

Intrusion Detection on the Internet of Things: A Comprehensive Review and Gap Analysis Toward Real-Time, Lightweight, Adaptive, and Autonomous Security

Suzan Sallam ^{1,*}, May El Barachi ¹ and Nan Li ²

¹ School of Computer Science, University of Wollongong in Dubai, Dubai Knowledge Park, Dubai P.O. Box 20183, United Arab Emirates; maiebarachi@uowdubai.ac.ae

² School of Computer Science, University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia; nanl@uow.edu.au

* Correspondence: suzansallam@uowdubai.ac.ae

Abstract

The rapid growth of the Internet of Things (IoT) has exposed billions of interconnected, heterogeneous, and resource-constrained devices to increasingly sophisticated threats. To evaluate the readiness of current intrusion detection systems (IDSs), this study reviews 32 recent IoT-IDS proposals spanning conventional, machine-learning, deep-learning, and hybrid approaches. Each system is assessed against 10 criteria that reflect practical IoT requirements, including real-time performance, latency, lightweight design, detection accuracy, mitigation capabilities, integrated detection-and-mitigation workflows, adaptability, resilience to advanced attacks, validation in realistic environments, and scalability. The results indicate that although many approaches achieve high detection accuracy, most do not meet real-time and lightweight thresholds commonly cited in IoT deployment literature. Mitigation features are often absent, adaptability is rarely implemented, and 29 out of 32 studies rely solely on offline datasets, thereby limiting confidence in their robustness to deployment. Scalability remains the most significant limitation, as none of the reviewed IDSs have tested their performance under realistic multi-node or high-traffic conditions, even though scalability is critical for large IoT ecosystems. Overall, the review suggests that future IoT IDS research should move beyond accuracy-focused models and toward lightweight, adaptive, and autonomous solutions that incorporate mitigation, support real-time inference, and undergo standardized evaluations under real-world operating conditions.

Keywords: Internet of Things (IoT); intrusion detection system (IDS); IoT security; machine learning (ML); deep learning (DL); lightweight models; real-time detection; adaptive security; autonomous mitigation; cyber threats

Academic Editors: Amiya Nayak and Firoz Khan

Received: 21 December 2025

Revised: 25 January 2026

Accepted: 5 February 2026

Published: 7 February 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The Internet of Things (IoT) marks a significant shift by connecting devices such as traffic sensors, wearables, industrial robots, and home appliances through sensing, computing, and communication capabilities [1,2]. IoT technologies continue to reshape sectors such as healthcare, transportation, energy, and city planning by strengthening integration and operational decision-making [3]. Yet, the rapid expansion of IoT ecosystems

introduces challenges related to device heterogeneity, the lack of universal standards, persistent security and privacy gaps, ethical considerations, and growing energy demands [4]. IoT devices differ greatly in capabilities, standards, and network protocols, making security and interoperability more difficult [5]. IoT deployments also heighten privacy and safety risks, as cyberattacks can expose data or disrupt services, even when mitigated by encryption or authentication mechanisms [6]. Ethical issues, particularly those related to user privacy, data ownership, and algorithmic bias, underscore the need for transparent governance and transparency frameworks [7]. Battery life and energy efficiency remain critical factors, with energy-aware hardware and low-power communication protocols used to minimize overall power consumption [8]. The IoT technologies now underpin critical functions across smart cities, healthcare, industrial systems, and smart homes, but this widespread integration has expanded the attack surface and elevated cybersecurity risks [9]. While IoT devices in healthcare enhance clinical workflow and patient monitoring, their limited built-in security exposes sensitive medical data and can threaten patient safety [10]. Therefore, privacy and data protection concerns continue to hinder widespread adoption of IoT [11]. In smart cities, the dense concentration of connected devices amplifies the risk to both digital infrastructure and citizen privacy, and even a single unauthorized access event can disrupt essential municipal services [12,13]. The IIoT sector also faces risks stemming from sensitive operational data that is vulnerable to theft or disruption, as well as attacks that can jeopardize safety and production processes [14]. Despite its convenience, the smart home environment remains a common target due to weak authentication and poorly secured networks, often compounded by low user security awareness [15]. Recent real-world incidents highlight the practical impact of IoT vulnerabilities across sectors. For example, the 2015 BlackEnergy attack on Ukrainian utilities caused widespread power outages affecting more than 230,000 residents and exposed weaknesses in control systems [16]. In healthcare, the 2017 WannaCry ransomware encrypted medical data and forced widespread service cancellation across the UK's National Health Service (NHS), underscoring the fragility of connected medical devices [17]. Smart home devices like cameras and baby monitors face hijacking risks that threaten privacy and broader breaches, prompting manufacturers to adopt a security-by-design approach and to educate consumers [18]. Collectively, these incidents illustrate how cybersecurity failures in IoT systems can translate directly into operational, economic, and safety consequences.

