honey pots as a tool for data collection and analysis

Honey pots provide a complementary mechanism for collecting and analyzing botnet activity [75]. They intentionally expose monitored services or systems in order to attract malicious interactions, allowing researchers to record authentication attempts, commands, downloaded malware, network connections, and infection behavior [76].

Low-interaction honey pots emulate a restricted set of services and can be deployed comparatively safely at scale, whereas high-interaction honey pots provide richer behavioral information at greater operational risk [77]. Cowrie, for example, is an SSH/Telnet honey pot that records login attempts, commands, and file downloads associated with attacks against remotely accessible systems [78].

Honey pot observations can contribute valuable malicious samples and behavioral indicators for dataset construction and ML-based analysis [79]. However, they do not provide an unbiased representation of complete IoT traffic because the collected activity depends on the exposed services, network location, attacker population, and interaction level. Honey pots are therefore most useful when combined with controlled testbeds, public benchmarks, physical-device traffic, and complementary benign data sources.

4. Related surveys and comparative analysis

Research on IoT botnets has generated a substantial body of survey and review literature covering botnet architecture, attack lifecycle, detection methods, machine learning (ML), deep learning (DL), datasets, and, more recently, deployment constraints and trustworthy AI. The focus of this literature has evolved considerably. Earlier surveys primarily organized detection approaches according to botnet phases, attack mechanisms, or algorithmic families, whereas recent reviews increasingly consider dataset limitations, operational constraints, explainability, federated learning, adversarial behavior, and emerging AI paradigms. Consequently, the distinction between contemporary surveys is no longer determined simply by whether these topics are mentioned, but by how they are operationalized, compared, and integrated within a common analytical framework.

Wazzan et al. [80] provided a systematic analysis of IoT botnet detection approaches with particular emphasis on the phases of botnet formation and the malicious activities associated with those phases. Their phase-oriented organization remains useful for understanding where detection mechanisms operate within the botnet lifecycle. Nazir et al. [66] subsequently conducted a systematic review of ML- and DL-based IoT botnet detection, examining algorithms, benchmark datasets, data-preprocessing techniques, evaluation metrics, and methodological limitations. Their study established a more explicitly AI-centered view of the field and highlighted challenges associated with dataset diversity, real-time detection, heterogeneous IoT environments, and evolving attacks.

Gelgi et al. [81] examined IoT-botnet-driven distributed denial-of-service (DDoS) attacks through a systematic literature review. Their analysis covers botnet architecture, attack methodologies, and detection mechanisms, including ML- and DL-based solutions. Its deliberately DDoS-oriented scope provides substantial depth for this attack class but does not aim to represent the full range of IoT botnet detection and classification objectives.

The survey literature expanded considerably in 2025. Al-Shurbaji et al. [82] reviewed DL-based intrusion-detection systems for IoT botnet attacks, concentrating on architectures such as convolutional and recurrent neural networks and related deep models, together with their strengths, limitations, and research challenges. Lefoane et al. [83] examined AI-based IoT botnet detection within the broader IoT ecosystem, including enabling technologies and requirements associated with cloud computing, ML, and emerging 5G/6G communication environments. These works provide complementary perspectives: the former offers a technique-specific DL synthesis, whereas the latter places AI-based detection within a broader technological and architectural context.

Liu et al. [84] presented a large systematic literature review organized around the IoT botnet lifecycle. After quality assessment, 134 studies were reviewed, with particular attention to the features used for detection and their relationship to different lifecycle phases. The study also develops a malicious-activity map and identifies challenges including interpretability, false positives, and adaptation to evolving threats. Garg et al. [85] approached the problem from a broader botnet-ecosystem perspective, reviewing IoT botnet evolution, architecture, structural and behavioral characteristics, publicly available datasets, analysis tools, and intrusion-detection techniques.

Other recent reviews have moved closer to the operational and trustworthiness concerns considered in the present work. Bankó et al. [86] systematically analyzed ML-based intrusion detection for DDoS attacks in IoT networks, explicitly considering datasets, performance, deployment alternatives, and the implications of resource-constrained IoT environments. Although its scope is DDoS-oriented rather than specifically centered on IoT botnet classification, it is an important related study because deployment architecture and dataset choice are treated as explicit analytical dimensions. Mahboubi et al. [87] provided a broader adversarial-aware review of modern botnet detection, examining behavioral, flow-based, and graph-based approaches together with encrypted command-and-control communication, evasion strategies, and adversarial manipulation of ML/AI features. Its scope extends beyond IoT-specific botnets, but its adversarial perspective is directly relevant to the robustness of AI-driven IoT botnet detectors.

Two 2026 reviews are particularly close to the scope of the present survey. Wani et al. [88] comprehensively review AI-driven IoT botnet detection through botnet attack phases, conventional and AI-based detection methods, dataset and performance-metric comparisons, practical implementation challenges, and emerging directions. Their analysis extends beyond conventional model comparison

to discuss explainable AI (XAI), adversarial resilience, federated learning, resource overhead at the edge, concept drift, continuous learning, model compression, generative AI, and Digital Twin-assisted security. Sonune and Kulkarni [89] similarly emphasize operational limitations in IoT and mobile botnet detection, examining ML, DL, SDN-enabled, hybrid, and XAI-based approaches while explicitly considering latency, memory footprint, scalability, real-time operation, and robustness to evasion and obfuscation.

Table 5 summarizes the principal scope and analytical emphasis of these representative reviews. The comparison intentionally focuses on stable methodological characteristics rather than citation counts, which change over time and do not provide a reliable basis for establishing research novelty.

Table 5
Scope and analytical emphasis of representative surveys related to IoT botnet detection.

Survey	Year	Primary Scope	Principal Analytical Emphasis	Relationship to the Present Survey
[80]	2021	IoT botnet detection	Detection methods, botnet phases, and malicious-activity scenarios organized according to the botnet lifecycle.	Provides a lifecycle-oriented foundation; the present survey separately codes detection/-classification capability, dataset generalization, deployment evidence, and trustworthiness indicators at study level.
[66]	2023	ML/DL-based IoT botnet detection	ML/DL techniques, benchmark datasets, preprocessing, evaluation metrics, performance, and associated challenges.	Closely related AI-focused review; the present survey adds a unified coding framework for classification depth, cross-dataset evidence, deployment evidence, XAI, zero-day evaluation, and adversarial robustness.
[81]	2024	IoT botnet DDoS attacks	Botnet architecture, DDoS attack methodologies, and detection techniques including ML/DL-based solutions.	Provides greater depth on IoT-botnet-driven DDoS, whereas the present survey covers multiple botnet detection and classification objectives beyond DDoS.
[82]	2025	DL-based IoT botnet IDS	Deep-learning architectures for IoT botnet detection, methodological comparison, shortcomings, challenges, and future directions.	Offers detailed DL-specific coverage; the present survey compares DL with ML, hybrid, graph/GNN, DRL, and federated approaches under common analytical criteria.
[83]	2025	AI-based IoT botnet detection	AI detection approaches together with IoT ecosystem characteristics, cloud computing, ML, and emerging 5G/6G environments.	Provides a broad AI and ecosystem perspective; the present survey emphasizes reproducible study-level quantitative coding and operational evidence.
[86]	2025	ML-based IoT DDoS intrusion detection	ML techniques, dataset usage, performance comparison, and deployment/platform alternatives in resource-constrained IoT environments.	Particularly relevant to deployment analysis but DDoS-oriented; the present survey evaluates deployment evidence across a botnet-specific, multi-technique corpus and additionally considers classification granularity and adversarial robustness.
[84]	2025	Lifecycle-oriented IoT botnet detection	Detection features mapped to botnet lifecycle phases, detection methods, malicious activities, and limitations of current models.	Provides extensive lifecycle-oriented synthesis; the present survey operationalizes classification as separate non-cumulative binary, attack-type, malware-family, and lifecycle-stage capabilities.
[85]	2025	IoT botnet ecosystem and defense	Botnet evolution and architecture, structural/behavioral taxonomy, public datasets, analysis tools, and IDS techniques.	Offers broad architectural and defensive coverage; the present survey concentrates on quantitative comparison of empirical AI-based detection/classification studies and their generalization and deployment evidence.
[87]	2025	Adversarial-aware botnet detection	Behavioral, flow-based, and graph-based detection; encrypted C&C; evasion; and adversarial manipulation of ML/AI features.	Provides a strong adversarial perspective across the broader botnet domain; the present survey measures how frequently adversarial robustness is empirically evaluated within an IoT-botnet-specific corpus.
[88]	2026	AI-driven IoT botnet detection	Attack phases, AI techniques, datasets and metrics, practical implementation, edge-resource considerations, XAI, adversarial resilience, federated learning, concept drift, generative AI, and emerging directions.	The closest broad contemporary comparator. The present survey differs principally in its explicit study-level coding of classification depth, dataset breadth and cross-dataset validation, deployment evidence, zero-day evaluation, XAI, and adversarial testing across the same empirical corpus.
[89]	2026	IoT/mobile botnet detection	ML, DL, SDN, hybrid, and XAI methods with explicit consideration of latency, memory, scalability, real-time constraints, and evasion/obfuscation robustness.	Provides a deployment- and robustness-oriented review across IoT and mobile settings; the present survey couples these concerns with botnet-specific quantitative coding of classification and generalization evidence.
Present survey	2026	AI-based IoT botnet detection and classification	Quantitative study-level analysis of technique category, classification depth, dataset usage and breadth, cross-dataset validation, deployment evidence, evaluation practices, XAI, zero-day evaluation, and adversarial robustness.	Integrates methodological, classification, generalization, deployment, and trustworthiness evidence within one consistently coded corpus of 43 empirical studies.

The comparison in Table 5 shows that the recent literature already addresses many issues that were only sparsely treated in earlier surveys. Dataset limitations are examined explicitly in several reviews; lifecycle analysis is central to Wazzan et al. [80] and Liu et al. [84]; deployment constraints receive direct attention in Bankó et al. [86], Wani et al. [88], and Sonune and Kulkarni [89]; and adversarial robustness and explainability are increasingly visible in recent work, particularly in Mahboubi et al. [87] and Wani et al. [88]. Therefore, the remaining research gap should not be framed as the complete absence of these topics. Rather, the gap lies in the limited integration of these dimensions into a consistent evidence-coding framework that permits their relationships to be examined across the same set of primary studies.

The present survey addresses this integration problem through a structured quantitative analysis of 43 empirical AI-based IoT botnet studies. First, each study is assigned one mutually exclusive primary technique category—ML, DL, hybrid, graph/GNN, DRL, or federated learning—while secondary methodological components can still be discussed qualitatively. Second, classification capability is coded using four non-cumulative levels: binary botnet detection, attack-type classification, malware-family classification, and lifecycle-stage modeling. This separation avoids treating multiclass prediction as synonymous with malware-family or lifecycle-aware classification.

Third, dataset analysis separates three concepts that are frequently conflated: dataset popularity, dataset breadth, and explicit cross-dataset validation. The use of several datasets in independent experiments is therefore distinguished from genuine transfer or generalization evaluation across distinct data sources. Fourth, practical applicability is assessed through an evidence-based Low/Medium/High deployment rating derived from reported hardware testing, timing measurements, and resource or operational indicators rather than from the stated intent of the proposed model. This distinction is particularly important because a model described as lightweight or edge-oriented is not necessarily validated on constrained hardware.

Finally, the same 43-study corpus is audited for evaluation-metric reporting, explicit XAI, zero-day evaluation, and empirical adversarial robustness. This unified design makes it possible to examine whether increasingly sophisticated AI techniques are accompanied by deeper classification, broader validation, stronger operational evidence, and greater attention to trustworthy and adversarially robust deployment. The contribution of the present survey is therefore not simply broader topical coverage, but the systematic operationalization of these dimensions within a common study-level analytical framework.

5. Methodology for paper selection

This survey follows a structured methodology for identifying and analyzing recent studies on artificial intelligence (AI)-driven IoT botnet detection and classification. The review is designed to support both qualitative synthesis and quantitative cross-study analysis across methodological techniques, classification depth, dataset usage, generalization, evaluation practices, deployment evidence, explainability, and adversarial robustness.

The scope of the quantitative analysis is limited to AI-driven IoT botnet detection and classification approaches. Traditional signature-based, rule-based, anomaly-based, and other non-learning techniques are discussed in the background sections where relevant, but the primary study corpus focuses on learning-based detection and classification methods.

5.1. Search strategy and data sources

Relevant studies published between 2020 and August 2026 were identified through searches of major academic sources, including IEEE Xplore, ACM Digital Library, SpringerLink, Elsevier ScienceDirect, Scopus, and Google Scholar. No single database or bibliographic index was used as an exclusive eligibility filter; studies identified across the listed sources were assessed using the common inclusion and exclusion criteria described below. The search vocabulary was organized around three main concepts: IoT environments, botnet detection or classification, and AI-based detection techniques.

Representative search terms included “IoT botnet detection”, “IIoT botnet detection”, “machine learning botnet detection”, “deep learning botnet classification”, “graph neural network IoT botnet”, “federated learning IoT botnet”, “reinforcement learning IoT botnet”, “IoT botnet adversarial robustness”, and “explainable AI botnet detection”. A representative Boolean formulation was:

(IoT OR IIoT OR “Internet of Things”) AND (botnet OR “botnet detection” OR “botnet classification”) AND (“machine learning” OR “deep learning” OR “graph neural network” OR “federated learning” OR “reinforcement learning” OR ensemble OR transformer OR “adversarial robustness” OR “explainable AI”)

The search was complemented by reference-based exploration of relevant studies to improve coverage of closely related work. The resulting publications were screened according to the criteria described below.

5.2. Inclusion and exclusion criteria

To maintain a consistent focus on empirically evaluated AI-based IoT botnet research, the following criteria were applied.

Inclusion Criteria:

- Studies published between 2020 and August 2026.
- Studies focusing on IoT or Industrial IoT (IIoT) botnet detection, classification, behavior analysis, or related botnet-defense tasks.
- Studies using ML, DL, hybrid or ensemble learning, graph-based learning, deep reinforcement learning (DRL), federated learning, or closely related AI techniques.

- Studies providing empirical evaluation using benchmark, experimental, honeypot, testbed, or real-world data.
- Peer-reviewed journal or conference publications.

Exclusion Criteria:

- Purely theoretical or conceptual studies without empirical validation.
- Studies focusing on general network intrusion detection without a clear IoT-botnet component.
- Studies based exclusively on non-learning detection mechanisms.
- Studies whose experimental setting or reported results were insufficient for the analytical dimensions considered in this survey.

5.3. Selection process

The study-selection process consisted of identification, title-and-abstract screening, full-text assessment, and final inclusion. Where necessary, the representative search formulation described in [Section 5.1](#) was adapted to the syntax of the individual academic source without changing its conceptual scope. Candidate studies identified through either database searching or reference-based exploration were subjected to the same screening procedure. Full-text eligibility was assessed using the inclusion and exclusion criteria defined in [Section 5.2](#), and only studies satisfying the defined scope and empirical-evaluation requirements were retained. The literature search and eligibility assessment were finalized in August 2026.

Application of these criteria resulted in a corpus of **43 primary studies**, which forms the basis of the quantitative analysis presented in the subsequent sections. The overall selection workflow is summarized in [Fig. 6](#).

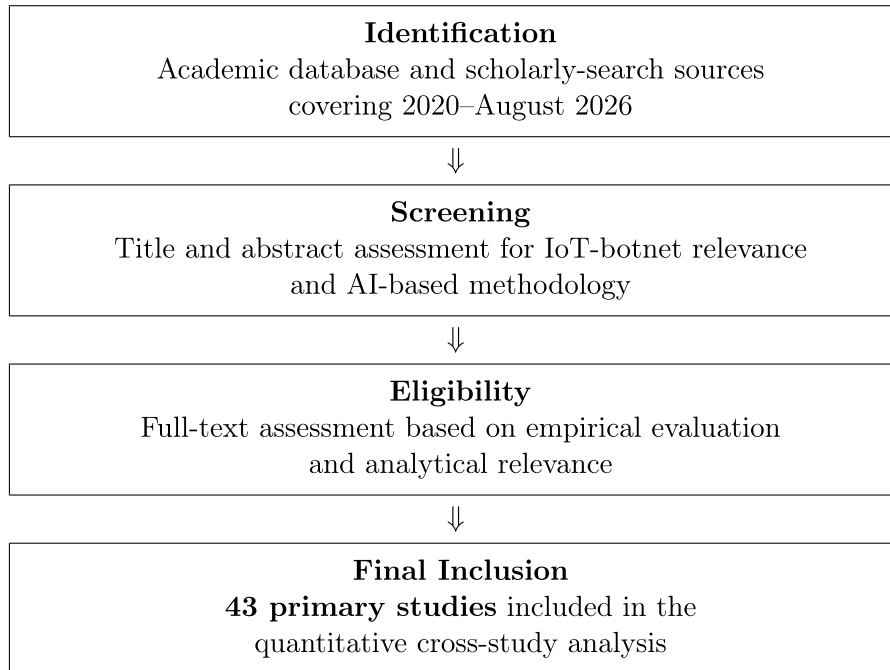


Fig. 6. Paper-selection workflow used in this survey.

5.4. Characteristics of the selected study corpus

The selected studies span the period from 2020 to 2026 and represent a range of methodological directions in AI-based IoT botnet detection. The temporal distribution comprises 2 studies from 2020, 4 from 2021, 5 from 2022, 8 from 2023, 6 from 2024, 11 from 2025, and 7 from 2026.

The corpus includes studies published across journals and conferences in cybersecurity, IoT, networking, and artificial intelligence. Recent studies increasingly explore graph-based learning, federated learning, explainable models, lightweight architectures, edge-oriented detection, and adversarially robust learning in addition to conventional ML and DL approaches.

[Table 6](#) summarizes the selected studies and their primary methodological categories.

Table 6
Details of the 43 selected primary studies.

ID	Reference	Year	Venue	Primary Technique
1	[90]	2025	International Journal of Information Technology	Hybrid
2	[91]	2020	IEEE Internet of Things Journal	DL
3	[92]	2023	Computers & Security	DL
4	[93]	2023	IEEE Internet of Things Journal	DRL
5	[94]	2020	International Computer Symposium (ICS), IEEE	DL
6	[75]	2022	Computers & Electrical Engineering	Graph/GNN
7	[95]	2022	IEEE Access	ML
8	[96]	2023	IEEE Access	Hybrid
9	[97]	2022	Computers, Materials & Continua	Hybrid
10	[98]	2025	Expert Systems with Applications	DL
11	[65]	2023	Applied Sciences	DL
12	[99]	2021	IEEE Access	Graph/GNN
13	[100]	2025	Journal of Information Security and Applications	Hybrid
14	[101]	2021	Sensors	DL
15	[102]	2022	IEEE Access	DL
16	[103]	2024	Alexandria Engineering Journal	Hybrid
17	[104]	2023	Scientific Reports	ML
18	[105]	2021	Journal of Ambient Intelligence and Humanized Computing	Hybrid
19	[106]	2023	Sensors	Federated
20	[107]	2024	Cognitive Models and Artificial Intelligence Conference	Hybrid
21	[108]	2023	IEEE International Conference on Business Analytics for Technology and Security (ICBATS)	Hybrid
22	[109]	2023	Global Conference on Information Technologies and Communications (GCITC)	DL
23	[110]	2024	International Conference for Innovation in Technology (INOCON)	DL
24	[111]	2022	Computers & Security	ML
25	[112]	2025	Intelligent Systems with Applications	Federated
26	[113]	2024	Electronics	Graph/GNN
27	[114]	2024	Journal of Advances in Information Technology	Graph/GNN
28	[115]	2025	Scientific Reports	Hybrid
29	[116]	2026	Future Generation Computer Systems	Graph/GNN
30	[117]	2026	Cybersecurity	Hybrid
31	[118]	2026	Scientific Reports	Federated
32	[119]	2026	Computers	Hybrid
33	[120]	2026	Journal of The Institution of Engineers (India): Series B	Hybrid
34	[121]	2026	Memetic Computing	ML
35	[122]	2026	Applied Sciences	Federated
36	[123]	2025	Scientific Reports	ML
37	[124]	2025	IEEE Internet of Things Journal	Graph/GNN
38	[125]	2025	ICCK Transactions on Machine Intelligence	Federated
39	[126]	2025	Computer Communications	Hybrid
40	[127]	2025	IEEE Transactions on Network Science and Engineering	Hybrid
41	[128]	2024	SN Computer Science	DL
42	[129]	2021	International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE	Graph/GNN
43	[130]	2025	Sādhanā	Federated

5.5. Data extraction and coding framework

A structured study-level extraction framework was used to support consistent cross-study comparison. Information was recorded for each study regarding the proposed technique, datasets, classification objectives, evaluation metrics, generalization, deployment characteristics, interpretability, and robustness.

For each included study, information was extracted from the full text using the same predefined coding template. Evidence was obtained from the methodology, experimental setup, reported results, tables, figures, and implementation details where applicable. Before aggregation, the extracted fields were cross-checked against the corresponding source evidence, and dataset names and recurring categorical terminology were normalized to support consistent counting across studies. Information that was not explicitly reported was not inferred; it was recorded as NR, while partial or indirect evidence was recorded as Partial according to the definitions below. Where a value was derived from other reported results rather than stated directly by the original study, it was distinguished from directly reported evidence.

Categorical analytical fields were represented using the following standardized values:

- **Y**: the characteristic was explicitly reported, tested, or demonstrated;
- **N**: the characteristic was not part of the reported evaluation;
- **Partial**: the characteristic was addressed only partially or indirectly;

- **NR**: the relevant information was not reported.

The distinction between *N* and *NR* prevents missing information from being interpreted as evidence of poor model performance or poor deployment suitability.

5.5.1. Primary technique categorization

Each study was assigned to one primary technique category according to its dominant methodological contribution:

- **ML**: classical machine-learning approaches such as Random Forest, Support Vector Machine, Decision Tree, and k-nearest neighbors;
- **DL**: deep neural architectures such as CNNs, RNNs, LSTMs, GRUs, autoencoders, DNNs, and transformer-based models;
- **Hybrid**: approaches combining multiple learning models, ensemble methods, optimization procedures, or multi-stage architectures;
- **Graph/GNN**: approaches in which graph representations, graph analytics, or graph neural networks form the principal modeling mechanism;
- **Federated**: approaches in which federated or distributed privacy-preserving learning forms the principal learning framework;
- **DRL**: approaches based primarily on deep reinforcement learning.

The use of a single primary category avoids double counting in the technique-distribution analysis, while secondary methodological components are retained in the qualitative discussion.

5.5.2. Classification-depth coding

Classification capability was represented using four analytical levels:

- **Level 1**: binary botnet-versus-benign detection;
- **Level 2**: attack-type classification;
- **Level 3**: malware-family classification;
- **Level 4**: lifecycle-stage detection or modeling.

The levels are not treated as cumulative. A study is assigned only the levels corresponding to classification tasks that are explicitly evaluated. For example, a study may perform binary detection and malware-family classification without performing attack-type classification.

Anomaly-based and behavior-based approaches are also distinguished conceptually. Anomaly-based detection identifies deviations from a learned or statistically defined representation of normal activity, whereas behavior-based detection explicitly models malicious actions, communication relationships, temporal sequences, coordinated activity, or attack stages.

5.5.3. Dataset breadth and generalization

Dataset names were normalized to ensure that naming and spelling variants of the same benchmark were treated consistently. Since individual studies may employ several datasets, dataset-frequency counts are non-mutually exclusive.

Dataset breadth was categorized as follows:

- **Low**: evaluation using one distinct dataset or data source without cross-dataset validation;
- **Medium**: evaluation using two distinct datasets or data sources without cross-dataset validation;
- **High**: evaluation using three or more distinct datasets/data sources, or explicit cross-dataset validation.

Cross-dataset validation was recorded only when a model was explicitly evaluated for transfer or generalization across different datasets. Independent experiments conducted separately on multiple datasets were not treated as cross-dataset validation.

Dataset breadth is analyzed separately from dataset characteristics such as traffic diversity, attack coverage, labeling, class imbalance, encrypted traffic, and support for multiclass or unseen-attack evaluation.

5.5.4. Deployment-evidence assessment

Deployment feasibility was assessed using operational evidence reported in each study. The extracted indicators include real-time evaluation, inference time, training time, memory requirements, computational complexity, model size, physical edge-device testing, SDN integration, cloud dependency, energy consumption, communication overhead, and scalability.

The strength of reported deployment evidence was categorized as:

- **High**: physical testing on an edge, gateway, embedded, or comparable constrained platform, together with timing evidence and at least one resource or operational indicator;
- **Medium**: at least one explicit runtime, resource, or deployment-oriented indicator is reported, but the evidence does not satisfy all conditions required for the High category;
- **Low**: little or no explicit operational deployment evidence is reported.

The rating reflects the *reported deployment evidence* rather than an independent judgment of whether a model could theoretically be deployed. Training time is treated separately from inference efficiency because computationally intensive offline training does not necessarily imply expensive runtime inference. Similarly, SDN integration and cloud dependency are considered architectural characteristics rather than direct indicators of deployment quality.

5.5.5. Evaluation, explainability, and robustness coding

Performance information was extracted for commonly reported metrics including accuracy, precision, recall, F1-score, false-positive or false-alarm rate, and ROC-AUC. Deployment-related measurements, such as inference time, training time, memory requirements, computational complexity, energy consumption, model size, communication overhead, and scalability, were analyzed separately from predictive performance.

Explainability was distinguished between *explicit XAI* and broader model interpretability. Explicit XAI refers to the implementation of formal explanation techniques such as SHAP, LIME, Anchors, or rule-extraction mechanisms. Interpretability refers to evidence such as feature importance, structural visualization, attention analysis, or inherently interpretable representations without a dedicated XAI framework.

Adversarial robustness was recorded when a study empirically evaluated the detector against adversarial machine-learning attacks or perturbations, including evasion, poisoning, backdoor, gradient-based, or adversarial model-update attacks.

These operational definitions form the basis of the quantitative synthesis presented in the following sections.

5.6. Scope and validity considerations

Several considerations affect the interpretation of cross-study results. The survey focuses specifically on AI-driven IoT botnet research; consequently, the reported technique distributions characterize this research area rather than the complete spectrum of botnet-detection mechanisms.

Experimental settings also vary considerably across the reviewed studies in terms of datasets, preprocessing procedures, class distributions, hardware platforms, and validation protocols. Direct comparison of reported accuracy or F1-score values across studies should therefore be interpreted cautiously. The quantitative synthesis primarily examines methodological trends, evaluation practices, classification capabilities, dataset usage, generalization, and deployment evidence.

In addition, absence of a reported characteristic is treated as missing evidence rather than as evidence of poor performance. This distinction is particularly important for deployment indicators such as memory usage, energy consumption, latency, and communication overhead.

Accordingly, the reported distributions should be interpreted within the defined search sources, search vocabulary, temporal scope, and eligibility criteria, as alternative retrieval strategies may produce somewhat different absolute study distributions.

6. State-of-the-art AI-based IoT botnet detection and classification

AI-based IoT botnet detection has progressed from conventional feature-driven classifiers toward deep, hybrid, graph-based, federated, and deployment-oriented learning frameworks. In parallel, the objectives of detection systems have expanded beyond binary benign-versus-malicious decisions to include attack-type classification, malware-family identification, behavioral modeling, zero-day detection, and robustness against adversarial manipulation.

Despite this methodological diversity, three concerns recur across the literature. First, strong predictive performance is frequently demonstrated on controlled benchmark datasets, while cross-dataset and cross-device generalization remain more difficult to establish. Second, increasing model complexity may improve representation learning but also introduces computational, memory, latency, and interpretability constraints. Third, deployment-oriented evidence remains less consistently reported than conventional classification metrics. The following subsections examine these methodological directions qualitatively, while their quantitative distribution across the reviewed studies is analyzed separately in [Section 7](#).

6.1. Feature-driven machine learning approaches

Classical machine-learning (ML) methods remain relevant for IoT botnet detection because they can operate effectively on compact flow-level or statistical feature representations. Common classifiers include Random Forests, Decision Trees, Support Vector Machines, Logistic Regression, k-nearest neighbors, and related ensemble methods [95,104,111,121,123]. Compared with deeper architectures, these models often require fewer computational resources during inference and can provide clearer relationships between input features and classification decisions.

Feature engineering and feature selection are therefore central to their effectiveness. Kalakoti et al. [95] demonstrated the importance of identifying compact subsets of discriminative traffic features, while Hossain and Islam [104] combined feature selection with ensemble learning to reduce redundant information while maintaining strong detection performance. Such approaches are particularly relevant to gateway-level monitoring, where the computational cost of feature extraction and inference must remain bounded.

The main limitation of feature-driven ML is its dependence on the representation used during training. Features that distinguish benign and malicious traffic in one dataset may encode device-, capture-, or environment-specific characteristics that do not transfer reliably to another network. Consequently, lightweight inference alone does not guarantee robust deployment. Cross-dataset validation, device-independent evaluation, and resistance to changing traffic distributions remain important requirements.

Interpretability is a relative advantage of many classical models, particularly when feature rankings, rule-based explanations, or post-hoc explanation mechanisms are available. Recent work has increasingly combined conventional detection models with explicit explainability mechanisms to improve analyst understanding of model decisions [123].

6.2. Imbalance-aware and multiclass detection

Class imbalance is a persistent challenge in IoT botnet datasets because benign traffic and high-volume attack categories may substantially outnumber less frequent attack behaviors. This becomes particularly problematic for multiclass detection, where high aggregate accuracy may conceal poor recall for minority classes.

Several studies address this problem using resampling or cost-sensitive strategies, including SMOTE, class weighting, undersampling, and related imbalance-handling methods [99,101,102,105]. These approaches can improve recognition of minority attack classes and reduce bias toward dominant traffic categories.

However, balancing techniques introduce methodological trade-offs. Synthetic oversampling modifies the original training distribution and may generate samples that do not fully represent realistic network behavior. Likewise, a model evaluated on an artificially balanced test distribution may not reflect operational conditions in which malicious traffic is comparatively rare. For this reason, imbalance-aware evaluation should be accompanied by per-class precision, recall, F1-score, confusion matrices, and false-alarm measurements rather than relying on accuracy alone.

Multiclass analysis also has practical importance beyond improving metric reporting. Distinguishing attack types or malware families provides richer information for incident response than binary detection, although finer-grained classification generally requires more representative training data and introduces greater sensitivity to class imbalance and dataset-specific labeling conventions.

6.3. Deep learning-based approaches

Deep learning (DL) techniques have been widely adopted because of their capacity to learn nonlinear representations from raw, statistical, sequential, or transformed network data [65,91,92,94,98,101,109,128]. Common architectures include deep neural networks, convolutional neural networks, recurrent neural networks, LSTMs, GRUs, autoencoders, variational encoders, and transformer-based models.

Autoencoder-based methods learn compact representations of normal or mixed traffic and can identify deviations through reconstruction error. CNN-based architectures are suitable for extracting local patterns from structured traffic representations, while RNNs and LSTMs are designed to capture sequential dependencies that may reflect evolving communication behavior [94,101]. More recent transformer-based approaches exploit attention mechanisms to capture longer-range dependencies within traffic representations [109].

A major advantage of DL is reduced reliance on manually selected feature combinations. Nevertheless, this advantage comes with additional training cost, memory demand, architectural complexity, and reduced transparency. The practical cost also depends strongly on where processing occurs. A model trained using high-performance GPU hardware may still be deployable if inference is sufficiently lightweight, whereas a comparatively compact network may remain unsuitable if it requires expensive preprocessing or long observation windows.

Consequently, reported classifier inference time should not be interpreted in isolation. End-to-end detection delay may additionally include packet capture, flow aggregation, feature generation, sequence construction, and decision-window duration. This distinction is particularly important when evaluating claims of real-time detection.

Deep models are also vulnerable to dataset shift. High performance within one benchmark does not necessarily demonstrate transfer to unseen devices, botnet families, capture environments, or encrypted traffic. Cross-device and cross-dataset evaluation therefore provide stronger evidence of generalization than random train-test partitions taken from a single source.

6.4. Hybrid and optimization-based models

Hybrid approaches combine multiple learning stages or methodological paradigms in order to exploit complementary strengths. Examples include feature selection followed by classification, deep feature extraction followed by conventional classifiers, multi-stage neural architectures, ensemble learning, and optimization-based parameter selection [90,96,97,100,103,115,119,120,126,127].

Such approaches are attractive because different components can address different aspects of the detection problem. Dimensionality reduction can decrease redundant input information, optimization methods can improve model configuration, and ensembles can reduce dependence on a single classifier. Hybrid architectures may also combine spatial, temporal, or statistical representations.

The primary challenge is that each additional processing stage may introduce computational and implementation overhead. A system that performs feature selection, transformation, optimization, and multiple classifier stages can achieve strong benchmark performance while becoming harder to maintain or deploy under constrained conditions. The relevant question is therefore not simply whether the hybrid architecture improves accuracy, but whether the additional performance gain justifies its resource cost.

Deployment-oriented assessment of hybrid models should consequently include end-to-end inference latency, preprocessing cost, model size, memory footprint, and scalability in addition to conventional classification metrics.

6.5. Graph-based and behavior-centric approaches

Graph-based methods provide an alternative to treating traffic observations as independent feature vectors. They model relationships among devices, flows, hosts, executable components, or communication events, allowing the detector to capture structural and coordinated characteristics of botnet activity [99,113,114,116,124,129].

In graph-based traffic analysis, nodes and edges can represent communicating entities and their interactions, while graph neural networks learn representations from both node attributes and neighborhood structure. Dynamic graph approaches extend this representation by considering how relationships evolve over time [124]. Other studies use graph-derived representations for static malware analysis rather than network-flow analysis, demonstrating that structural information can also be extracted from IoT executable behavior [129].

These approaches are particularly relevant to coordinated botnet activity because individual flows may appear benign in isolation while their relationships reveal scanning, propagation, command-and-control, or synchronized attack behavior. They therefore offer a natural basis for behavior-centric and sequential analysis.

However, graph methods introduce additional deployment challenges. Constructing and maintaining graphs requires state, memory, and preprocessing, while message-passing operations may become costly as graph size and connectivity increase. Real-time applicability therefore depends not only on classifier complexity but also on graph construction, update frequency, neighborhood sampling, and the scale of the monitored IoT network.

6.6. Federated, edge, and deployment-oriented approaches

Deployment-oriented research attempts to reduce the gap between offline benchmark evaluation and operational IoT security. Relevant approaches include edge inference, gateway-level monitoring, SDN-assisted detection, lightweight neural models, and federated learning [91,92,106,111,112,118,122,125,130].

Edge-based detection processes traffic close to the monitored devices rather than transmitting all observations to centralized infrastructure. This can reduce network delay and bandwidth requirements, but the detector must operate within the processor, memory, storage, and energy constraints of the target platform. Consequently, a meaningful deployment evaluation should identify the hardware used and report operational measurements such as inference latency, throughput, memory usage, model size, CPU utilization, or energy consumption.

Several studies specifically design lightweight models for edge or gateway operation [91,92,111]. Physical evaluation is particularly informative because workstation-based inference measurements do not necessarily represent performance on embedded hardware. For example, ARAFL-BAD evaluates its federated framework using Raspberry Pi edge devices together with resource and timing measurements [130], providing stronger deployment evidence than architecture-level claims alone.