As IoT continues to expand, intrusion detection has become a central component of its defense strategy. IDSs are crucial for network security, but they face challenges in adapting to IoT environments due to their unique characteristics [19,20]. **Resource constraints**, limited processing power, memory, and battery capacity, make traditional IDS architectures difficult to deploy on IoT devices and necessitate energy-efficient designs that balance security with operating lifetime [21,22]. The **diverse IoT ecosystem**, with various device types, protocols such as Wi-Fi, Bluetooth, Zigbee, and LoRaWAN, and data formats, complicates the design of unified security strategies. Different protocols require specialized detection, and heterogeneous data streams also challenge feature extraction across simple sensors and advanced edge devices [23]. Additionally, **scalability** challenges in large deployments generate high volumes of data and traffic, necessitating real-time detection with low latency. Also, the **variety of attack** methods and normal behaviors complicates the development of detection models [20]. IoT **datasets** often face issues such as class imbalance, noise, inconsistency, and limited labeled data [24]. **Privacy** restrictions further limit data sharing, hindering large-scale training and evaluation of IDS models [25].

Traditional IDS solutions were initially designed for resource-rich environments and often require computing and memory capacities that IoT devices cannot provide. These

systems are typically trained on standard network data, which usually lacks the adaptability to IoT-specific traffic patterns and attack methodologies. Network features that are practical in conventional IT networks may not apply to IoT environments. Moreover, traditional rules may fail to detect IoT-specific threats such as physical tampering and firmware exploits. Additionally, many evaluations rely on datasets that do not accurately reflect IoT environments, resulting in overly optimistic performance claims [26,27]. In contrast, AI-driven IoT intrusion detection includes supervised, unsupervised, Deep Learning (DL), hybrid, and federated models. Supervised classifiers such as Support Vector Machines (SVMs) and Random Forests (RFs) remain effective for identifying known attack patterns [28]. Unsupervised techniques detect anomalies with fewer labeled data but often have higher false positive rates [29]. Ensemble approaches can improve robustness but also introduce higher computational overhead and greater sensitivity to adversarial conditions [30]. DL models, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and hybrid architectures, achieve high accuracy at the expense of substantial resource consumption [31]. Sequential models such as Long Short-Term Memory (LSTM) networks are well-suited to modeling temporal dependencies, while convolutional models excel at extracting local features. Autoencoders (AEs) and representation learning are used for unsupervised anomaly detection and feature compression when labels are limited [27]. Additionally, hybrid frameworks that combine ML and DL aim to enhance detection accuracy but are often complex and less transparent [32]. However, distributed deployment shifts intensive processing tasks to online services and gateways, with models typically distributed via federated learning (FL) [33].

This review explains real-time intrusion detection as an IDS's ability to continuously analyze streaming traffic and make quick, latency-sensitive detection decisions based on deployment context. For example, node-level IDSs should operate within 1–10 ms for immediate response, while edge or gateway IDSs should respond within 20–50 ms for coordinated actions. Cloud-based IDSs, often using batch analytics and long-term data, usually have a latency of 100 ms or more. Lightweight deployment enables IDS to run on the limited memory, processing power, and energy of small IoT devices. MCU devices typically need models under 500 KB, with 64–128 KB RAM, and consume 1–10 mJ per detection. SBC gateways support larger models but face constraints compared to cloud systems. These criteria form the basis for evaluation in this study.

This paper provides a structured, critical review of intrusion detection in IoT, emphasizing real-time, lightweight, adaptive, and autonomous security measures. It includes a comprehensive analysis of over 30 recent IoT IDS proposals. Each solution is assessed against 10 operational criteria: real-time operation, low latency, resource efficiency, high detection accuracy, autonomy level, integration of detection and mitigation, adaptive response, handling of sophisticated attacks, validation in real-world environments, and scalability to large IoT networks. The analysis highlights gaps between reported detection performance and real-world deployment of end-to-end operational protection. It underscores the lack of unified frameworks capable of meeting real-time, lightweight, adaptive, and autonomous requirements simultaneously. The remaining sections of this paper are organized as follows: Section 2 positions this study in the context of existing IoT intrusion detection surveys, emphasizing its unique focus and contributions. Section 3 describes the materials and methods, including the review methodology and evaluation criteria. Section 4 presents the results, offering a detailed comparative analysis of the selected studies. Section 5 discusses the findings, identifies challenges, and outlines future research directions for autonomous and adaptive IoT mitigation frameworks. Finally, Section 6 concludes the paper with key insights and closing remarks.