Federated learning addresses a different deployment constraint: the distribution and privacy of IoT data. Rather than transmitting raw traffic to a central training repository, participating devices or gateways train local models and exchange model updates [106,112,125]. This architecture can improve data locality, but it introduces communication overhead, synchronization challenges, client heterogeneity, non-independent and identically distributed data, and vulnerability to malicious clients.

Training and inference costs should also be separated when evaluating federated systems. Expensive distributed training may still be compatible with lightweight local inference, whereas frequent retraining or large model-update exchanges may make a theoretically decentralized solution costly in practice. Communication volume, number of rounds, client participation, update size, and robustness to client failure are therefore important deployment indicators alongside predictive accuracy.

6.7. Adversarial robustness and explainability

As learning-based security systems become more capable, the assumption that test inputs remain independent of the detector becomes increasingly restrictive. Adversarial machine-learning attacks deliberately manipulate inputs, training data, or distributed model updates in order to degrade detection performance.

Recent IoT botnet studies have begun to evaluate this problem explicitly. Krishnan and Shrinath [128] examine gradient-based adversarial attacks, including FGSM, PGD, and BPDA, and employ a denoising autoencoder as an adversarial defense. Ngo et al. [129] analyze adversarial structural modifications against graph-based IoT botnet detection and investigate adversarial training as a defensive mechanism. In federated settings, ARAFL-BAD considers both adversarial examples and malicious-training behavior while combining asynchronous learning with zero-day botnet detection [130].

These studies show that robustness under conventional test data and robustness under adaptive attack are distinct properties. High clean-data accuracy does not establish resilience against evasion, poisoning, backdoor attacks, or malicious federated clients. Adversarial evaluation should therefore report both clean and attacked performance and should specify the threat model, attacker capability, perturbation constraints, and defensive assumptions.

Explainability addresses a related but distinct requirement: understanding why a model produces a security decision. Recent studies incorporate formal explanation methods such as SHAP, LIME, Anchors, and rule-extraction mechanisms [117,118,123,126]. Such mechanisms can support feature-level interpretation, false-alarm investigation, and analyst validation.

Formal explainable AI should, however, be distinguished from general model interpretability. Feature importance, attention visualization, graph structure, or an inherently transparent classifier can assist interpretation without constituting a dedicated XAI framework. For operational IoT security, explanation quality must ultimately be evaluated in terms of stability, fidelity, usefulness, and computational overhead rather than visualization alone.

6.8. Emerging AI directions

Several AI paradigms that are increasingly relevant to cybersecurity remain comparatively immature in IoT botnet detection and are not yet strongly represented in the primary study corpus considered in this survey.

Continual and online learning are relevant [88] because IoT traffic distributions, device populations, and botnet behavior change over time. A detector capable of controlled model adaptation could address concept drift more effectively than a static model trained

once on a fixed benchmark. Such systems must nevertheless address catastrophic forgetting, unsafe updates, and the risk that malicious observations contaminate continual retraining.

Self-supervised learning provides another promising direction because large volumes of IoT traffic can be collected without reliable labels. Pretraining representations from unlabeled traffic and subsequently fine-tuning them for botnet detection could reduce dependence on manually labeled attack data and improve transfer to new environments.

Foundation models and large language models (LLMs) [131] should currently be considered complementary to, rather than replacements for, packet- or flow-level detection models. Their more immediate potential lies in analyzing security logs, correlating alerts, extracting threat intelligence, explaining detection outputs, and supporting analyst-facing investigation workflows.

Similarly, *retrieval-augmented generation (RAG)* [132] can ground language-model outputs in curated threat reports, malware descriptions, vulnerability databases, or local security documentation. Such systems may help contextualize detector alerts, but their reliability depends on retrieval quality, evidence provenance, latency, and resistance to malicious or outdated knowledge sources.

These paradigms therefore represent emerging research opportunities rather than established replacements for the detection architectures reviewed in the preceding subsections.

6.9. Comparative technique-level assessment

Table 7 summarizes the typical computational and deployment characteristics of the major methodological categories. The entries describe broad tendencies observed across the reviewed literature rather than fixed properties of every individual implementation; actual resource requirements depend on architecture size, preprocessing, hardware, traffic rate, and deployment configuration.

No single methodological family simultaneously optimizes predictive performance, interpretability, computational efficiency, behavioral depth, privacy, robustness, and deployment cost. Classical ML remains attractive for lightweight inference, whereas DL provides stronger representation learning at greater computational cost. Hybrid models trade additional complexity for complementary modeling capability. Graph-based approaches provide richer structural and behavioral representations, while federated learning addresses data locality but introduces communication and distributed-security challenges.

Accordingly, technique selection should be driven by the intended operating environment rather than benchmark accuracy alone. A cloud-based forensic classifier, an IoT gateway detector, and a continuously operating embedded detector impose fundamentally different computational and latency requirements.

6.10. Summary of key trends

The state-of-the-art literature demonstrates a clear movement from isolated traffic classification toward richer and more deployment-aware learning frameworks. Classical ML continues to provide efficient feature-driven detection, while DL and hybrid approaches offer stronger representation capacity at higher computational cost. Graph-based methods provide a means of modeling coordinated and sequential botnet behavior, and federated learning supports distributed training where raw traffic cannot be centrally collected.

At the same time, several limitations cut across technique categories. Generalization is frequently constrained by dependence on benchmark-specific data; deployment claims are not always accompanied by hardware, latency, memory, energy, or communication measurements; and fine-grained behavioral or lifecycle classification remains more difficult than conventional detection. Explainability and adversarial robustness are receiving increasing attention but are not yet routine components of evaluation.

These observations motivate the quantitative cross-study analysis in the following section, where technique distribution, classification depth, dataset dependence, generalization, evaluation practices, deployment evidence, explainability, and adversarial robustness are examined systematically across the selected studies.

7. Quantitative cross-study analysis

This section provides a quantitative synthesis of the 43 primary studies included in the survey. While **Section 6** examined the methodological characteristics of the reviewed approaches, the present section evaluates recurring patterns across the study corpus. The analysis covers primary technique distribution, dataset usage, classification depth, deployment evidence, evaluation-metric reporting, dataset breadth and generalization, and emerging trustworthiness dimensions including explainability, zero-day evaluation, and adversarial robustness.

Unless otherwise stated, percentages are computed relative to the complete corpus of 43 studies:

$$\text{Percentage} = \frac{\text{Number of Studies in Category}}{43} \times 100. \quad (1)$$

The primary technique categories are mutually exclusive because each study is assigned one dominant methodological category according to the coding framework defined in **Section 5.5**. In contrast, classification-depth levels, dataset usage, deployment indicators, and robustness characteristics are non-mutually exclusive; therefore, their percentages may sum to more than 100%. For evidence-reporting variables, only characteristics coded as explicitly reported or demonstrated (Y) are included in the principal counts, while partial evidence is discussed separately where relevant.

Table 7
Qualitative comparison of major AI Technique categories for IoT botnet detection.

Technique	Computational Complexity	Interpretability	Training / Inference Characteristics	Memory / Communication Requirements	Scalability	Deployment Suitability
Classical ML	Usually low-moderate after feature extraction	Moderate-high for trees, rules, and feature-based models	Training and inference are generally lightweight relative to deep architectures	Usually modest model memory; feature extraction may dominate runtime cost	Generally good when feature dimensionality remains controlled	Well suited to gateways and edge systems when features can be extracted efficiently
Deep Learning	Moderate-high; strongly architecture-dependent	Generally low without dedicated explanation mechanisms	Training is often expensive; inference ranges from lightweight to hardware-intensive	Model and intermediate-tensor memory can be substantial	Good with suitable acceleration, but hardware dependent	Suitable when compressed, lightweight, or accelerated; otherwise often gateway/server oriented
Hybrid / Ensemble	Moderate-high because several processing or learning stages are combined	Low-moderate, depending on constituent models	Training and inference may require several sequential stages	Memory increases with the number of models, transformations, or optimization stages	Highly dependent	Potentially effective but requires explicit end-to-end runtime and resource evaluation
Graph / GNN	Moderate-high due to graph construction and message passing	Moderate because structural relationships can aid explanation	Both graph construction and inference contribute to latency	Memory requirement increases with graph size, connectivity, and retained state	Can become challenging for large dynamic graphs	Promising for gateway/server deployment; constrained-device use requires graph and model optimization
DRL	High during training because of iterative interaction and policy optimization	Generally moderate	Training is expensive; learned-policy inference can be considerably lighter	Depends on state representation and policy-network size	Sensitive to state/action-space growth	Potentially useful for adaptive response and lifecycle-aware decisions, but deployment evidence remains limited
Federated Learning	Local model cost plus repeated distributed aggregation	Inherited from the local model unless XAI is added	Training is communication-intensive; inference can remain local	Communication overhead, update buffers, and local-model memory are important	Potentially strong, but affected by client heterogeneity and non-IID data	Well aligned with distributed IoT when communication and client-resource costs are controlled

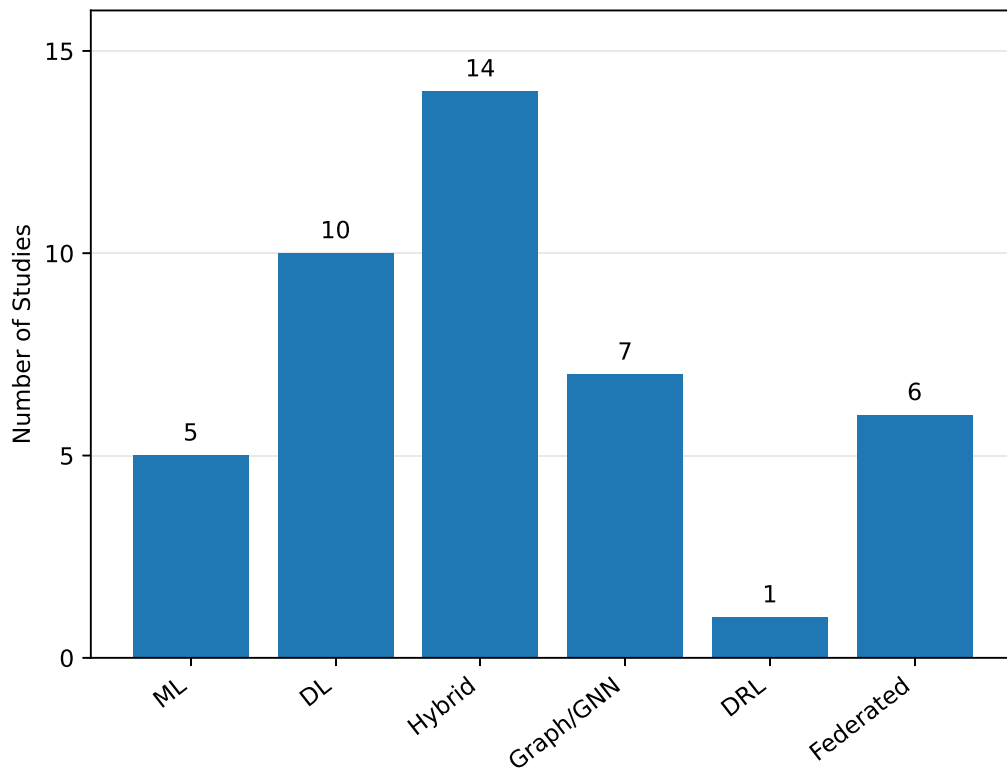


Fig. 7. Distribution of primary AI technique categories across the 43 reviewed studies.

7.1. Distribution of primary AI techniques

Table 8 presents the mutually exclusive distribution of primary methodological categories across the 43 studies. Hybrid approaches constitute the largest category, with 14 studies (32.6%), followed by deep learning with 10 studies (23.3%). Graph/GNN-based approaches account for 7 studies (16.3%), federated learning for 6 studies (14.0%), classical ML for 5 studies (11.6%), and DRL for 1 study (2.3%).

Table 8

Distribution of primary technique categories across the 43 reviewed studies.

Primary Technique Category	No. of Studies	Percentage
Classical Machine Learning	5	11.6%
Deep Learning	10	23.3%
Hybrid / Ensemble / Optimization-Based	14	32.6%
Graph-Based / GNN-Based	7	16.3%
Deep Reinforcement Learning	1	2.3%
Federated Learning	6	14.0%

Hybrid and DL approaches together represent 24 of the 43 studies (55.8%), showing the continued importance of representation learning and multi-stage architectures. However, the distribution also demonstrates increasing methodological diversity. Graph/GNN and federated approaches together account for 13 studies (30.2%), reflecting growing interest in relational traffic modeling and distributed privacy-preserving learning.

Fig. 7 visualizes the primary technique distribution.

The predominance of hybrid and DL methods does not by itself establish greater practical suitability. More complex architectures may improve representation capacity while simultaneously increasing preprocessing, memory, and computational requirements. Technique prevalence must therefore be interpreted together with the deployment analysis presented later in this section.

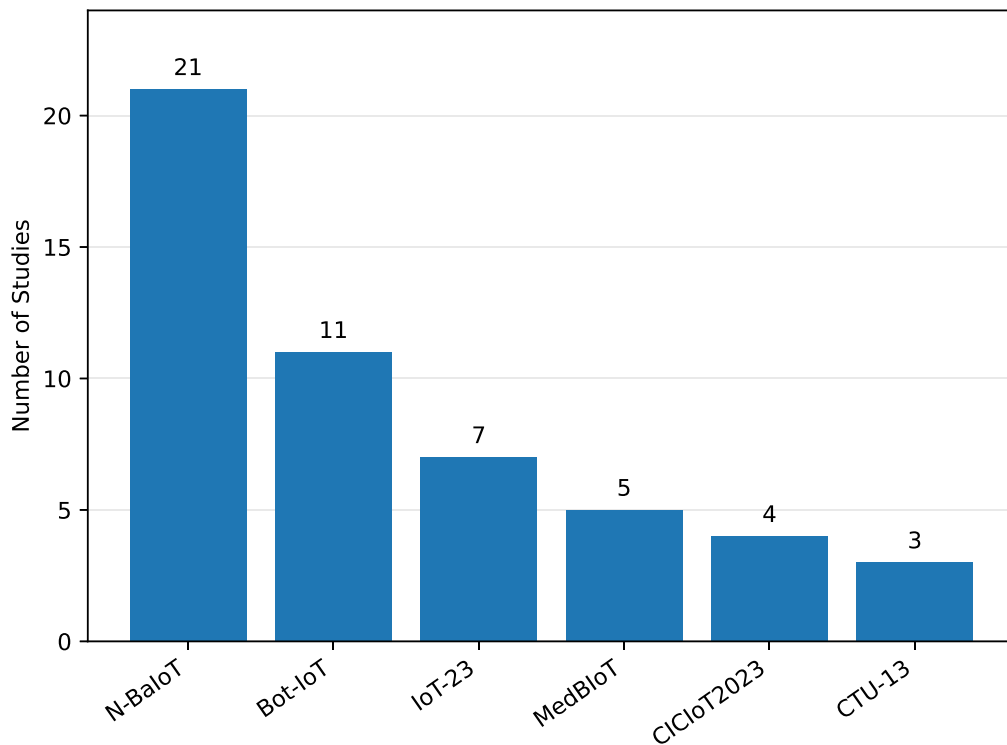


Fig. 8. Frequency of the most commonly used datasets across the reviewed studies. Counts are non-exclusive because individual studies may use multiple datasets.

7.2. Dataset usage patterns

Dataset selection strongly influences the conclusions that can be drawn from IoT botnet detection experiments. [Table 9](#) presents the most frequently used normalized datasets in the reviewed corpus. Dataset counts are non-exclusive because individual studies may evaluate more than one benchmark.

Table 9
Most frequently used datasets across the reviewed studies.

Dataset	No. of Studies	Percentage	Role in the Reviewed Literature
N-BaIoT	21	48.8%	The most frequently used benchmark in the corpus; widely used for IoT-specific Mirai and BASHLITE detection and classification.
Bot-IoT	11	25.6%	Frequently used for IoT attack detection, multiclass evaluation, and comparative ML/DL experiments.
IoT-23	7	16.3%	Used for malicious IoT traffic analysis, multi-dataset experiments, and behavior-oriented evaluation.
MedBioT	5	11.6%	Used for evaluating IoT botnet traffic involving multiple malware families and heterogeneous devices.
CICIoT2023	4	9.3%	A more recent IoT-focused benchmark supporting evaluation across a broad range of attack categories.
CTU-13	3	7.0%	A foundational botnet benchmark that remains useful for comparison but is not IoT-specific.

N-BaIoT appears in 21 of the 43 studies (48.8%), making it substantially more common than any other individual dataset. This concentration is important when interpreting the high detection performance reported across the literature. N-BaIoT provides traffic from IoT devices infected primarily by Mirai and BASHLITE [133]; consequently, repeated evaluation on this benchmark provides limited evidence regarding transfer to different malware families, capture environments, device populations, or emerging botnet behaviors.

The concentration on N-BaIoT does not in itself demonstrate model overfitting. However, it increases the possibility that reported results reflect dataset-specific characteristics unless evaluation is complemented by device-held-out, attack-held-out, temporal, or cross-dataset validation. The broader properties and limitations of the major datasets are examined in detail in [Section 8](#).

[Fig. 8](#) visualizes the frequency of the most commonly used datasets.

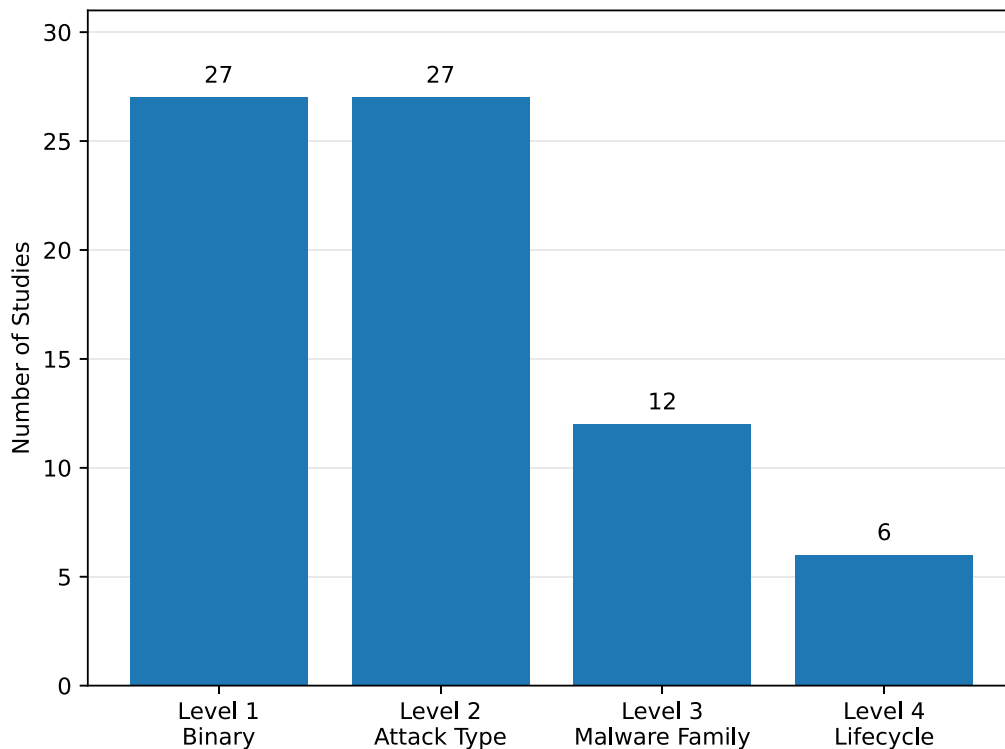


Fig. 9. Classification-depth coverage across the reviewed studies. Levels are non-mutually exclusive because a study may evaluate more than one classification objective.

Several additional datasets and custom data sources occur in only one or two studies, including CICIOT2022, UNSW-NB15, CICD-DoS2019, custom IoT testbeds, honeypot data, and specialized malware or flow representations. The resulting distribution therefore reveals both increasing dataset variety and a persistent concentration around a small number of benchmarks.

7.3. Classification-depth analysis

Classification depth was analyzed using the four non-cumulative levels defined in Section 5.5: binary botnet detection (Level 1), attack-type classification (Level 2), malware-family classification (Level 3), and lifecycle-stage modeling (Level 4). A single study may contribute to more than one level when it explicitly evaluates several classification objectives. Table 10 summarizes the resulting classification-depth coverage.

Table 10

Classification-depth coverage across the 43 reviewed studies.

Classification Depth	No. of Studies	Percentage
Level 1 — Binary Detection	27	62.8%
Level 2 — Attack-Type Classification	27	62.8%
Level 3 — Malware-Family Classification	12	27.9%
Level 4 — Lifecycle-Stage Modeling	6	14.0%

An important result is that the expanded evidence base does not support a simple interpretation in which binary detection alone dominates the literature. Binary detection and attack-type classification are each evaluated in 27 studies (62.8%). The sharper reduction occurs at deeper levels of characterization: only 12 studies (27.9%) perform malware-family classification, while only 6 studies (14.0%) explicitly model or detect lifecycle stages.

Fig. 9 visualizes this distribution.

These results indicate that the principal limitation is not an absence of multiclass evaluation, but rather the limited progression toward malware attribution and lifecycle-aware analysis. Attack-type classification can distinguish categories such as DoS, DDoS, reconnaissance, or scanning, whereas malware-family and lifecycle analysis provide additional information for attribution, early intervention, and adaptive mitigation.

The non-cumulative coding is particularly important for this analysis. A study assigned Level 3 is not automatically assumed to perform Levels 1 and 2; only explicitly evaluated classification tasks are counted. This prevents higher-level studies from artificially inflating lower-level frequencies.

7.4. Deployment-evidence analysis

Deployment feasibility was evaluated using the evidence-based framework defined in Section 5.5. Rather than treating deployment suitability as a binary characteristic, each study was assigned a Low, Medium, or High deployment-evidence rating according to reported hardware, runtime, and resource measurements.

Table 11 shows that only 6 studies (14.0%) provide High deployment evidence. Most studies, 28 (65.1%), provide Medium evidence, while 9 studies (20.9%) provide little or no explicit operational evidence and are therefore categorized as Low.

Table 11
Deployment-evidence ratings across the reviewed studies.

Deployment Evidence	No. of Studies	Percentage
High	6	14.0%
Medium	28	65.1%
Low	9	20.9%

The relatively large Medium category indicates that deployment considerations are increasingly acknowledged, but comprehensive physical validation remains uncommon. Many studies report one or more indicators such as inference time or computational complexity without demonstrating the complete detection pipeline on constrained IoT or edge hardware.

To examine this distinction more closely, Table 12 reports the frequency of individual operational indicators. Only directly reported or demonstrated evidence is counted.

Table 12
Directly reported deployment and operational indicators across the reviewed studies.

Deployment Indicator	No. of Studies	Percentage	Interpretation
Computational Complexity Discussed	29	67.4%	Provides some assessment of computational requirements or algorithmic cost.
Inference Time Reported	20	46.5%	Provides direct evidence regarding runtime execution cost.
Training Time Reported	17	39.5%	Quantifies model-training cost, although this does not directly determine inference suitability.
Real-Time Evaluation	13	30.2%	Evaluates the detector under explicitly time-sensitive or real-time conditions.
Model Size / Parameters Reported	11	25.6%	Provides information relevant to storage and memory feasibility.
Scalability Evaluated	11	25.6%	Examines behavior as traffic, devices, clients, or computational scale increases.
Memory Usage Reported	9	20.9%	Provides direct information about runtime or experimental memory requirements.
Physical Edge Deployment Tested	9	20.9%	Evaluates the model on an edge, gateway, embedded, or comparable deployment platform.
Detection Delay Reported	8	18.6%	Quantifies delay between traffic observation and detection decision.
SDN Integration	4	9.3%	Integrates the detection framework with programmable network control.
Communication Overhead Reported	3	7.0%	Measures communication cost, particularly relevant to distributed and federated learning.
Energy Consumption Reported	2	4.7%	Directly assesses energy requirements of the proposed framework.

The results reveal a more nuanced deployment picture than an offline-versus-online distinction. Computational complexity is discussed relatively frequently, and almost half of the studies report inference time. However, physical edge testing, memory measurement, detection delay, energy consumption, and communication overhead remain substantially less common.

Energy reporting is particularly limited, appearing in only 2 studies (4.7%), while communication overhead is quantified in only 3 studies (7.0%). These omissions are important because both factors directly affect continuously operating and distributed IoT security systems.

Fig. 10 summarizes selected deployment indicators.

These results reinforce the distinction between *deployment motivation* and *deployment evidence*. A model described as lightweight or edge-oriented cannot be considered strongly validated for deployment solely on the basis of architectural design or high predictive accuracy. Hardware-level testing together with timing and resource measurements provides substantially stronger evidence of operational feasibility.

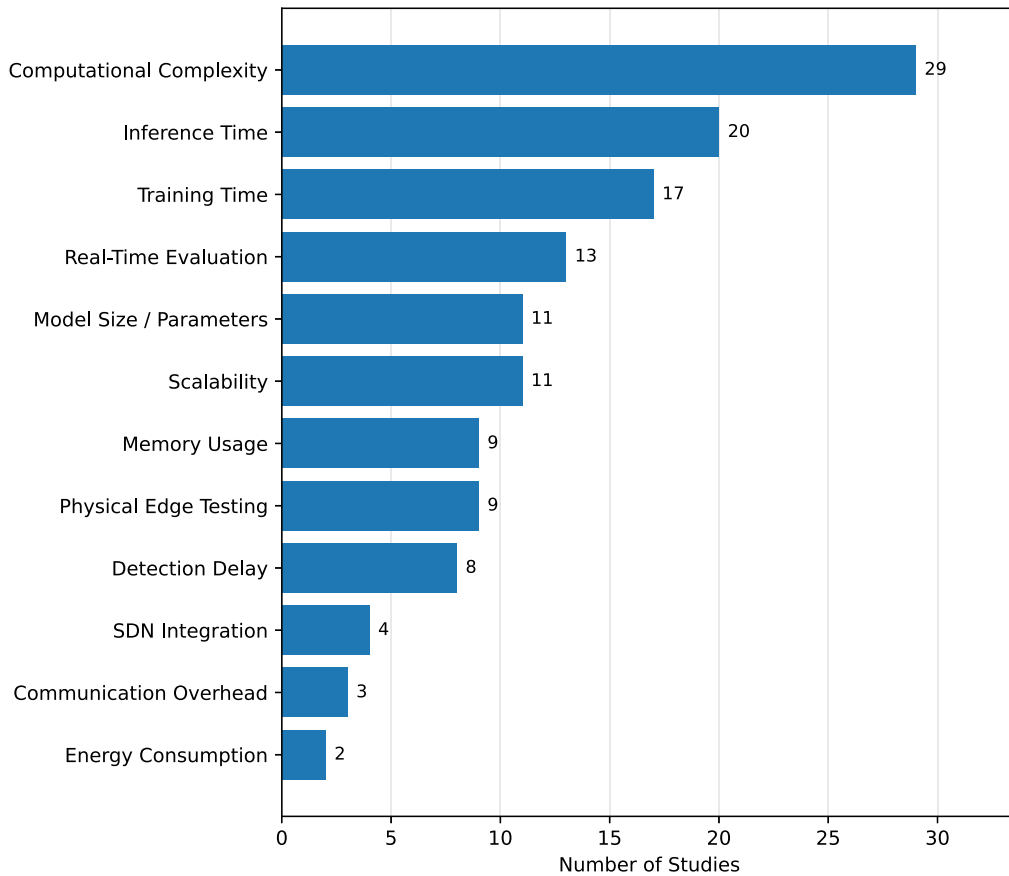


Fig. 10. Directly reported deployment and operational indicators across the reviewed studies. Categories are non-mutually exclusive.

7.5. Evaluation-metric reporting

The evaluation audit shows that conventional predictive metrics are widely reported, although reporting completeness varies substantially across individual measures. Table 13 summarizes the frequency with which the main evaluation metrics are directly reported.

Table 13
Reporting frequency of predictive evaluation metrics.

Metric	Direct	Partial	Direct (%)
Accuracy	39	2	90.7%
Precision	36	1	83.7%
Recall	37	0	86.0%
F1-score	37	1	86.0%
FPR / FAR	15	8	34.9%
ROC-AUC	17	1	39.5%

The results show that the literature is not characterized by accuracy-only reporting: precision, recall, and F1-score are also directly reported in more than 80% of the corpus. Nevertheless, a reporting imbalance remains. Accuracy is available for 39 studies (90.7%), whereas false-positive or false-alarm rates are directly reported by only 15 studies (34.9%) and ROC-AUC by 17 studies (39.5%).

This difference is operationally important. IoT botnet datasets are frequently imbalanced, and high aggregate accuracy may conceal poor detection of minority attack classes or an unacceptable rate of false alarms. Precision, recall, per-class F1, confusion matrices, and false-positive measures are therefore important for interpreting detection performance in realistic monitoring environments.

The *Partial* column identifies cases in which a metric is discussed qualitatively, inferred from figures, or derived from other reported information rather than directly provided by the study. Such values are not counted as directly reported evidence in the percentages above.

Complementary measures such as Matthews correlation coefficient (MCC) and throughput are also relevant to comprehensive evaluation. MCC is particularly informative under class imbalance, whereas throughput characterizes operational processing capacity; these measures are treated as complementary indicators rather than corpus-wide quantitative variables in the present analysis.

7.6. Dataset breadth and generalization

Evaluation on multiple datasets provides broader evidence than evaluation on a single benchmark, but dataset count alone is insufficient to establish generalization. The dataset-breadth framework therefore combines the number of independent data sources with explicit cross-dataset validation, as defined in [Section 5.5](#).

[Table 14](#) summarizes the dataset-breadth categories, while [Fig. 11](#) visualizes their distribution.

Table 14
Dataset-breadth categories across the reviewed studies.

Dataset Breadth	No.	Percentage	Interpretation
Low	20	46.5%	One data source without explicit cross-dataset validation.
Medium	9	20.9%	Two data sources without explicit cross-dataset validation.
High	14	32.6%	Three or more sources, or explicit cross-dataset validation.

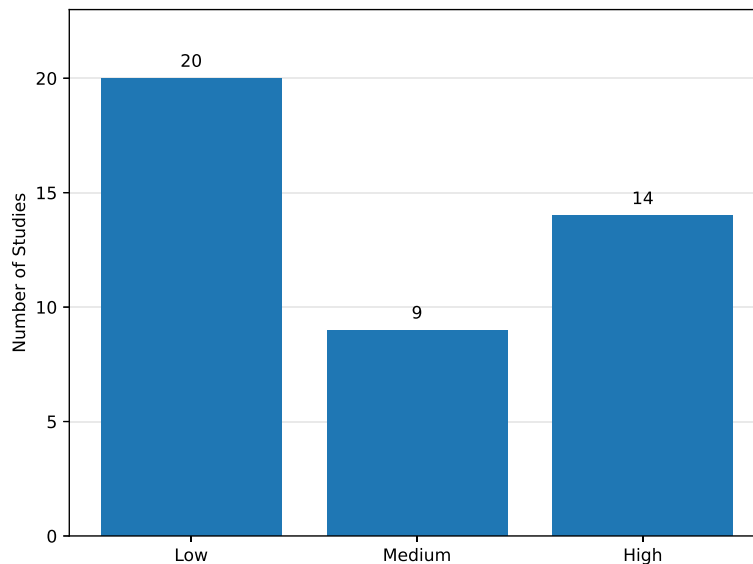


Fig. 11. Dataset-breadth distribution across the reviewed studies.

Twenty studies (46.5%) fall into the Low dataset-breadth category, demonstrating continued dependence on single-source evaluation. Nine studies (20.9%) use two distinct data sources without explicit cross-dataset validation, while 14 studies (32.6%) satisfy the High-breadth criterion.

Independently of the breadth categories, only 14 of the 43 studies (32.6%) explicitly conduct cross-dataset validation. Therefore, using several datasets does not necessarily demonstrate transferability: many studies train and test independently within each benchmark rather than training in one environment and evaluating in another.

For clarity, evaluating a model on Dataset A and Dataset B in two separate experiments, with training and testing performed independently within each dataset, is treated here as multi-dataset evaluation rather than cross-dataset validation. Cross-dataset validation requires an explicit transfer or generalization protocol across distinct data sources, for example by training on one benchmark and evaluating on another, or through an equivalent cross-domain design. The former broadens benchmark coverage, whereas the latter provides more direct evidence of robustness to distribution shift.

This distinction is central to claims of generalization. Random train–test partitions from the same capture environment may preserve device-specific, temporal, or collection-specific characteristics across both subsets. Stronger validation strategies include cross-dataset transfer, unseen-device evaluation, temporal hold-out experiments, and attack- or malware-family hold-out testing.

The combination of N-BaIoT concentration and relatively limited cross-dataset validation therefore represents a more substantial generalization concern than dataset count alone.

7.7. Explainability, zero-day evaluation, and adversarial robustness

Beyond predictive accuracy and deployment, trustworthy IoT botnet detection requires consideration of model transparency and robustness to evolving or deliberately manipulated threats. Table 15 summarizes the corresponding evidence extracted from the reviewed studies.

Table 15

Explainability and robustness indicators across the reviewed studies.

Indicator	No. of Studies	Percentage
Explicit XAI	4	9.3%
Interpretability without Formal XAI	6	14.0%
Explicit Zero-Day Evaluation	8	18.6%
Adversarial-Robustness Evaluation	3	7.0%

Only 4 studies (9.3%) employ an explicit XAI mechanism, while another 6 studies (14.0%) provide some form of model interpretability without implementing a dedicated XAI framework. This distinction is important because feature importance, attention visualization, or graph visualization can improve understanding of model behavior but should not automatically be treated as formal explainable AI.

Explicit zero-day evaluation is reported in 8 studies (18.6%). Even within these studies, the meaning of “zero-day” depends on the experimental protocol. Stronger evidence requires genuinely unseen attacks, malware families, devices, or environments to be withheld from training rather than merely relabeling existing test samples as unknown.

Adversarial machine-learning robustness is evaluated in only 3 studies (7.0%), making it one of the least explored analytical dimensions in the corpus. These studies examine adversarial evasion, structural manipulation, poisoning, or related defensive mechanisms [128–130]. Among the six studies categorized primarily as federated learning, only one explicitly evaluates poisoning or backdoor resilience.

This result indicates that conventional test-set robustness and adversarial robustness remain largely separate concerns in current IoT botnet research. High performance on an unmodified benchmark provides no direct evidence that a detector will remain reliable when an attacker deliberately adapts traffic, input features, graph structure, training samples, or federated model updates to evade detection.

Several methodological factors may contribute to the comparatively limited coverage of these dimensions. Formal XAI introduces an additional evaluation layer because providing an explanation is not sufficient by itself; its stability, fidelity, and usefulness must also be considered. Zero-day evaluation is difficult to standardize because the unseen element may refer to an attack type, malware family, device, temporal period, or complete dataset, while most commonly used benchmarks do not provide a uniform zero-day protocol. Adversarial-robustness evaluation similarly requires an explicit threat model, attack-generation procedure, perturbation assumptions, and evaluation under both clean and manipulated inputs. These additional experimental requirements make XAI, zero-day, and adversarial evaluation more involved than conventional within-dataset classification and may partly explain their limited adoption in current IoT botnet studies.

7.8. Key findings from the quantitative analysis

The quantitative analysis yields several principal findings.