2. Positioning Against Existing Surveys

This section contextualizes the current study within the landscape of previous IoT intrusion detection surveys, emphasizing both thematic continuities and significant advancements. Although many previous surveys have examined IoT intrusion detection using machine learning, deep learning, or architectural approaches, most narrow their focus to algorithmic accuracy and give limited attention to latency, energy efficiency, scalability, and autonomy. To address this shortcoming, we synthesize insights from 22 recent surveys alongside our systematic analysis to establish a comprehensive, comparative overview. This section categorizes these surveys by methodological focus (conventional, ML-, DL-, and hybrid-based detection), deployment environments (device, edge, fog, cloud), and evaluation practices (datasets, metrics, resource profiling). Previous surveys are compared along axes such as threat models (known attacks vs. anomaly- or zero-day detection), datasets and evaluation methods (offline benchmarks vs. streaming), performance metrics (accuracy, latency, resource use, energy), and deployment assumptions (device, edge, cloud). These show how surveys address or overlook deployment considerations. This approach aims to identify strengths and common limitations, propose an objective taxonomy of prior research, and lay the foundation for our focus on resource-aware, adaptive, and real-time IoT intrusion detection systems, and highlight the gap between experimental accuracy and practical deployment requirements, as summarized in Table 1. This review is included as a reference, showing broader, more operational coverage than prior IoT IDS surveys. Unlike earlier surveys summarized in the table, this survey explicitly assesses operational readiness across deployment tiers and includes a quantitative gap analysis.

Table 1. Summary of IoT-IDS Surveys by Category.

Survey (Year)	General IoT IDS	ML/DL-Based IDS	Edge/Fog/Cloud & FL	Protocol/SDN/NFV	Datasets & Evaluation	Lightweight DL
Zarpelo et al., 2017 [34]	✓	✓*	—	—	✓*	—
Elrawy et al., 2018 [35]	✓	✓*	—	—	✓*	—
Benkhelifa et al., 2018 [26]	✓	✓*	—	✓*	✓*	—
Hajiheidari et al., 2019 [36]	✓	✓*	—	—	✓*	—
Chaabouni et al., 2019 [37]	✓*	✓	—	—	✓*	—
Costa et al., 2019 [38]	✓*	✓	—	—	✓*	—
Naithani, 2019 [39]	—	✓*	—	—	✓*	✓
Al-Garadi et al., 2020 [23]	✓*	✓	—	—	✓	✓*
Hussain et al., 2020 [40]	✓*	✓	—	—	✓*	—
Kamaldeep et al., 2020 [41]	✓*	✓*	—	✓	✓*	—
Mazhar et al., 2020 [42]	—	✓*	—	✓	✓*	—
Albulayhi et al., 2021 [43]	✓*	✓	—	—	✓	—
Alsoufi et al., 2021 [44]	—	✓	—	—	✓*	—
Inayat et al., 2022 [45]	✓*	✓	—	—	✓*	—
Santhosh Kumar et al., 2023 [46]	✓*	✓	—	—	✓*	—
Aldhaheri et al., 2023 [47]	—	✓	—	—	✓*	—
Meziane et al., 2023 [48]	—	✓	—	—	✓	—
Kumari et al., 2023 [49]	—	✓*	—	—	✓	—
Rafique et al., 2024 [50]	✓*	✓	—	—	✓	—
Isma'ila et al., 2024 [51]	—	✓*	✓	—	✓*	—
Ali Abdul Hassan et al., 2024 [52]	—	✓*	—	—	✓*	✓
Blali et al., 2025 [53]	—	✓	—	—	✓*	—
This Review 2025	✓	✓	✓	✓*	✓	✓

Note: ✓ indicates that the survey provides primary and dedicated coverage of the corresponding theme; ✓* denotes secondary, partial, or indirect coverage; — indicates that the theme is not covered in the cited survey.

General IoT IDS surveys [26,34–36] map the IoT attack surface, compare signature-, anomaly-, specification-, hybrid-based IDS families, and review common architectures and datasets. These surveys also identify recurring issues (dataset age/quality, imbalance, and missing run-time metrics). They establish standard taxonomies and identify how IoT differs from IT (heterogeneity, constrained devices, protocol diversity). However, they consistently note challenges, including a lack of guidance for operational deployment (device/edge/cloud partitioning), the difficulty of implementing repeatable system-level evaluations (latency, memory, energy), and the pursuit of cross-domain generalization beyond laboratory datasets.

While **ML/DL-Centric IDS surveys** [23,37,38,40,44–47,50,53] evaluate ML and DL pipelines, by comparing feature engineering with models such as CNNs, RNNs, LSTMs, AEs, DBNs, and GANs, across datasets like KDD, NSL-KDD, UNSW-NB15, CIC-IDS, and BoT-IoT to identify distinct performance trends. While DL models typically achieve higher accuracy, traditional approaches remain lighter and more computationally efficient. Key limitations repeatedly noted across these surveys include overfitting, data leakage, class imbalance, and limited model interpretability. Several surveys also emphasize the need for benchmarking that reflects device-level constraints rather than server-class hardware. Further research is needed into online learning, adversarial robustness, privacy compliance, reproducibility, and device-specific latency and energy concerns.

Conversely, **Edge/Fog/Cloud and Federated-Learning (FL) surveys** [51] analyze the deployment of training and inference infrastructure (such as fog and cloud), the handling of non-IID data in federated aggregation, and the communication overhead associated with distributed learning. The survey highlights how these design choices impact privacy, latency, and the selection of FedAvg variants and validation metrics. These surveys also highlight several unresolved gaps, including the lack of empirical measurements of CPU, RAM, and energy consumption on IoT devices, limited examination of robustness against poisoning or backdoor attacks, and inadequate focus on long-term lifecycle drift in IoT deployments.