First, the methodological distribution is broader than a purely DL-dominated landscape. Hybrid methods constitute the largest category (32.6%), while graph/GNN and federated approaches together account for 30.2% of the corpus, indicating growing interest in relational and distributed learning.

Second, dataset dependence remains substantial. N-BaIoT alone is used by 48.8% of the reviewed studies, while only 32.6% explicitly perform cross-dataset validation. High within-dataset performance should therefore not be interpreted as evidence of generalization without additional transfer-oriented evaluation.

Third, the classification-depth analysis provides a more nuanced result than a simple binary-versus-multiclass distinction. Binary detection and attack-type classification are equally prevalent, each appearing in 62.8% of the corpus. The principal reduction occurs at malware-family classification (27.9%) and lifecycle-stage modeling (14.0%), indicating limited progression toward deeper botnet characterization.

Fourth, deployment-related reporting is uneven. Although computational complexity is discussed in 67.4% of studies and inference time is directly reported in 46.5%, only 14.0% of the corpus satisfies the High deployment-evidence criterion. Physical edge evaluation, memory measurement, energy consumption, communication overhead, and detection delay remain considerably less common.

Fifth, predictive reporting is relatively mature for accuracy, precision, recall, and F1-score, but false-alarm and ROC-AUC reporting are much less consistent. This limits assessment of operational false-positive behavior and threshold-dependent performance.

Finally, explainability and adversarial robustness remain relatively immature dimensions of IoT botnet evaluation. Explicit XAI is present in 9.3% of the corpus, adversarial robustness in 7.0%, and explicit zero-day evaluation in 18.6%.

Overall, the evidence suggests that IoT botnet research has progressed substantially beyond basic binary detection, but important gaps remain in cross-domain generalization, malware-family and lifecycle analysis, hardware-level deployment validation, resource

reporting, explainability, and adversarial resilience. These findings provide the quantitative basis for the dataset analysis and research-gap discussion presented in the following sections.

8. Datasets used in botnet detection and classification

Datasets play a central role in the development and evaluation of AI-based IoT botnet detection systems. The quantitative analysis in Section 7 shows that dataset usage remains concentrated around a relatively small number of benchmarks. In particular, N-BaIoT appears in 21 of the 43 reviewed studies (48.8%), followed by Bot-IoT (25.6%), IoT-23 (16.3%), MedBIoT (11.6%), and CICIoT2023 (9.3%). This concentration is important because the properties of frequently reused datasets can strongly influence reported model performance and the conclusions drawn regarding generalization.

Dataset quality cannot be assessed solely by size or popularity. For IoT botnet detection, important characteristics include device diversity, attack and malware coverage, traffic-generation environment, labeling procedure, class distribution, availability of raw traffic, presence of encrypted protocols, support for multiclass classification, and suitability for evaluating unseen attacks. These dimensions are therefore considered explicitly in this section.

8.1. IoT-oriented benchmark datasets

N-BaIoT [133] is the most frequently used dataset in the reviewed corpus. It contains traffic collected from nine commercial IoT devices that were authentically infected with Mirai and BASHLITE. The released data provide statistical traffic features organized by device and attack, with malicious traffic corresponding to ten attack classes in addition to benign traffic. The original evaluation adopts an anomaly-detection setting in which models are trained on benign traffic and evaluated using the remaining benign and malicious observations.

N-BaIoT is attractive because it provides IoT-specific device traffic, clear attack organization, and a large number of observations. However, its malware coverage is restricted to only two botnet families. Consequently, very high within-dataset performance does not establish that a model will generalize to botnets with different propagation, command-and-control, or communication behavior. Its heavy use across the reviewed literature therefore introduces a risk of repeated optimization toward the characteristics of the same devices and malware families.

Bot-IoT [134] was developed in the UNSW Cyber Range to provide normal and malicious traffic for IoT-oriented network-security research. It contains DoS, DDoS, operating-system and service scanning, keylogging, and data-exfiltration attacks, and is available in PCAP, Argus-flow, and CSV representations. The dataset is especially useful for binary and multiclass learning because the traffic is organized according to attack categories and subcategories.

Nevertheless, Bot-IoT represents an experimentally constructed cyber-range environment in which IoT services and attack scenarios are generated under controlled conditions. Its complete traffic distribution is also highly imbalanced, with malicious observations substantially outnumbering benign traffic. Consequently, evaluation protocols using this dataset should report class-sensitive metrics and avoid interpreting high aggregate accuracy as sufficient evidence of operational performance.

IoT-23 [135] contains 23 IoT traffic scenarios comprising 20 malware captures and three benign captures. The malicious scenarios were generated by executing real IoT malware samples in a controlled environment with Internet connectivity, while benign traffic was collected from real consumer IoT devices including a smart lamp, an Amazon Echo, and a smart door lock. The release provides original PCAP files and labeled Zeek connection logs.

A major strength of IoT-23 is its behavior-oriented labeling. Individual flows may be labeled according to malicious behaviors such as command-and-control communication, scanning, attacks, file downloading, or related activities. Its protocol summaries also explicitly include SSL traffic. However, detailed label frequencies are highly imbalanced, and the number of benign device types remains relatively small. Thus, IoT-23 offers richer behavioral information than many benchmark datasets but does not eliminate the problem of laboratory-specific traffic characteristics.

MedBIoT [136] was designed to represent botnet activity in a medium-sized IoT environment containing 83 devices, including three physical devices and 80 emulated devices. Real samples of Mirai, BashLite, and Torii were deployed, and traffic associated with infection, propagation, and command-and-control behavior was collected. The dataset is publicly available in PCAP format and supports both binary and multiclass evaluation.

MedBIoT is particularly useful for studying botnet behavior at stages preceding the final attack because it contains infection and communication activity in addition to attack traffic. Its main limitations are the use of only three malware families and the large proportion of emulated devices. Consequently, it provides useful behavioral diversity but remains different from a large heterogeneous production IoT environment.

CICIoT2023 [137] provides a more recent and comparatively large IoT attack benchmark. It was generated using an IoT topology consisting of 105 devices and includes 33 attacks grouped into seven categories: DDoS, DoS, reconnaissance, Web-based attacks, brute force, spoofing, and Mirai. The release provides both PCAP traffic and extracted CSV features and supports binary, grouped multiclass, and fine-grained attack classification.

The dataset substantially expands device and attack diversity compared with earlier IoT-specific benchmarks. However, its attack distribution is highly imbalanced, with large-volume flooding attacks dominating many other categories. Moreover, although it increases experimental scale, the traffic is still produced within a controlled IoT laboratory. Performance on CICIoT2023 should therefore be interpreted as evidence of robustness across a broad controlled topology rather than direct evidence of Internet-scale deployment.

CICIoT2022 [138] focuses primarily on IoT-device profiling and behavioral characterization. It contains traffic from 60 physical IoT devices using technologies including Wi-Fi, ZigBee, and Z-Wave. Data were collected during device power-on, idle, active, interaction, and smart-home scenario experiments. The dataset also contains flood DoS and RTSP brute-force attack captures.

Its physical-device diversity makes CICIoT2022 useful for device-behavior and transferability studies. However, its attack coverage is narrow compared with datasets designed specifically for botnet or intrusion classification. It should therefore be regarded primarily as a profiling-oriented dataset with limited attack scenarios rather than a comprehensive botnet benchmark.

TON-IoT [139] provides a heterogeneous Industry 4.0 testbed spanning IoT/IIoT telemetry, network traffic, Windows and Linux operating-system traces, and edge/fog/cloud components. The dataset includes normal activity and several attack types, including DoS, DDoS, ransomware, and other network- and host-oriented attacks. Network data are available in PCAP, Zeek, and CSV representations, while telemetry and host data provide additional modalities.

The principal advantage of TON-IoT is its heterogeneous and multimodal structure. However, it is a general IoT/IIoT cybersecurity dataset rather than a botnet-specific benchmark. Its heterogeneous modalities also make direct comparison with traffic-only botnet datasets difficult unless preprocessing and feature representations are carefully standardized.

8.2. General and legacy network-security datasets

Several studies also rely on datasets that were not specifically developed for IoT environments. These datasets remain valuable for comparative and cross-dataset evaluation, but their domain differences must be considered when interpreting IoT-specific conclusions.

CTU-13 [140] consists of thirteen scenarios containing real botnet traffic mixed with normal and background traffic. The scenarios were manually analyzed and labeled, and labeled bidirectional NetFlow representations are publicly available. CTU-13 remains useful for evaluating conventional botnet detection and for cross-dataset comparison. However, the captures originate from 2011 and do not represent contemporary IoT devices, IoT communication protocols, or recent botnet architectures.

UNSW-NB15 [141,142] was generated in the UNSW Cyber Range by combining normal network activity with synthetically generated attacks. It contains nine attack categories and provides PCAP, Bro/Zeek, Argus, CSV, and ground-truth files. Its standardized train/test sets make it useful for general intrusion-detection experiments, but its enterprise-oriented environment limits direct transfer to IoT botnet detection.

CICIDS2017 [143] represents an enterprise-style testbed containing normal user activity and several attack categories, including brute force, DoS, DDoS, infiltration, Web attacks, Heartbleed, botnet activity, and scanning. Both PCAP and labeled flow representations are available. The normal traffic profile explicitly incorporates protocols such as HTTP, HTTPS, FTP, SSH, and email, providing some encrypted-protocol coverage. Nevertheless, the dataset is not IoT-specific and contains only limited botnet-oriented activity.

CSE-CIC-IDS2018 [144] extends the CIC methodology to a larger enterprise environment hosted on AWS. It includes seven major attack scenarios, 420 victim machines, 30 servers, 50 attacking machines, network captures, system logs, and CICFlowMeter features. HTTPS and SSH are among the protocols represented by the testbed profiles. Despite its scale and traffic diversity, its enterprise architecture differs substantially from resource-constrained IoT deployments.

ISCX Botnet 2014 [145] combines benign and malicious traces using an overlay-and-replay methodology. Its training and test sets contain multiple botnet families and communication styles, with the test set intentionally containing a broader set of botnet types than the training set. This design provides an early example of novelty-oriented evaluation. However, the dataset is legacy, non-IoT-specific, and the original provider currently reports that it is no longer available for download.

8.3. Comparative dataset analysis

Table 16 compares the major datasets in terms of environment, traffic diversity, attack coverage, data representation, and IoT relevance.

Table 17 focuses on analytical properties that directly influence model evaluation. Rather than assigning a subjective “labeling quality” score, the table records the documented ground-truth mechanism. Encrypted-traffic coverage is marked only when protocols such as SSL, HTTPS, or SSH are explicitly documented by the dataset source. Similarly, “zero-day support” distinguishes a predefined novelty-oriented protocol from a hold-out experiment that researchers may construct themselves.

8.4. Dataset bias, generalization, and evaluation limitations

Several observations emerge from the comparative analysis.

- **Concentration on a Few Benchmarks:** Nearly half of the reviewed studies use N-BaIoT. While this facilitates reproducibility and comparison, repeated training and optimization on the same benchmark can encourage models to exploit dataset-specific device, capture, or malware characteristics rather than general botnet behavior.
- **Device and Environment Dependence:** IoT datasets differ considerably in their physical-device populations and collection environments. N-BaIoT contains nine commercial devices, IoT-23 contains a small set of benign devices with controlled malware execution, MedBioT relies predominantly on emulated devices, while CICIoT2023 uses a substantially larger laboratory topology. Results obtained in one environment therefore cannot automatically be generalized to another.
- **Restricted Malware and Attack Coverage:** Some widely used IoT-specific datasets provide relatively narrow malware coverage. N-BaIoT contains Mirai and BASHLITE, while MedBioT contains Mirai, BashLite, and Torii. Such datasets are valuable for

Table 16
Environment, traffic, and attack characteristics of commonly used datasets.

Dataset	Environment / Traffic Diversity	Attack / Malware Coverage	Data Representation	IoT-Specific	Primary Generalization Limitation
N-BalIoT	9 commercial IoT devices with authentic infections	Mirai and BASHLITE; 10 malicious attack classes	Statistical feature CSVs	Yes	Only two malware families and nine devices; susceptible to family-, device-, and laboratory-specific learning.
Bot-IoT	UNSW cyber range with simulated IoT services and controlled attack infrastructure	DDoS, DoS, OS/service scan, keylogging, data exfiltration	PCAP, Argus, CSV	IoT-oriented	Controlled/simulated IoT environment and highly attack-dominated traffic distribution.
IoT-23	20 malware captures plus benign traffic from three real consumer IoT devices	Multiple IoT malware families and behavior labels including C&C, scan, attack, and DDoS	PCAP, labeled Zeek flows	Yes	Controlled malware execution and limited benign-device diversity.
MedBloT	83-device environment: 3 physical and 80 emulated devices	Mirai, BashLite, Torii; infection, propagation, and C&C activity	PCAP	Yes	Only three malware families and predominantly emulated devices.
CICIoT2023	105-device IoT laboratory with malicious IoT devices acting against IoT targets	33 attacks in 7 categories, including Mirai, DDoS, DoS, Recon, Web, brute force, and spoofing	PCAP, CSV features	Yes	Controlled topology and strongly uneven attack volumes.
CICIoT2022	60 physical devices using Wi-Fi, ZigBee, and Z-Wave	Flood DoS and RTSP brute force; extensive device-profiling scenarios	PCAP	Yes	Primarily profiling-oriented with narrow attack diversity.
TON-IoT	IoT/IoT telemetry, network, Windows/Linux, edge/fog/cloud testbed	DoS, DDoS, ransomware, and other host/network attacks	PCAP, Zeek, CSV, telemetry, OS logs	Yes, broad IoT/IIoT	Not botnet-specific; heterogeneous modalities complicate direct comparison.
CTU-13	13 real botnet scenarios with normal and background network traffic	Multiple legacy botnet scenarios and C&C behavior	Botnet PCAP, NetFlow/Argus, logs	No	Legacy 2011 traffic and no IoT-specific device behavior.
UNSW-NB15	Enterprise cyber range combining normal and synthetic attack traffic	9 attack categories including DoS, exploits, reconnaissance, backdoors, and worms	PCAP, Zeek/Bro, Argus, CSV	No	Enterprise environment and synthetic attack generation.
CICIDS2017	Five-day enterprise-style testbed with heterogeneous hosts and user profiles	Brute force, DoS, DDoS, Web attacks, infiltration, Heartbleed, Bot, scan	PCAP, flow CSV	No	Enterprise rather than IoT traffic; limited botnet-specific coverage.
CSE-CIC-IDS2018	Large AWS enterprise testbed with 420 machines, 30 servers, and 50 attacking hosts	Brute force, Heartbleed, Botnet, DoS, DDoS, Web attacks, infiltration	PCAP, event logs, flow CSV	No	Large but enterprise-oriented environment differs from IoT deployments.
ISCX Botnet 2014	Overlay/replay combination of benign traffic and multiple botnet traces	Multiple IRC-, HTTP-, and P2P-oriented botnet families	Network traces / flow data	No	Legacy, non-IoT, replay-based, and no longer available from the original provider.

Table 17
Evaluation characteristics of common IoT botnet and intrusion-detection datasets.

Dataset	Ground-Truth / Labeling Basis	Class-Balance Characteristic	Encrypted Traffic	Multiclass Support	Zero-Day / Unseen-Attack Suitability
N-BaIoT	Known benign periods and controlled authentic Mirai/BASHLITE infections	No standardized balanced multi-class split; original protocol uses benign-only training	Not specifically characterized	Yes	Malicious traffic is unseen in the original anomaly-training protocol, but there is no native held-out-family zero-day split.
Bot-IoT	Known attack infrastructure and traffic organized by attack category/subcategory	Strongly imbalanced; attack traffic dominates the full dataset	Not specifically characterized	Yes	No native zero-day protocol; attack-class hold-outs can be constructed.
IoT-23	Manual malware-traffic analysis followed by documented flow-labeling rules	Highly imbalanced across detailed behavioral labels	Yes; SSL is explicitly present	Yes	No predefined zero-day split; malware/scenario hold-outs can be constructed.
MedBIoT	Traffic organized according to benign/malicious source, malware family, phase, and device type	No standardized balanced train/test split	Not specifically characterized	Yes	No native protocol; malware-family hold-out is possible.
CICIoT2023	Controlled attack scenarios captured and labeled according to executed attack	Strongly imbalanced across attack categories	Encrypted protocols/features are present, but encryption is not the benchmark focus	Yes	No native unseen-attack protocol; requires explicit attack hold-out.
CICIoT2022	Controlled device identities, experimental states, scenarios, and attack folders	No standardized balanced attack benchmark	Not specifically characterized	Limited for attack classes	Not designed as a zero-day botnet benchmark.
TON-IoT	Ground-truth security-event files with normal/attack and attack-type labels	Balance varies across network, telemetry, Windows, and Linux subdatasets	Not specifically characterized as an encryption benchmark	Yes	No native fixed zero-day protocol; class or modality hold-outs can be constructed.
CTU-13	Manual analysis and labels applied to flow records	Strongly scenario-dependent; not designed as a balanced benchmark	Not specifically characterized	Partial	No native protocol; scenario/botnet hold-outs can be constructed.
UNSW-NB15	Ground-truth and event files with official labeled train/test partitions	Official partitions are provided but are not designed as class-balanced sets	Not specifically characterized	Yes	Known attack categories occur in the official splits; zero-day evaluation requires custom hold-out.
CICIDS2017	Attack schedule and flow identifiers including time, IP addresses, ports, and protocol	Uneven class frequencies across scheduled scenarios	Yes; HTTPS and SSH explicitly represented	Yes	No native zero-day split; attack/day hold-outs can be constructed.
GSE-CIC-IDS2018	Attack schedules and flow-level identifiers used with generated profiles	Uneven traffic volumes across scenarios	Yes; HTTPS and SSH explicitly represented	Yes	No native zero-day split; custom attack hold-out is required.
ISCX Botnet 2014	Known malicious source traces/IPs combined using overlay and replay	Overall benign/malicious proportions are moderate, but botnet-family frequencies are strongly skewed	Not specifically characterized	Yes	Native novelty-oriented design: the test set contains botnet types not represented in the training set.

controlled experiments but provide limited evidence regarding detection of botnets with substantially different communication or propagation mechanisms.