Furthermore, **Protocol-/Vertical-Focused and SDN/NFV-Assisted Defense** [41,42] connect IDS to specific stacks (e.g., SDN, MUD) or integrated defenses (IDS/IPS/IRS), as well as to programmable mitigation. They offer insights into “Detection-to-action” patterns using SDN controllers, unified IDPS taxonomies, and policy enforcement discussions. They also identify overlooked issues, including the safety risk posed by false positives, policy conflicts in multi-tenant environments, and the need for scalable validation on heterogeneous hardware.

Articles on **Datasets, Evaluation Practice and Comparative Performance** [43,48,49] offer comparative assessments of algorithms and datasets, taxonomies supported by reference architectures, and benchmarking studies emphasizing experimental rigor, such as data splits and metrics beyond accuracy, and transparency regarding the datasets’ suitability. They also note the continued absence of streaming benchmarks with energy and latency labels, unified testbeds, and standardized cross-dataset protocols for testing generalization.

Moreover, surveys on **Lightweight Deep Learning for Constrained Devices** [39,52] focus on pruning, quantization, distillation, parameter sharing, and collaborative optimization to customize DL models for edge devices. They explain compression methods and how to keep accuracy when using smaller models. However, these surveys also note the absence of systematic benchmarking for latency and energy use across microcontrollers

and SoCs, and the limited integration of lightweight DL with adaptive or federated strategies under intermittent connectivity.

While several previous surveys partially overlap with aspects of this work, such as ML/DL technique comparisons, dataset analysis, or architectural categorization, they typically examine these areas separately and do not consider deployment challenges. This survey uniquely evaluates IoT deployment readiness across tiers using 10 criteria, supported by data that reveals gaps, such as unreported latency and scalability issues. Unlike earlier studies, this review emphasizes real-world operational challenges through a structured, data-driven gap analysis of 32 IoT IDS studies.

3. Materials and Methods

This review provides a structured comparison of recent IoT IDS developments and highlights their progress, limitations, and research gaps in real-time, lightweight, adaptive, and autonomous security.

3.1. Search Strategy and Selection

A targeted literature search was conducted across five primary scholarly sources: IEEE Xplore, ScienceDirect, SpringerLink, Scopus, and MDPI journals (used as a publishing platform rather than a standalone bibliographic database), from **March to December 2025**. To capture both foundational and recent developments, studies published between **2013 and 2025** were included. For transparency, the core search string used (with syntax tailored for each database) was: (“IoT intrusion detection” OR “IoT IDS”) AND (“real-time” OR latency OR lightweight OR resource-efficient) AND (adaptive OR autonomous OR mitigation). Database-specific syntax variations (such as field tags and wildcard handling) were applied while maintaining consistent semantic constraints across sources. No additional topical, venue, or citation-count filters were used beyond the stated constraints of peer-reviewed publication, English language, and relevance to IoT intrusion detection.

Records from different sources were consolidated in a reference manager (Mendeley Reference Manager, 2.141.2) for automated duplicate detection, followed by manual verification using an Excel (Microsoft Excel, 16.89.1) screening sheet. Duplicates were removed before relevance screening. Citation chaining and similarity-based searches were also used to identify relevant studies.

A four-stage screening process was used: 1. **Identification:** Retrieving publications with the keyword groups described above. 2. **Screening:** Reviewing titles and abstracts to verify relevance to IoT-focused intrusion detection. 3. **Eligibility:** Evaluating full texts based on predefined inclusion and exclusion criteria. 4. **Inclusion:** Selecting studies suitable for structured comparison. A single reviewer conducted screening and eligibility assessments using predefined criteria. Ambiguous cases were re-evaluated at the full-text stage for clarity. The review followed a structured protocol based on specific criteria, so no disagreements between reviewers were observed, occurred, or needed to be resolved.

Inclusion criteria required: (i) a clear focus on IoT-based intrusion detection or prevention; (ii) the use of simulation, experimental, analytical, or conceptual evaluation with a defined methodology; and (iii) publication in peer-reviewed venues in English.

Exclusion criteria removed studies that focused solely on traditional IT or enterprise networks, offered only theoretical discussion without validation, or reused benchmark models without providing methodological advancement.

The initial search yielded 124 records. After removing duplicates through both automated and manual methods, 96 unique studies were identified. During title and abstract screening, 41 papers were excluded, and an additional 23 studies were eliminated during full-text review, resulting in 32 articles included in the final selection. Each selected research was analyzed using a structured comparison matrix that captured: detection

approach, deployment level, attack types covered, dataset(s) used, real-time capability, resource efficiency, mitigation features, adaptiveness, and stated limitations. This matrix served as the basis for the evaluation criteria in Section 3.2 and the comparative analysis in Sections 4 and 5.