- **Class Imbalance:** Class imbalance is common but differs substantially across datasets. Bot-IoT and CICIoT2023 contain particularly uneven attack distributions, while behavioral labels in IoT-23 can also differ by several orders of magnitude. Therefore, accuracy alone is insufficient for comparing models trained on these datasets. Per-class precision, recall, F1-score, false-positive rate, and confusion matrices are important for determining whether minority behaviors are detected reliably.
- **Encrypted Traffic Coverage:** Encrypted protocols are explicitly present in some datasets, including SSL traffic in IoT-23 and HTTPS/SSH traffic in CICIDS2017 and CSE-CIC-IDS2018. However, many IoT-specific dataset descriptions do not explicitly characterize encrypted and unencrypted traffic as separate experimental dimensions. Consequently, strong results on these datasets should not automatically be interpreted as evidence of robustness to encrypted botnet communication.
- **Inconsistent Zero-Day Evaluation:** A dataset containing several attack classes does not automatically constitute a zero-day benchmark. In most datasets, unseen-attack evaluation requires the researcher to deliberately withhold a malware family, attack type, device, or scenario during training. N-BaIoT's original benign-only anomaly-training setting exposes the model to previously unseen malicious observations, but it does not constitute a held-out-malware-family protocol. ISCX Botnet 2014 provides a stronger native novelty-oriented design because its test set intentionally contains a broader collection of botnet types than its training set.
- **Laboratory-to-Real-World Transfer:** Most modern IoT datasets are collected in controlled environments because safe collection of labeled botnet traffic from production networks is difficult. Controlled testbeds provide reproducible ground truth but cannot reproduce the full variation in Internet-scale device populations, background traffic, routing conditions, software versions, encrypted services, or attacker adaptation. This limits the extent to which within-dataset performance can be interpreted as operational effectiveness.

8.5. Implications for IoT botnet evaluation

The comparative analysis indicates that dataset selection should be treated as part of the experimental methodology rather than merely as a source of training samples. No individual benchmark simultaneously provides broad malware diversity, large physical-device coverage, Internet-scale realism, balanced classes, detailed behavioral labels, encrypted-traffic characterization, and a standardized unseen-attack protocol.

Accordingly, stronger evidence of generalization requires evaluation protocols that deliberately reduce overlap between the training and test environments. Useful strategies include device-held-out evaluation, malware-family-held-out evaluation, time-based splitting, attack-type hold-out experiments, and cross-dataset transfer. In particular, training on one benchmark and evaluating on a distinct dataset provides substantially stronger evidence of domain robustness than conducting independent random train-test splits on multiple datasets.

Researchers should also report the exact dataset version, preprocessing procedure, class distribution, feature representation, split strategy, and resampling method. These details are essential because different preprocessing choices can produce substantially different experimental settings even when studies nominally use the same benchmark.

Overall, benchmark datasets remain indispensable for reproducible IoT botnet research, but high within-dataset accuracy should not be interpreted as sufficient evidence of real-world generalization. Greater emphasis on heterogeneous data sources, held-out devices and malware families, cross-dataset transfer, encrypted traffic, and temporally evolving behavior is required to bridge the gap between controlled evaluation and operational IoT security.

9. Analysis, research gaps, and open challenges

The quantitative and qualitative analyses presented in the preceding sections show that AI-based IoT botnet detection has advanced considerably in terms of methodological diversity and predictive performance. The reviewed literature includes classical ML, deep learning, hybrid architectures, graph-based models, federated learning, and adversarially robust approaches. However, several challenges continue to limit generalization, interpretability, robustness, and operational deployment.

The principal gaps identified in the 43-study corpus are not confined to detection accuracy. They concern the depth of botnet characterization, dependence on benchmark datasets, limited cross-domain validation, incomplete deployment evidence, inconsistent operational reporting, vulnerability to adversarial manipulation, and the difficulty of adapting models to evolving IoT environments.

9.1. Limitations of AI-based detection techniques

Classical ML and DL approaches provide different advantages for IoT botnet detection. ML techniques such as Random Forests, Decision Trees, Support Vector Machines, and related classifiers can provide relatively lightweight inference and clearer relationships between input features and model decisions. However, their performance often depends strongly on handcrafted or precomputed traffic features, which may encode dataset- or environment-specific characteristics.

DL approaches reduce some dependence on manual feature design by learning nonlinear, spatial, sequential, or latent representations directly from traffic features or transformed inputs. CNNs, RNNs, LSTMs, autoencoders, transformers, and related architectures have demonstrated strong classification performance across several IoT botnet benchmarks. Their increased representation capacity, however, is often accompanied by higher training cost, greater memory demand, and reduced interpretability.

Hybrid and ensemble methods further increase modeling flexibility by combining feature selection, optimization, representation learning, and multiple classifiers. Although these approaches can improve predictive performance, the additional stages may increase preprocessing, memory, and inference cost. Therefore, increased architectural sophistication does not necessarily translate into greater operational suitability.

A broader limitation across all technique categories is sensitivity to distribution shift. A model that performs well on traffic drawn from the same devices, capture environment, and dataset used for training may not retain the same performance when device populations, communication patterns, malware families, or background traffic change. This makes generalization a central challenge across both classical and deep learning approaches.

9.2. Limited fine-grained and lifecycle-aware classification

The classification-depth analysis reveals a more nuanced limitation than a simple dominance of binary detection. Binary detection and attack-type classification are each represented in 27 of the 43 studies (62.8%). The principal reduction occurs at deeper levels of botnet characterization: malware-family classification is evaluated in only 12 studies (27.9%), while lifecycle-stage modeling appears in only 6 studies (14.0%).

This gap is important because attack-type classification and botnet-family identification provide different forms of security information. Identifying a traffic flow as DDoS or reconnaissance may support immediate mitigation, whereas malware-family identification can assist attribution and threat intelligence. Lifecycle-stage modeling can provide even earlier operational value by identifying scanning, propagation, command-and-control establishment, or other pre-attack behavior.

Consequently, the main classification challenge is not merely moving from binary to multiclass detection, but progressing toward richer and more behavior-aware representations of botnet activity. Such systems must also account for overlapping attack behaviors, evolving malware variants, and the possibility that the same botnet family may exhibit different behaviors across environments.

9.3. Reactive detection and limited early-stage awareness

Many IoT botnet detectors identify malicious activity after a device has already become compromised or when clearly observable attack traffic is present. This reactive orientation reduces the opportunity for intervention before large-scale propagation or attack execution.

Earlier detection requires identifying weaker indicators such as scanning activity, repeated connection attempts, anomalous peer relationships, emerging command-and-control communication, or unusual temporal sequences. Graph-based and sequential approaches provide promising mechanisms for modeling such behavior because they capture relationships and temporal dependencies that may not be evident from isolated flows.

However, early-stage detection introduces additional difficulties. Pre-attack behaviors may overlap with benign device activity, increasing the risk of false alarms. Furthermore, datasets often provide stronger labels for completed attacks than for intermediate lifecycle stages. Improving proactive detection therefore depends not only on model design but also on richer behavioral labeling and temporally structured datasets.

9.4. Dataset bias and limited cross-domain generalization

Dataset dependence remains one of the clearest limitations identified in the survey. N-BaIoT alone is used by 21 of the 43 studies (48.8%), while only 14 studies (32.6%) explicitly perform cross-dataset validation.

This concentration creates a risk that models learn characteristics associated with specific devices, malware families, traffic-generation procedures, or preprocessing pipelines rather than general botnet behavior. The issue is particularly relevant to N-BaIoT because its malware coverage is dominated by Mirai and BASHLITE. Strong performance on this benchmark therefore provides limited evidence regarding transfer to substantially different botnet families.

Evaluation on several datasets is useful, but it should not automatically be interpreted as cross-dataset generalization. Many studies train and test independently within each benchmark. Stronger evidence requires training under one distribution and evaluating under a genuinely different distribution, such as an unseen dataset, device population, malware family, time period, or network environment.

Laboratory-to-real-world transfer remains especially challenging because most labeled IoT attack datasets are generated under controlled conditions. Such environments cannot reproduce the full diversity of production IoT networks, including changing firmware, background traffic, routing behavior, encrypted services, device failures, and adaptive attackers.

9.5. Evaluation protocol and metric inconsistency

The reviewed literature generally reports conventional predictive metrics well: accuracy is directly reported in 90.7% of the studies, while precision, recall, and F1-score are each reported in more than 80% of the corpus. However, reporting becomes much less consistent for metrics that are particularly important in operational security.

False-positive or false-alarm rates are directly reported in only 34.9% of the studies, and ROC-AUC in 39.5%. This makes it difficult to evaluate whether models with high aggregate accuracy would generate acceptable alert volumes in operational IoT environments.

Differences in preprocessing, class balancing, train-test splitting, feature extraction, and dataset versions further complicate direct comparison. Two studies using the same benchmark may effectively operate under substantially different experimental conditions.

Therefore, performance comparison across studies should be interpreted cautiously unless the evaluation protocol is sufficiently documented. Greater consistency is needed in reporting class distributions, per-class results, confusion matrices, false-positive behavior, data partitions, preprocessing procedures, and generalization protocols. Future studies would additionally benefit from wider reporting of MCC under class imbalance and throughput under deployment-oriented evaluation.

9.6. Limited real-world deployment evidence

The deployment analysis demonstrates that practical feasibility remains substantially less mature than predictive evaluation. Only 6 of the 43 studies (14.0%) satisfy the High deployment-evidence criterion defined in Section 5.5. Most studies provide partial operational evidence without complete hardware-level validation.

Computational complexity is discussed in 67.4% of the corpus and inference time is directly reported in 46.5%, indicating increasing awareness of efficiency. However, physical edge testing is reported in only 20.9%, memory usage in 20.9%, detection delay in 18.6%, communication overhead in 7.0%, and energy consumption in only 4.7%.

These omissions are particularly important for continuously operating IoT security systems. A classifier may exhibit low standalone inference latency while still requiring expensive feature extraction, traffic aggregation, graph construction, or long observation windows. End-to-end detection cost is therefore more informative than classifier execution time alone.

Realistic deployment evaluation should also distinguish between training and inference requirements. Resource-intensive centralized or federated training can remain practical if local inference is lightweight, whereas a seemingly compact model may still be unsuitable if it requires frequent retraining or high communication overhead.

9.7. Federated learning: privacy, communication, and security challenges

Federated learning is increasingly relevant to IoT security because it allows participating devices or gateways to contribute to model training without sharing raw traffic. This provides an important mechanism for improving data locality and reducing direct exposure of sensitive network observations [106,112,122,125,130].

However, federated architectures introduce challenges that do not occur in centralized training. IoT clients may differ substantially in hardware capability, traffic volume, device function, and local class distribution, producing highly non-independent and identically distributed (non-IID) data. Slow or intermittently connected clients can further affect convergence and synchronization.

Communication cost is another important limitation. Repeated transmission of model updates may become expensive for bandwidth- or energy-constrained devices, yet communication overhead is directly quantified in only a small fraction of the overall corpus.

Federated systems also enlarge the attack surface of the learning process. Malicious clients may perform data poisoning, model poisoning, backdoor insertion, or manipulated update submission. Although adversarially resilient federated approaches are beginning to appear [130], systematic evaluation of federated-learning attacks remains limited.

9.8. Adversarial robustness and adaptive attackers

Conventional benchmark evaluation assumes that test traffic is generated independently of the deployed detector. In practice, botnet operators may deliberately adapt network behavior or model inputs in response to known detection mechanisms.

Only 3 of the 43 reviewed studies (7.0%) explicitly evaluate adversarial machine-learning robustness [128–130]. These works demonstrate that evasion, structural perturbation, poisoning, and related attacks can affect IoT botnet detectors, but such evaluation remains uncommon across the broader literature.

This creates an important distinction between statistical generalization and adversarial robustness. A model may generalize successfully to naturally different traffic while remaining vulnerable to intentionally crafted perturbations. Conversely, robustness to one adversarial attack does not guarantee protection against different attacker capabilities.

Future robustness evaluation therefore requires explicit threat models defining the attacker's knowledge, access, perturbation constraints, and objectives. Evaluation should consider both clean-data performance and performance under attack to quantify the trade-off introduced by defensive mechanisms.

9.9. Limited explainability and trustworthiness

Interpretability is particularly important in security applications because detection decisions may trigger blocking, isolation, forensic investigation, or other operational responses. Nevertheless, explicit explainable-AI mechanisms are implemented in only 4 of the 43 studies (9.3%), while another 6 studies (14.0%) provide some form of interpretability without dedicated XAI methods.

Recent studies using SHAP, LIME, Anchors, rule extraction, and graph-oriented explanation illustrate the potential of XAI for identifying influential features and providing analyst-facing explanations [117,118,123,126]. However, explanation availability alone does not establish trustworthiness.

Explanations should also be evaluated for fidelity, stability, computational cost, and usefulness to security analysts. Furthermore, explanations themselves may become unreliable under dataset shift or adversarial manipulation. Trustworthy IoT botnet detection therefore requires stronger integration of robustness, transparency, calibration, uncertainty estimation, and reproducibility.

9.10. Zero-Day detection, concept drift, and model adaptation

Explicit zero-day evaluation is present in only 8 studies (18.6%). Moreover, the term “zero-day” is used under different experimental assumptions. In some cases, malicious samples are absent from training but originate from known datasets; in others, attack classes or families are deliberately withheld.

A more rigorous zero-day evaluation requires clearly specifying what is unseen during training: an attack type, malware family, device, environment, or temporal period. Without this distinction, zero-day performance reported across different studies cannot be directly compared.

Long-term deployment introduces the related problem of concept drift. Benign IoT behavior changes as firmware, applications, user patterns, and network configurations evolve, while attackers continuously modify malware and communication strategies. Static models trained on historical benchmarks may therefore lose accuracy over time.

Continual, online, transfer, and self-supervised learning provide potential mechanisms for adaptation, but they introduce additional concerns such as catastrophic forgetting, unstable updates, poisoning risk, and computational cost. Developing safe adaptation mechanisms remains an important open challenge.

9.11. Challenges in data collection, labeling, and reproducibility

Collecting realistic botnet traffic is difficult because live malware experiments create ethical and operational risks, while production IoT networks rarely provide complete attack ground truth. Consequently, many datasets rely on controlled testbeds, emulated devices, honeypots, or scheduled attack execution.

Each approach introduces trade-offs. Honeypots provide valuable malicious interaction traces but may capture only attackers that interact with the exposed service and may be recognized by sophisticated malware. Emulated devices increase experimental scale but do not fully reproduce the timing, firmware, or physical behavior of heterogeneous real devices. Controlled attack generation improves ground-truth accuracy but may produce traffic patterns that differ from uncontrolled Internet activity.

Labeling is similarly difficult, particularly for behavioral or lifecycle analysis. Binary benign/malicious labels are considerably easier to construct than reliable labels for scanning, propagation, command-and-control establishment, persistence, or coordinated attack stages.

Reproducibility also depends on reporting the dataset version, preprocessing pipeline, feature extraction method, sampling strategy, random seeds, hardware environment, and model configuration. Incomplete reporting of these details limits the ability to reproduce results and determine whether observed improvements arise from the model itself or from differences in the experimental pipeline.

9.12. Summary of research gaps

The analysis identifies the following major research gaps:

- **Limited fine-grained classification:** Attack-type classification is increasingly common, but malware-family and lifecycle-stage modeling remain substantially less explored.
- **Insufficient proactive detection:** Most systems remain focused on identifying established malicious activity rather than detecting early scanning, propagation, or command-and-control behavior.
- **Dataset concentration and weak cross-domain validation:** A small number of benchmarks dominate evaluation, while explicit cross-dataset, cross-device, and malware-family transfer remain limited.
- **Incomplete operational evaluation:** Hardware-level testing, memory consumption, energy use, communication cost, and end-to-end detection delay are substantially less reported than conventional predictive metrics.
- **Inconsistent evaluation protocols:** Differences in preprocessing, class balancing, data splitting, metric reporting, and zero-day definitions limit fair cross-study comparison.
- **Limited adversarial robustness:** Only a small proportion of studies evaluate evasion, poisoning, backdoor, or related adversarial-learning threats.
- **Limited explainability and trustworthiness:** Formal XAI remains uncommon, and evaluation of explanation fidelity, uncertainty, stability, and analyst usefulness is still limited.
- **Federated-learning security and communication challenges:** Client heterogeneity, non-IID traffic, malicious participants, synchronization, and update overhead remain unresolved barriers to large-scale federated IoT security.
- **Weak support for long-term adaptation:** Zero-day detection, concept drift, continual learning, and safe adaptation to evolving botnet behavior remain insufficiently addressed.
- **Data collection and reproducibility limitations:** Controlled testbeds, limited device diversity, incomplete lifecycle labels, and inconsistent reporting continue to constrain realistic evaluation.

Table 18 summarizes the relationship between the principal research gaps identified above and the corresponding future research directions.

These gaps indicate that progress in IoT botnet detection should no longer be assessed primarily through incremental improvements in benchmark accuracy. The central challenge is to develop systems that retain reliable performance across changing devices, datasets,

Table 18
Mapping of identified research gaps to future research directions.