Figure 1 summarizes the end-to-end methodological workflow adopted in this review, from literature search and study selection to evaluation and synthesis.

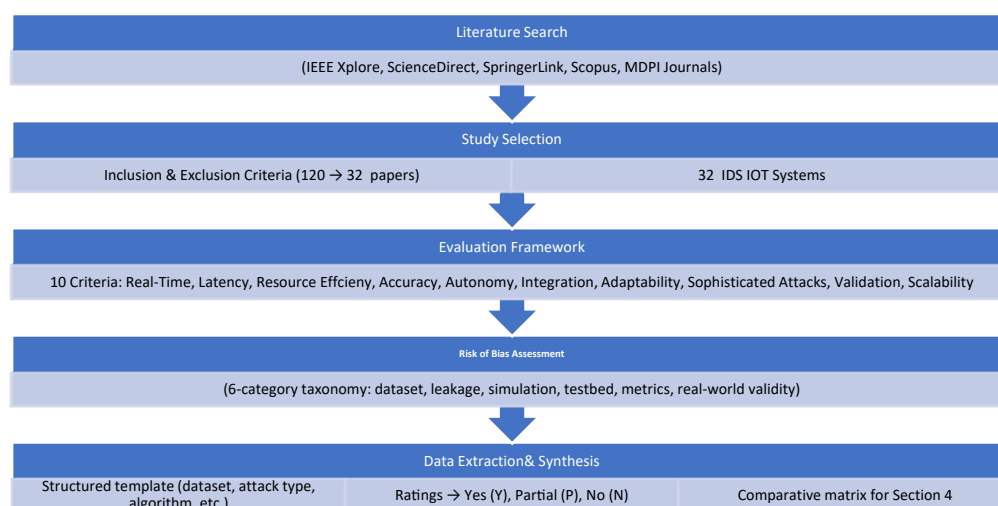


Figure 1. Methodological framework summarizing the workflow from literature search and study selection to classification, evaluation, and synthesis.

3.2. Key Evaluation Criteria

Clear evaluation criteria aim to compare IoT intrusion detection systems at the system level, considering deployment factors beyond accuracy. They assess if an IDS can operate within the timing, resource, and safety limits of real IoT environments. This framework ensures consistent, auditable evaluation of operational readiness across devices, deployment levels, and testing methods. To enable an effective and deployment-aware comparison of existing IoT intrusion detection systems, this study evaluates all reviewed approaches against a comprehensive set of operational criteria. These criteria are based on three main IoT deployment categories: **node-level devices** (MCU-Class IoT Nodes), typical hardware like ARM Cortex-M0/M3/M4/M7 and ESP32, which are highly constrained endpoints; **gateway-level systems** (SBC-Class Edge/Gateway Nodes), including standard hardware like Raspberry Pi 3/4/5 and NVIDIA Jetson Nano (lower tier), responsible for intermediate processing at the edge or fog environments; and **cloud-level infrastructures** (High Compute + Accelerator Support), such as GPU server hardware, multi-core x86, and high-end edge AI boxes, which provide centralized analytics and virtually unlimited computational capacity. Organizing the criteria around these three levels of deployment ensures that the assessment accurately captures the diversity of IoT environments while remaining broadly applicable across different device types and application domains.

Operational thresholds were set based on prior IoT intrusion detection, edge security research, hardware constraints, and real-world system needs [23,26,34,52]. MCU latency standards reflect sensing and actuation timing and influence physical processes, while similar standards are used in real-time IoT IDS and embedded analytics [40,52]. Gateway and edge thresholds account for limited buffering and data aggregation, ensuring prompt responses for mitigation and analysis, as in fog- and edge-based IDS [23,51]. Cloud thresholds are higher, factoring in network delays and cloud roles in batch analytics and training [34]. Resource thresholds for model size, RAM, and energy align with popular IoT hardware such as ARM Cortex-M, ESP32, and Raspberry Pi, following lightweight AI practices

[38,52]. Accuracy and false-positive thresholds follow standard intrusion detection practices for imbalanced data and safety-critical environments [26,37]. These thresholds are practical reference points for evaluating IDS readiness across various IoT settings, based on real-world data rather than universal standards.

Beyond performance, evaluation also considers ethical, privacy, and adversarial risks in deploying IoT intrusion detection systems. Factors like false positives, autonomous responses, and real-world testing address safety, fairness, and reliability concerns. Excessive false alarms, secretive decision processes, or unverified autonomous actions could pose operational or ethical challenges, especially in safety-critical contexts. These factors influence thresholds and result interpretations, with further discussion of ethical and adversarial issues in Section 5.4.

Real-time detection evaluates whether the IDS can continuously monitor traffic and system activity as events happen, providing timely outputs within the deadlines required by IoT control loops. A system might have low latency but still fail in real-time operation if it processes data in batches, analyzes data retrospectively, or cannot sustain line-rate detection. Real-time capability is assessed by (i) continuous streaming operation, (ii) per-packet or per-event inference, and (iii) ability to keep pace with the data rate without buffering delays. A system is marked *Yes* if it performs continuous online detection without relying on batch processing, *Partial* indicates near-real-time behavior with occasional delays, and *No* indicates offline or batch-only analysis.

Low detection latency, in contrast, is checked by ensuring systems detect threats within suitable latency ranges for each deployment class. MCU-class target latency $\leq 1\text{--}10$ ms, as sensors and actuators operate near physical processes and often lack buffering, making timely decisions essential. SBC-class IDS aims for $\leq 20\text{--}50$ ms latency, manages data between devices and the cloud, and enables batch detection and complex inference without sacrificing responsiveness. Cloud IDS typically exhibits latencies ≥ 100 ms due to network propagation delays and their role in batch analytics, model training, and long-term correlation, rather than immediate response, since physical distance causes delays often over 50–100 ms, unsuitable for immediate threat response, focusing instead on batch analytics over larger time windows. A system is marked *Yes* when it consistently meets the latency expected for its deployment tier, *Partial* when latency is higher but still usable, and *No* when latency exceeds acceptable limits or is not reported.

A **resource-efficient** IDS must minimize memory, compute, and energy usage while maintaining acceptable detection performance. **a. Model size** is the storage required for the trained IDS parameters. MCUs have minimal Flash storage, typically 64 KB to 1 MB, so IDS models must stay below 500 KB. SBCs have 512 MB to 4 GB of RAM and SD/eMMC storage, capable of hosting moderate ML/DL models but still struggle with extensive neural networks. Models over 50 MB can significantly increase inference latency and energy consumption on SBCs. Cloud environments impose no practical size limit. **b. Peak RAM**, or runtime memory usage, is the maximum amount of RAM required during inference; spikes in RAM use can cause devices to crash or reset. MCUs with 2–512 KB RAM should typically have a peak RAM threshold of $\leq 20\text{--}64$ KB, with an upper limit of 128 KB. SBCs with 0.5–4 GB RAM can handle peak RAM up to 250–500 MB, with a maximum of around 1 GB. Cloud environments, with 16–256 GB RAM or more, have realistic peak RAM thresholds in the tens of GBs; there is no fixed upper limit. **c. Compute per inference**, measured in FLOPs (floating-point operations) or MACs (multiply-accumulate operations), indicates how many mathematical operations the model performs on one input. MCU CPUs operate at 16–240 MHz, so the compute threshold per inference should be $\leq 10^4\text{--}10^6$ FLOPs (10 thousand to 1 million). SBC CPUs, with 1–4 cores @ 1.2–1.8 GHz, often include small GPUs, and their per-inference compute limit is about $10^7\text{--}10^9$ FLOPs (10 million to 1 billion). Cloud servers have no strict limit. **d. Energy per inference**, measured in

millijoules (mJ) for MCUs and joules (J) for SBCs, indicates the power consumed during detection. MCUs often rely on batteries or energy harvesting. The ideal energy per inference is less than 1 mJ, with an absolute maximum around 1–10 mJ. SBCs powered externally or connected to mains should operate within 1–100 mJ per inference. Clouds have no strict energy limits; they often consume 1–10 J per inference, while optimized GPU models typically use 100–500 mJ. The reported resource thresholds reflect representative capabilities of commonly used IoT platforms (e.g., ARM Cortex-M MCUs, ESP32, Raspberry Pi-class gateways) rather than optimal or idealized configurations. We assign *Yes* when model size, RAM usage, compute cost, and energy consumption fall within the thresholds of the intended device tier; *Partial* when limits are slightly exceeded; and *No* when the footprint is clearly unsuitable or unreported.

High detection accuracy is essential for IoT IDS, ensuring malicious activities are identified correctly while minimizing operational disruptions. Accuracy should be at least 90%, and the F1-score should be at least 0.85 to address the typical class imbalance in IoT traffic. Precision for attack classes should stay above 0.90 to prevent false alarms from overwhelming automated mitigation systems, while recall should be ≥ 0.85 in general IoT deployments and ≥ 0.90 in safety-critical sectors such as healthcare or industrial control. The false-positive rate should remain below 5% to maintain operational stability. Although these thresholds are universal, their practical achievability varies by deployment type: node-level MCUs often achieve lower accuracy due to resource and feature limitations, gateway platforms support more robust models with higher performance, and cloud deployments generally exceed 95% accuracy through deep learning or ensemble methods. A system is marked *Yes* when it meets all three metrics ($\geq 90\%$ accuracy, ≥ 0.85 F1, $\leq 5\%$ FPR), *Partial* applies when some but not all criteria are met, and *No* if it fails to reach these minimum detection criteria. Low false-positive rates are significant in autonomous or safety-critical IoT applications, where false alarms could cause unnecessary panic or even dangerous actions.