Identified Research Gap	Corresponding Future Research Direction
Limited fine-grained classification	Develop behavior-centric, malware-family-aware, temporal, and lifecycle-stage detection models.
Insufficient proactive detection	Model early indicators such as scanning, propagation, emerging C&C relationships, and temporal attack sequences.
Dataset concentration and weak cross-domain validation	Increase cross-dataset, cross-device, malware-family-held-out, and temporal evaluation; investigate transfer learning, domain adaptation, and self-supervised learning.
Incomplete operational evaluation	Develop lightweight and hardware-aware edge AI and report end-to-end latency, memory, model size, energy use, and communication overhead.
Inconsistent evaluation protocols	Standardize preprocessing, data splits, class-sensitive metrics, zero-day definitions, and reproducible experimental configurations.
Limited adversarial robustness	Define explicit threat models and evaluate evasion, poisoning, backdoor, and other adversarial attacks together with appropriate defenses.
Limited explainability and trustworthiness	Evaluate XAI fidelity and stability together with uncertainty estimation, calibration, and analyst-oriented explanations.
Federated-learning security and communication challenges	Develop communication-efficient and secure federated learning that accounts for non-IID data, heterogeneous clients, poisoning, backdoors, and malicious participants.
Weak support for long-term adaptation	Investigate continual and online learning for concept drift and evolving botnet behavior while controlling unsafe updates and catastrophic forgetting.
Data collection and reproducibility limitations	Develop more diverse physical-device datasets, richer lifecycle labels, Digital Twin-assisted experimentation, and better-documented reproducible evaluation pipelines.

malware families, and operating conditions while satisfying the computational, security, interpretability, and latency constraints of practical IoT deployments.

10. Conclusion and future research

This survey presented a quantitative and deployment-aware review of 43 studies on AI-driven IoT botnet detection and classification published between 2020 and 2026. The reviewed approaches span classical machine learning, deep learning, hybrid and ensemble architectures, graph-based methods, deep reinforcement learning, and federated learning. The analysis shows that hybrid approaches constitute the largest methodological category (32.6%), followed by deep learning (23.3%). Although binary detection and attack-type classification are both widely investigated (62.8% each), deeper characterization remains limited, with malware-family classification appearing in 27.9% of studies and lifecycle-stage modeling in only 14.0%.

The results also reveal persistent limitations in generalization and deployment evaluation. N-BaIoT is used by 48.8% of the reviewed studies, whereas only 32.6% explicitly perform cross-dataset validation, indicating continued dependence on a small number of benchmark environments. Similarly, only 14.0% of the studies provide High deployment evidence. Physical edge testing, memory usage, detection delay, energy consumption, and communication overhead remain considerably less investigated than conventional predictive metrics. Trustworthiness-related evaluation is also relatively limited, with explicit XAI, zero-day evaluation, and adversarial-robustness analysis appearing in 9.3%, 18.6%, and 7.0% of the corpus, respectively.

Future research should therefore emphasize cross-dataset, cross-device, and malware-family generalization using transfer learning, domain adaptation, self-supervised learning, and temporally separated evaluation. Lightweight and hardware-aware edge AI should be assessed using end-to-end latency, memory footprint, energy consumption, and sustained runtime performance. Greater attention is also needed for behavior- and lifecycle-aware detection, continual learning under concept drift, Digital Twin-assisted experimentation, and secure federated learning that accounts for non-IID data, communication overhead, poisoning, and malicious clients. Adversarial robustness and explainability should become more systematic components of evaluation, supported by clearly defined threat models and interpretable, stable, and trustworthy model decisions.

Overall, future progress in IoT botnet detection depends on moving beyond benchmark-centered accuracy toward systems that are generalizable, deployment-efficient, behavior-aware, robust, and interpretable. More realistic and continuously evolving datasets, standardized evaluation protocols, and reproducible experimental settings will be essential for narrowing the gap between strong laboratory performance and dependable operation in real IoT environments.

CRediT authorship contribution statement

Sara Afzal: Writing – original draft, Visualization, Methodology, Conceptualization; **Abraham Gebrehiwot:** Writing – review & editing, Validation, Supervision, Conceptualization; **Filippo Maria Lauria:** Writing – review & editing, Validation, Supervision, Conceptualization; **Farrukh Aslam Khan:** Validation, Supervision.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (OpenAI) to assist with language refinement, manuscript organization, and improving the clarity of the presentation. The authors reviewed, edited, and verified the resulting content and take full responsibility for the content of the publication.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No new experimental network-traffic data were generated in this study. The quantitative analysis is based on study-level information extracted from the 43 reviewed publications. The coding and evidence table supporting the cross-study analysis is provided as supplementary material. Availability of the benchmark datasets discussed in Section 8 is reported individually; some legacy datasets may no longer be available from their original providers.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.iot.2026.102109](https://doi.org/10.1016/j.iot.2026.102109).

References

- [1] S. Afzal, A. Gebrehiwot, F.M. Lauria, F.A. Khan, Unsupervised Correlation-Based Classification of IoT Malware for Consumer Devices, in: 2026 IEEE International Conference on Consumer Electronics (ICCE) (2026) 1–6.
- [2] S. Afzal, F.A. Khan, A. Gebrehiwot, F.M. Lauria, Two-View HoneyPot Framework for Behavioral Analysis of Consumer IoT Attack Campaigns, in: 2026 IEEE International Conference on Consumer Electronics (ICCE) (2026) 1–6.
- [3] A. Oliveri, F. Lauria, Sagishi: an undercover software agent for infiltrating IoT botnets, *Netw. Secur.* 2019 (1) (2019) 9–14.
- [4] F. Lauria, How to footprint, report and remotely secure compromised IoT devices, *Netw. Secur.* 2017 (12) (2017) 10–16.
- [5] Z.A. Khan, N. Javaid, A. Saeed, I. Ahmed, F.A. Khan, Towards IoT device privacy & data integrity through decentralized storage with blockchain and predicting malicious entities by stacked machine learning, *Internet Things* 32 (2025) 101642.
- [6] H. Hamid, R.M. Noor, S.N. Omar, I. Ahmedy, S.S. Anjum, S.A.A. Shah, S. Kaur, F. Othman, E.M. Tamil, IoT-based botnet attacks systematic mapping study of literature, *Scientometrics* 126 (2021) 2759–2800.
- [7] A. Mangino, M.S. Pour, E. Bou-Harb, Internet-scale insecurity of consumer internet of things: an empirical measurements perspective, *ACM Trans. Manag. Inf. Syst.* 11 (4) (2020) 1–24.
- [8] J. Margolis, T.T. Oh, S. Jadhav, Y.H. Kim, J.N. Kim, An in-depth analysis of the mirai botnet, in: 2017 International Conference on Software Security and Assurance (ICSSA), IEEE, 2017, pp. 6–12.
- [9] L. Böck, V. Sundermann, I. Fusari, S. Karuppayah, M. Mühlhäuser, D. Levin, The end of the canonical IoT botnet: a measurement study of Mirai's descendants, *arXiv preprint arXiv:2309.01130*. (2023).
- [10] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: techniques, systems and challenges, *Comput. Secur.* 28 (1–2) (2009) 18–28.
- [11] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, E. Gillum, Botgraph: large scale spamming botnet detection, in: NSDI, 9, 2009, pp. 321–334.
- [12] R.A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro, Survey and taxonomy of botnet research through life-cycle, *ACM Comput. Surv.* 45 (4) (2013) 1–33.
- [13] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, L. Bian, Botnet detection using graph-based feature clustering, *J. Big Data* 4 (2017) 1–23.
- [14] A. Bansal, S. Mahapatra, A comparative analysis of machine learning techniques for botnet detection, in: Proceedings of the 10th International Conference on Security of Information and Networks, 2017, pp. 91–98.
- [15] Y. Shang, S. Yang, W. Wang, Botnet detection with hybrid analysis on flow based and graph based features of network traffic, in: International Conference on Cloud Computing and Security, Springer, 2018, pp. 612–621.
- [16] Somya, P. Bansal, T. Ahmad, Methods and techniques of intrusion detection: a review, in: Smart Trends in Information Technology and Computer Communications: First International Conference, SmartCom 2016, Jaipur, India, August 6–7, 2016, Revised Selected Papers 1, Springer, 2016, pp. 518–529.
- [17] M. Ahmed, A.N. Mahmood, Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection, *Ann. Data Sci.* 2 (1) (2015) 111–130.
- [18] W. Wu, J. Alvarez, C. Liu, H.-M. Sun, Bot detection using unsupervised machine learning, *Microsyst. Technol.* 24 (2018) 209–217.
- [19] S.-H. Li, Y.-C. Kao, Z.-C. Zhang, Y.-P. Chuang, D.C. Yen, A network behavior-based botnet detection mechanism using PSO and K-means, *ACM Trans. Manag. Inf. Syst.* 6 (1) (2015) 1–30.
- [20] Z. Chu, S. Gianvecchio, H. Wang, Bot or human? A behavior-based online bot detection system, *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday* (2018) 432–449.
- [21] K. Wang, C.-Y. Huang, L.-Y. Tsai, Y.-D. Lin, Behavior-based botnet detection in parallel, *Secur. Commun. Netw.* 7 (11) (2014) 1849–1859.
- [22] F. Tegeler, X. Fu, G. Vigna, C. Kruegel, Botfinder: finding bots in network traffic without deep packet inspection, in: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, 2012, pp. 349–360.
- [23] A. Pektaş, T. Acarman, Botnet detection based on network flow summary and deep learning, *Int. J. Netw. Manag.* 28 (6) (2018) e2039.
- [24] F. Haddadi, D. Le Cong, L. Porter, A.N. Zincir-Heywood, On the effectiveness of different botnet detection approaches, in: International Conference on Information Security Practice and Experience, Springer, 2015, pp. 121–135.
- [25] A. Soleymani, F. Arabgol, A novel approach for detecting DGA-based botnets in DNS queries using machine learning techniques, *J. Comput. Netw. Commun.* 2021 (1) (2021) 4767388.

- [26] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova, A technique for the botnet detection based on DNS-traffic analysis, in: *International Conference on Computer Networks*, Springer, 2015, pp. 127–138.
- [27] R. Santos, D. Souza, W. Santo, A. Ribeiro, E. Moreno, Machine learning algorithms to detect DDoS attacks in SDN, *Concurr. Comput.: Pract. Exp.* 32 (16) (2020) e5402.
- [28] A.A. Ahmed, W.A. Jabbar, A.S. Sadiq, H. Patel, Deep learning-based classification model for botnet attack detection, *J. Ambient Intell. Humaniz. Comput.* 13 (7) (2022) 3457–3466.
- [29] S.B. Gopal, C. Poongodi, D. Nanthiya, R.S. Priya, G. Saran, M.S. Priya, Mitigating DoS attacks in IoT using supervised and unsupervised algorithms—a survey, in: *IOP Conference Series: Materials Science and Engineering*, Vol. 1055, IOP Publishing, 2021, pp. 012072.
- [30] S. Almutairi, S. Mahfoudh, S. Almutairi, J.S. Alowibdi, Hybrid botnet detection based on host and network analysis, *J. Comput. Netw. Commun.* 2020 (1) (2020) 9024726.
- [31] N. Pandey, P.K. Mishra, Detection of DDoS attack in IoT traffic using ensemble machine learning techniques, *Netw. Heterog. Media* 18 (4) (2023).
- [32] R. Faek, M. Al-Fawa'reh, M. Al-Fayoumi, Exposing bot attacks using machine learning and flow level analysis, in: *International Conference on Data Science, E-learning and Information Systems* 2021, 2021, pp. 99–106.
- [33] S. García, V. Uhlir, M. Rehak, Identifying and modeling botnet C&C behaviors, in: *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, 2014, pp. 1–8.
- [34] N. Negash, X. Che, An overview of modern botnets, *Inf. Secur. J.: Glob. Perspect.* 24 (4–6) (2015) 127–132.
- [35] L. Böck, E. Vasilomanolakis, M. Mühlhäuser, S. Karuppayah, Next generation p2p botnets: monitoring under adverse conditions, in: *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10–12, 2018, Proceedings 21*, Springer, 2018, pp. 511–531.
- [36] P. Wang, L. Wu, B. Aslam, C.C. Zou, Analysis of peer-to-peer botnet attacks and defenses, *Propag. Phenom. Real World Netw.* (2015) 183–214.
- [37] M.J. Elhalabi, S. Manickam, L.B. Melhim, M. Anbar, H. Alhalabi, A review of peer-to-peer botnet detection techniques, *J. Comput. Sci.* 10 (1) (2014) 169.
- [38] H. Li, G. Hu, Y. Yang, Research on P2P Botnet network behaviors and modeling, in: *Information Computing and Applications: Third International Conference, ICIICA 2012, Chengde, China, September 14–16, 2012. Proceedings, Part I 3*, Springer, 2012, pp. 82–89.
- [39] L. Peng, C. Xiang, L. Shuhao, L. Chaoge, Poster: recoverable botnets: a hybrid c&c approach, in: *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 2011, pp. 833–836.
- [40] M. Jafari Dehkordi, B. Sadeghiyan, Reconstruction of C&C channel for P2P botnet, *IET Commun.* 14 (8) (2020) 1318–1326.
- [41] M. Abu Rajab, J. Zarfoss, F. Monrose, A. Terzis, A multifaceted approach to understanding the botnet phenomenon, in: *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, 2006, pp. 41–52.
- [42] A.H. Lashkari, S.G. Ghalebandi, M. Reza Moradhaseli, A wide survey on botnet, in: *Digital Information and Communication Technology and Its Applications: International Conference, DICTAP 2011, Dijon, France, June 21–23, 2011. Proceedings, Part I*, Springer, 2011, pp. 445–454.
- [43] G. Kirubavathi Venkatesh, R. Anitha Nadarajan, HTTP Botnet detection using adaptive learning rate multilayer feed-forward neural network, in: *IFIP International Workshop on Information Security Theory and Practice*, Springer, 2012, pp. 38–48.
- [44] P.A.A. Resende, A.C. Drummond, HTTP and contact-based features for Botnet detection, *Secur. Priv.* 1 (5) (2018) e41.
- [45] A. Udiyono, C. Lim, Lukas, Botnet detection using DNS and HTTP traffic analysis, in: *Proceedings of the 2020 International Conference on Engineering and Information Technology for Sustainable Industry*, 2020, pp. 1–6.
- [46] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, J. Zhang, Botnet: classification, attacks, detection, tracing, and preventive measures, *EURASIP J. Wirel. Commun. Netw.* 2009 (2009) 1–11.
- [47] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, I. Osipkov, Spamming botnets: signatures and characteristics, *ACM SIGCOMM Comput. Commun. Rev.* 38 (4) (2008) 171–182.
- [48] G.S. Tanwar, V. Goar, Tools, techniques & analysis of botnet, in: *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, 2014, pp. 1–5.
- [49] B. Miller, P. Pearce, C. Grier, C. Kreibich, V. Paxson, What's clicking what? techniques and innovations of today's clickbots, in: *Detection of Intrusions and Malware, and Vulnerability Assessment: 8th International Conference; DIMVA 2011, Amsterdam, the Netherlands, July 7–8, 2011. Proceedings 8*, Springer, 2011, pp. 164–183.
- [50] H. Griffioen, C. Doerr, Examining Mirai's battle over the internet of things, in: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 743–756.
- [51] R. McKay, B. Pendleton, J. Britt, B. Nakhavanit, Machine learning algorithms on botnet traffic: ensemble and simple algorithms, in: *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, 2019, pp. 31–35.
- [52] M.R. Stytz, S.B. Banks, Integrating botnet simulations with network centric warfare simulations, in: *Modeling and Simulation for Defense Systems and Applications V*, 7705, SPIE, 2010, pp. 131–142.
- [53] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, T. La Porta, On cellular botnets: measuring the impact of malicious devices on a cellular network core, in: *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009, pp. 223–234.
- [54] D. Acarali, M. Rajarajan, N. Komninos, B.B. Zarpelão, Modelling the spread of botnet malware in IoT-based wireless sensor networks, *Secur. Commun. Netw.* 2019 (1) (2019) 3745619.
- [55] S.G. Abbas, S. Zahid, F. Hussain, G.A. Shah, M. Husnain, A threat modelling approach to analyze and mitigate botnet attacks in smart home use case, in: *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, IEEE, 2020, pp. 122–129.
- [56] Y. Barse, S. Tidke, A study on BOTNET attacks and detection techniques, *IOSR J Electron Eng* 15 (3) (2020) 1–5.
- [57] T.A. Tuan, H.V. Long, L.H. Son, R. Kumar, I. Priyadarshini, N.T.K. Son, Performance evaluation of Botnet DDoS attack detection using machine learning, *Evol. Intell.* 13 (2) (2020) 283–294.
- [58] S. Baruah, D.J. Borah, V. Deka, Detection of peer-to-peer botnet using machine learning techniques and ensemble learning algorithm, *Int. J. Inf. Secur. Priv.* 17 (1) (2023) 1–16.
- [59] K.P. Grammatikakis, I. Koufos, N. Kolokotronis, C. Vassilakis, S. Shiaeles, Understanding and mitigating banking trojans: from zeus to emotet, in: *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, IEEE, 2021, pp. 121–128.
- [60] Z. Ismail, A. Jantan, M.N. Yusoff, M.U. Kiru, The effects of feature selection on the classification of encrypted botnet, *J. Comput. Virol. Hacking Tech.* 17 (2021) 61–74.
- [61] A. Alabdulatif, N.N. Thilakarathne, Z.K. Lawal, K.E. Fahim, R.Y. Zakari, Internet of nano-things (iont): a comprehensive review from architecture to security and privacy challenges, *Sensors* 23 (5) (2023) 2807.
- [62] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B.B. Gupta, Efficient IoT-based sensor BIG data collection–processing and analysis in smart buildings, *Future Gener. Comput. Syst.* 82 (2018) 349–357.
- [63] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M.H. Chaves, Í. Cunha, D. Guedes, W. Meira, The evolution of bashlite and mirai iot botnets, in: *2018 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2018, pp. 00813–00818.
- [64] H. Chunduri, T. Gireesh Kumar, P.V.S. Charan, A multi class classification for detection of IoT botnet malware, in: *International Conference on Computing Science, Communication and Security*, Springer, 2021, pp. 17–29.
- [65] M. Catillo, A. Pecchia, U. Villano, A deep learning method for lightweight and cross-device IoT botnet detection, *Appl. Sci.* 13 (2) (2023) 837.
- [66] A. Nazir, J. He, N. Zhu, A. Wajahat, X. Ma, F. Ullah, S. Qureshi, M.S. Pathan, Advancing IoT security: a systematic review of machine learning approaches for the detection of IoT botnets, *J. King Saud Univ.-Comput. Inf. Sci.* 35 (2023) 101820.
- [67] F. Suthar, N. Patel, S. Khanna, A signature-based botnet (emotet) detection mechanism, *Int. J. Eng. Trends Technol* 70 (5) (2022) 185–193.
- [68] W.-C. Shi, H.-M. Sun, DeepBot: a time-based botnet detection with deep learning, *Soft Comput.* 24 (21) (2020) 16605–16616.