Autonomous Mitigation describes an IoT security system's ability to independently analyze threats and take corrective action without ongoing human intervention. To clarify this capability, we define four measurable levels: L0-Detect Only, where the system identifies threats but does not act; L1-Action Recommendation, where the system suggests mitigation steps for operator approval; L2-Policy-Bound Autonomy, where the system executes predefined mitigation actions (such as blocking traffic or quarantining nodes) based on explicit safety policies; and L3-Closed-Loop Autonomy, where the system performs fully automated mitigation with guardrails, rollback options, and continuous feedback. We classify systems as *Yes* when they demonstrate L2–L3 autonomy, *Partial* for L1 or limited L2 actions, and *No* for purely detection-only designs. Since incorrect automated actions could lead to service disruptions or physical harm, we carefully assess mitigation measures, considering policy limitations, rollback options, and the acceptable rate of false positives.

Integrated Detection and Mitigation assesses whether intrusion detection and response function as a cohesive system rather than separate parts. Integrated systems combine threat identification with immediate, context-aware mitigation, reducing human delay, preventing lateral movement, and ensuring service continuity in dense IoT deployments. A system is marked *Yes* when detection and response form a coordinated workflow; *Partial* when both components exist but require manual linkage; and *No* when no response mechanism is provided.

Adaptive Response Capability refers to an IDS's ability to update its detection logic as threats evolve, automatically adjust thresholds, and maintain performance despite fluctuations in traffic patterns or concept drift. Systems are assessed based on three quantifiable functions: **a. Model Update Mechanism (Support for Learning Frequency):** *Online*

or *incremental learning*, capability to continuously or iteratively update model parameters. *Retraining intervals* categorize updates as ≤ 1 h (highly adaptive), ≤ 24 h (moderately adaptive), or > 24 h/no updates (static). **b. Concept Drift Detection:** Implementation of explicit drift detection modules. Proven ability to maintain $\leq 10\%$ degradation in F1-score following drift or in scenarios of evolving traffic. **c. Threshold or Policy Auto-Adjustment:** Enables automatic modification of anomaly thresholds, mitigation rules, or confidence levels without human intervention. For scoring, *Yes* indicates support for online learning, drift detection, and automatic threshold or policy adjustments. *Partial* signifies support for one or two of these features (typically retraining or threshold adaptation); *No* indicates static models with fixed thresholds and no awareness of drift.

Ability to **Handle Sophisticated Attacks** (Zero-Day, Botnets, APTs). This criterion evaluates whether an IDS can identify advanced, unseen, or stealthy attack behaviors. The assessment considers the following factors: **a. Zero-Day Detection Ability (Generalization Beyond Training Data):** Shows anomaly-based, behavior-based, or hybrid detection for unknown attack types, with F1-score for unseen attacks ≥ 0.80 (strong), ≥ 0.60 (moderate), and < 0.60 (weak). **b. Botnet/APT Behavior Detection:** Demonstrates testing on botnet datasets and can detect multi-stage or persistent attack sequences using temporal or correlation models; it can identify lateral movement, low-rate stealthy anomalies, and sequence reconstruction. **c. Robust Feature Set for Hidden/Stealthy Behavior:** use of temporal features, flow statistics, or behavioral profiles that enable detection of stealthy or low-rate attacks. For evaluation, *Yes* means detecting unseen attacks with $F1 \geq 0.80$, tested on botnet datasets, and supporting temporal or correlation-based modeling; *Partial* indicates detection of some sophisticated attacks ($F1 \geq 0.60$) or testing only on limited datasets; *No* refers to systems that are signature-based or entirely static, unable to handle unseen or stealthy attacks.

Real-World Validation (Testbeds, Pilot Deployments, Production Environments) evaluates whether an IDS has been tested under actual operating conditions rather than in simulations or static datasets. Real deployments reveal issues, such as sensor noise, protocol variability, and device heterogeneity, that controlled datasets can't capture. The validation strength is measured using three factors: **a. Environment Type Tested:** Full real-world deployment on physical IoT devices (e.g., smart homes, industrial sensors, cameras, gateways), lab validation with physical testbeds involving mixed protocols, real traffic, and multiple devices, or simulation-only with synthetic data or emulated traffic. **b. Device Diversity:** High (≥ 3 device types such as sensors, cameras, actuators, gateways), moderate (2 types), or low (single device type or none, dataset-only). **c. Traffic Realism:** Incorporates realistic noise, protocol heterogeneity, bursty loads, and long-term behavior, with performance reports under real-world noise or multi-day operation. Scoring, *Yes* indicates testing on real hardware or full testbeds with diverse devices and traffic; *Partial* signifies lab testing with limited diversity or traffic realism; *No* means evaluation was only through simulations or datasets without real devices.