- [69] A. Al-Nawasrah, A.A. Almomani, S. Atawneh, M. Alauthman, A survey of fast flux botnet detection with fast flux cloud computing, *Int. J. Cloud Appl. Comput.* 10 (3) (2020) 17–53.
- [70] S. Afrifa, V. Varadarajan, P. Appiahene, T. Zhang, E.A. Domfeh, Ensemble machine learning techniques for accurate and efficient detection of botnet attacks in connected computers, *Eng* 4 (1) (2023) 650–664.
- [71] H. Gebrye, Y. Wang, F. Li, Traffic data extraction and labeling for machine learning based attack detection in IoT networks, *Int. J. Mach. Learn. Cybern.* 14 (7) (2023) 2317–2332.
- [72] M. Shahin, F.F. Chen, H. Bouzary, A. Hosseinzadeh, R. Rashidifar, A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems, *Int. J. Adv. Manuf. Technol.* 123 (5) (2022) 2017–2029.
- [73] J. Shareena, A. Ramdas, H. AP, et al., Intrusion detection system for iot botnet attacks using deep learning, *SN Comput. Sci.* 2 (3) (2021) 1–8.
- [74] S.I. Popoola, B. Adebisi, M. Hammoudeh, H. Gacanin, G. Gui, Stacked recurrent neural network for botnet detection in smart homes, *Comput. Electr. Eng.* 92 (2021) 107039.
- [75] H. Wang, H. He, W. Zhang, W. Liu, P. Liu, A. Javadpour, Using honeypots to model botnet attacks on the internet of medical things, *Comput. Electr. Eng.* 102 (2022) 108212.
- [76] V.A. Memos, K.E. Psannis, AI-powered honeypots for enhanced IoT botnet detection, in: 2020 3rd World Symposium on Communication Engineering (WSCE), IEEE, 2020, pp. 64–68.
- [77] C. Guan, H. Liu, G. Cao, S. Zhu, T. La Porta, HoneyIoT: adaptive high-interaction honeypot for IoT devices through reinforcement learning, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2023, pp. 49–59.
- [78] W.Z. Cabral, C. Valli, L.F. Sikos, S.G. Wakeling, Advanced cowrie configuration to increase honeypot deceptiveness, in: IFIP International Conference on ICT Systems Security and Privacy Protection, Springer, 2021, pp. 317–331.
- [79] I.M.M. Martin, B. Rahardjo, The use of honeypot in machine learning based on malware detection: a review, in: 2020 8th International Conference on Cyber and IT Service Management (CITSIM), IEEE, 2020, pp. 1–6.
- [80] M. Wazzan, D. Algazzawi, O. Bamasqa, A. Albeshri, L. Cheng, Internet of things botnet detection approaches: analysis and recommendations for future research, *Appl. Sci.* 11 (12) (2021) 5713.
- [81] M. Gelgi, Y. Guan, S. Arunachala, M. Samba Siva Rao, N. Dragoni, Systematic literature review of IoT botnet DDOS attacks and evaluation of detection techniques, *Sensors* 24 (11) (2024) 3571.
- [82] T. Al-Shurbaji, M. Anbar, S. Manickam, I.H. Hasbullah, N. Alfrieat, B.A. Alabsi, A.R. Alzighaibi, H. Hashim, Deep learning-based intrusion detection system for detecting IoT botnet attacks: a review, *IEEE Access* 13 (2025) 11792–11822. <https://doi.org/10.1109/ACCESS.2025.3526711>
- [83] M. Lefoane, I. Ghafir, S. Kabir, I.-U. Awan, Internet of things botnets: a survey on artificial intelligence based detection techniques, *J. Netw. Comput. Appl.* (2025) 104110.
- [84] S. Liu, F. Fauzi, V.J. Kok, A systematic literature review: classifying IoT botnet data features based on its lifecycle, *IEEE Access* 13 (2025) 65040–65065. <https://doi.org/10.1109/ACCESS.2025.3559635>
- [85] U. Garg, P. Mishra, N. Gupta, E.S. Pilli, IoT botnets unveiled: architectural analysis, threat vectors, and cutting-edge detection techniques, *Clust. Comput.* 28 (2025) 945. <https://doi.org/10.1007/s10586-025-05633-1>
- [86] M.B. Bankó, S. Dyszewski, M. Králová, M.B. Limpek, M. Papaioannou, G. Choudhary, N. Dragoni, Advancements in machine learning-based intrusion detection in IoT: research trends and challenges, *Algorithms* 18 (4) (2025) 209. <https://doi.org/10.3390/a18040209>
- [87] A. Mahboubi, K. Luong, H. Aboutorab, H.T. Bui, S. Camtepe, K. Ansari, B. Barry, The evolving threat landscape of botnets: comprehensive analysis of detection techniques in the age of artificial intelligence, *Internet Things* 33 (2025) 101728. <https://doi.org/10.1016/j.iot.2025.101728>
- [88] A. Wani, N.K. Basha, M. Mohammed, I. Hussain, C. Ananth, H.M. Rai, AI-driven botnet detection in IoT networks: a comprehensive research review, *Comput. Sci. Rev.* 61 (2026) 100941.
- [89] H. Sonune, N. Kulkarni, Critical review of botnet sophistication and detection inefficiencies in IoT and mobile networks, *Discov. Internet Things* (2026). <https://doi.org/10.1007/s43926-026-00398-4>
- [90] N.V.M. Bindu, V.K. Nassa, P. Vasuki, G. Manikandan, R. Jeena, R. Mahaveerakannan, IoT botnet detection from software defined network using American zebra optimization algorithm with SSRNN-ELM, *Int. J. Inf. Technol.* 17 (2) (2025) 959–967.
- [91] Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks, *IEEE Internet Things J.* 7 (10) (2020).
- [92] C. Wei, G. Xie, Z. Diao, A lightweight deep learning framework for botnet detecting at the IoT edge, *Comput. Secur.* 129 (2023) 103195.
- [93] M. Al-Fawa'reh, J. Abu-Khalaf, P. Szczyk, J.J. Kang, MalBot-DRL: malware botnet detection using deep reinforcement learning in IoT networks, *IEEE Internet Things J.* 6 (2023) 9610–9629.
- [94] C.-Y. Chen, L.-A. Chen, Y.-Z. Cai, M.-H. Tsai, RNN-based DDoS detection in IoT scenario, in: 2020 International Computer Symposium (ICS), IEEE, 2020, pp. 448–453.
- [95] R. Kalakoti, S. Nömm, H. Bahsi, In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks, *IEEE Access* 10 (2022) 94518–94535.
- [96] F. Taher, M. Abdel-Salam, M. Elhoseny, I.M. El-Hasnony, Reliable machine learning model for IIoT botnet detection, *IEEE Access* 11 (2023) 49319–49336.
- [97] M.A. Haq, M.A. Rahim Khan, DNNBoT: deep neural network-based botnet detection and classification, *Comput. Mater. Contin.* 71 (1) (2022).
- [98] F. Abbasi, M. Naderan, S.E. Alavi, Dimensionality reduction with deep learning classification for botnet detection in the internet of things, *Expert Syst. Appl.* 267 (2025) 126149.
- [99] A. Alharbi, K. Alsubhi, Botnet detection approach using graph-based machine learning, *IEEE Access* 9 (2021) 99166–99180.
- [100] S.Y. Bobade, R.S. Apore, R.H. Borhade, P.N. Mahalle, Intelligent detection framework for IoT-botnet detection: DBN-RNN with improved feature set, *J. Inf. Secur. Appl.* 89 (2025) 103961.
- [101] S.I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, A.A. Atayero, Smote-drnn: a deep learning algorithm for botnet detection in the internet-of-things networks, *Sensors* 21 (9) (2021) 2985.
- [102] B.I. Hairab, M.S. Elsayed, A.D. Jurcut, M.A. Azer, Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks, *IEEE Access* 10 (2022) 98427–98440.
- [103] S. Ali, R. Ghazal, N. Qadeer, O. Saidani, F. Alhayan, A. Masood, R. Saleem, M.A. Khan, D. Gupta, A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks, *Alex. Eng. J.* 103 (2024) 88–97.
- [104] M.A. Hossain, M.S. Islam, A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection, *Sci. Rep.* 13 (1) (2023) 21207.
- [105] M.G. Karthik, M.B.M. Krishnan, Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks, *J. Ambient Intell. Humaniz. Comput.* (2021) 1–11.
- [106] F.L. de Caldas Filho, S.C.M. Soares, E. Oroski, R. de Oliveira Albuquerque, R.Z.A. Da Mata, F.L.L. De Mendonça, R.T. de Sousa Júnior, Botnet detection and mitigation model for IoT networks using federated learning, *Sensors* 23 (14) (2023) 6305.
- [107] Y.T. Salih, A. Fenjan, S.R. Ahmed, H. Ali, E.N. Abdulwahab, S. Algruri, N.A. Kurdi, M. Al-Sarem, J.F. Tawfeq, Machine learning approaches for botnet detection in network traffic, in: Proceedings of the Cognitive Models and Artificial Intelligence Conference, 2024, pp. 310–315.
- [108] A. Zaheer, S. Tahir, M.F. Almufareh, B. Hamid, A hybrid model for botnet detection using machine learning, in: 2023 International Conference on Business Analytics for Technology and Security (ICBATS), IEEE, 2023, pp. 1–8.
- [109] H. Wasswa, T. Lynar, A. Nanyonga, H. Abbass, IoT botnet detection: application of vision transformer to classification of network flow traffic, in: 2023 Global Conference on Information Technologies and Communications (GCITC), IEEE, 2023, pp. 1–6.
- [110] H. Wasswa, A. Nanyonga, T. Lynar, Impact of latent space dimension on IoT botnet detection performance: VAE-encoder versus ViT-encoder, in: 2024 3rd International Conference for Innovation in Technology (INOCON), IEEE, 2024, pp. 1–6.

- [111] A. Kumar, M. Shridhar, S. Swaminathan, T.J. Lim, Machine learning-based early detection of IoT botnets using network-edge traffic, *Comput. Secur.* 117 (2022) 102693.
- [112] L.K.G. Danquah, S.Y. Appiah, V.A. Mantey, I. Danlard, E.K. Akowuah, Computationally efficient deep federated learning with optimized feature selection for iot botnet attack detection, *Intell. Syst. Appl.* 25 (2025) 200462.
- [113] T. Altaf, X. Wang, W. Ni, G. Yu, R.P. Liu, R. Braun, GNN-based network traffic analysis for the detection of sequential attacks in IoT, *Electronics* 13 (12) (2024) 2274.
- [114] W.N. F. W.M. Zaki, R.S. Abdullah, W. Yassin, S.R. Selamat, M.S. Rosli, S. Yahya, Constructing iot botnet detection model based on degree centrality and path analysis, *J. Adv. Inf. Technol.* 15 (3) (2024) 330–339.
- [115] S. Ullah, J. Wu, Z. Lin, M.M. Kamal, H. Mostafa, M. Sheraz, T.C. Chuah, Comparative analysis of deep learning and traditional methods for IoT botnet detection using a multi-model framework across diverse datasets, *Sci. Rep.* 15 (1) (2025) 31072.
- [116] V. Sammartino, Hybrid quantum graph neural networks for robust botnet detection in modern IoT ecosystems, *Future Gener. Comput. Syst.* (2026) 108650.
- [117] R. Esmailyfard, Z. Shoaie, R. Javidan, GELAX: IoT botnet detection using dynamic graph pruning and anchored explainable AI, *Cybersecurity* 9 (1) (2026) 58.
- [118] F.S. Alsubaei, A.A. Almazroi, N. Ayub, Federated ConvNeXt-swin temporal fusion network for malware and botnet detection in IoT systems, *Sci. Rep.* 16 (1) (2026) 20147.
- [119] T. Babenko, K. Kolesnikova, Y. Bakhtiyarova, D. Yeskendiroya, K. Sansyzybay, A. Sysoyev, O. Kruchinin, Hybrid GNN–LSTM architecture for probabilistic IoT botnet detection with calibrated risk assessment, *Computers* 15 (1) (2026) 26.
- [120] K. Geetha, S.H. Brahmananda, Botnet detection through flow-based deep feature extraction and ensemble classification, *J. Inst. Eng. : B* 107 (1) (2026) 67–80.
- [121] Happy, R. Chhikara, N. Kashyap, ML-BoTDAM: machine learning driven botnet detection and alerting mechanism, *Memetic Comput.* 18 (1) (2026) 13.
- [122] A.M. Rasool, N.S. Safa, C. Mbarushimana, Towards privacy-preserving deep learning for intelligent IoT botnet detection, *Appl. Sci.* 16 (3) (2026) 1665.
- [123] M. Saied, S. Guirguis, Explainable artificial intelligence for botnet detection in internet of things, *Sci. Rep.* 15 (1) (2025) 7632.
- [124] N. Sun, L. Chen, G. Han, HADGA: Hierarchical attention-based dynamic GNN algorithm for IoT botnet detection, *IEEE Internet Things J.* 12 (16) (2025) 33520–33532.
- [125] P.K. Myakala, S. Kamatala, C. Bura, Privacy-Preserving federated learning for IoT botnet detection: a federated averaging approach, *ICCK Trans. Mach. Intell.* 1 (1) (2025) 6–16.
- [126] A.A. Korba, A. Diaf, M.A. Bouchiha, Y. Ghamri-Doudane, Mitigating iot botnet attacks: an early-stage explainable network-based anomaly detection approach, *Comput. Commun.* (2025) 108270.
- [127] W. Ma, X. Wang, J. Dong, M. Hu, Q. Zhou, A lightweight method for botnet detection in internet of things environment, *IEEE Trans. Netw. Sci. Eng.* 12 (4) (2025) 2458–2472.
- [128] D. Krishnan, P. Shrinath, Robust iot botnet detection framework resilient to gradient based adversarial attacks, *SN Comput. Sci.* 5 (7) (2024) 870.
- [129] Q.-D. Ngo, H.-T. Nguyen, V.-D. Nguyen, C.-M. Dinh, A.-T. Phung, Q.-T. Bui, Adversarial attack and defense on graph-based IoT botnet detection approach, in: 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE, 2021, pp. 1–6.
- [130] C. Lohani, A.L. Sangal, ARAFL-BAD: adversarial resilient asynchronous federated learning framework for zero-day IoT botnet attack detection, *Sādhanā* 50 (4) (2025) 302.
- [131] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. Yang, H. Zhu, D. Meng, When llms meet cybersecurity: a systematic literature review, *Cybersecurity* 8 (1) (2025) 55.
- [132] C. Wang, H. Li, W. Song, Y. Lin, Retrieval-augmented generation: a survey of security challenges and countermeasures, in: 2025 11th IEEE International Conference on Privacy Computing and Data Security (PCDS), IEEE, 2025, pp. 210–217.
- [133] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-baiot-network-based detection of iot botnet attacks using deep autoencoders, *IEEE Pervasive Comput.* 17 (3) (2018) 12–22.
- [134] N. Koroniotis, N. Moustafa, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset, *Future Gener. Comput. Syst.* 100 (2019) 779–796. <https://doi.org/10.1016/j.future.2019.05.041>
- [135] S. Garcia, A. Parmisano, M.J. Erquiaga, IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set], Zenodo (2020) <https://doi.org/10.5281/zenodo.4743746>.
- [136] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, S. Nömm, MedBioT: generation of an IoT botnet dataset in a medium-sized IoT network, in: *ICISSP*, 2020, pp. 207–218.
- [137] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A.A. Ghorbani, CICIOT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment, *Sensors* 23 (13) (2023) 5941. <https://doi.org/10.3390/s23135941>
- [138] S. Dadkhah, H. Mahdikhani, P.K. Danso, A. Zohourian, K.A. Truong, A.A. Ghorbani, Towards the development of a realistic multidimensional IoT profiling dataset, in: 2022 19th Annual International Conference on Privacy, Security & Trust (PST), IEEE, 2022, pp. 1–11. <https://doi.org/10.1109/PST55820.2022.9851966>
- [139] A. Alsaedi, N. Moustafa, Z. Tari, A.N. Mahmood, A. Anwar, TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems, *IEEE Access* 8 (2020) 165130–165150. <https://doi.org/10.1109/ACCESS.2020.3022862>
- [140] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, *Comput. Secur.* 45 (2014) 100–123.
- [141] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6.
- [142] N. Moustafa, J. Slay, The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, *Inf. Secur. J.: Glob. Perspect.* 25 (1–3) (2016) 18–31.
- [143] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, et al., Toward generating a new intrusion detection dataset and intrusion traffic characterization, *ICISSP* 1 (2018) 108–116.
- [144] Canadian Institute for Cybersecurity, CSE-CIC-IDS2018 on AWS, 2018, Intrusion detection dataset, University of New Brunswick.
- [145] Canadian Institute for Cybersecurity (CIC), ISCX Botnet Dataset, 2014, (<https://www.unb.ca/cic/datasets/botnet.html>). Accessed: 2025-07-22.