Scalability (Nodes, Traffic Volume, System Growth): Scalability measures whether an IDS can sustain acceptable performance as the IoT network expands in terms of nodes, data rate, and traffic complexity. IoT deployments, ranging from smart cities to industrial systems, can scale from tens to millions of devices, making it crucial to measure scalability. To assess scalability, three quantifiable metrics are used: **a. Node Scalability (Device Count Supported):** High scalability, tested with $\geq 10,000$ simulated nodes or ≥ 100 physical nodes. Moderate scalability, tested with 1000–10,000 simulated nodes or 10–99 physical nodes. Low scalability ≤ 1000 simulated nodes or ≤ 10 physical devices. **b. Throughput Scalability (Traffic Volume):** High, supporting ≥ 1 Gbps of total traffic or $\geq 10,000$ packets/s without degrading performance by more than 10%. Moderate, supporting 100 Mbps–1 Gbps or 1000–10,000 packets/s. Low, supporting less than 100 Mbps or fewer than 1000

packets/s. **c. Performance Stability Under Scale:** Assessed by changes in latency, detection accuracy, and CPU/memory usage as load increases. Considered stable when accuracy drops by $\leq 5\%$ and latency increases by ≤ 10 ms. Moderate when accuracy drops $\leq 10\%$ and latency increases ≤ 25 ms. Weak when accuracy drops $> 10\%$ or latency increases > 25 ms. We mark *Yes* for systems showing high scalability in at least two dimensions; *Partial* for moderate scalability or only one strong dimension; and *No* for limited scalability or a lack of scaling tests.

Where studies did not provide enough information to evaluate a criterion against these thresholds, the corresponding assessment was cautiously marked as “No” or “Partial,” reflecting reporting gaps rather than assumed non-compliance.

3.3. Risk of Bias and Evaluation Limitations

To ensure a transparent and reproducible assessment of the selected studies, a risk-of-bias evaluation was performed, focusing on common methodological weaknesses seen in IoT IDS research. Since most works rely on dataset-driven or simulation-based evaluations, several sources of bias were identified and organized into a structured taxonomy.

Dataset Bias: Many studies continue to rely on outdated or imbalanced datasets, such as KDD’99 and NSL-KDD, which can inflate accuracy and underestimate false positives. A limited variety of benign traffic or the absence of realistic device behaviors can also cause models to rely on pattern memorization rather than on generalizable detection.

Data Leakage and Improper Splits: Some studies report accuracy based on random train-test splits, in which flows from the same attack instance appear in both sets, leading to *temporal leakage* that artificially improves performance. The lack of cross-device, cross-time, or cross-scenario validation is seen as a moderate-to-high bias factor.

Simulation-Only Evaluation: Many IDS proposals rely on simulated network traffic or synthetic device behavior, rather than real hardware or field data. Simulation bias arises when evaluations use idealized conditions that lack realistic noise, jitter, packet loss, or protocol diversity. Such setups tend to overestimate runtime performance and weaken claims of real-time operation.

Single-Testbed or Controlled-Environment Bias: Studies that assess IDS solutions on a single testbed, such as a Raspberry Pi cluster or a Mininet simulation, risk limited generalizability. Performance metrics often do not apply to large-scale or latency-sensitive deployments. This bias is evident when resource usage (CPU, RAM, energy) is reported only for one device type.

Metric Selection Bias: Several studies report only accuracy, omitting metrics like F1-score, recall for minority attack classes, latency, or energy consumption. Ignoring these metrics skews evaluation toward overstated performance, especially with imbalanced IoT traffic, where accuracy alone does not accurately reflect an IDS’s effectiveness.

Lack of Real-World Validation: Few studies implement IDS solutions in pilot setups, multi-protocol IoT networks, or extended experiments. This absence of practical testing creates an external validity bias, reducing confidence in claims about real-time performance, adaptivity, or autonomous mitigation.

Each of the 32 included studies was assessed across six bias dimensions and labeled qualitatively as low, medium, or high bias. These labels help explain the comparisons in Sections 4 and 5. Bias labels were assigned using explicit rules for each dimension to ensure auditability and reproducibility. Dataset bias was assessed based on age, diversity, and balance; data leakage bias by temporal or cross-scenario separation; simulation bias by dependence on datasets or simulated traffic versus physical deployments; single-testbed bias by device and environment diversity; metric bias by reporting latency, resource, and energy metrics alongside accuracy; and real-world validation bias by pilot

deployments or long-term testbeds. These rules are summarized for each study in Supplementary Table S1.

In addition to the biases identified in the reviewed studies, this review has limitations. It examined only English peer-reviewed papers, potentially missing studies in other languages. Despite thorough database searches and citation chaining, relevant work outside these sources or in less common venues might have been overlooked. Although clear criteria and multiple review stages were used, some selection bias remains inevitable. Nonetheless, our structured, transparent approach aimed to minimize bias and provide a fair overview of IoT IDS research.

4. Results

The final corpus comprises 32 peer-reviewed articles published between 2013 and 2025 that focus on intrusion and anomaly detection across various IoT scenarios. The studies are categorized into four detection paradigms: conventional (non-AI) models, ML models, DL models, and hybrid approaches that combine multiple techniques. In the reviewed corpus, AI-based methods (ML, DL, and hybrid) are far more common than traditional rule- or signature-based approaches [36,54–56]. Out of the 32 studies, 28 (87.5%) use AI techniques (ML, DL, or hybrid), while 4 (12.5%) rely on traditional detection methods. The level of AI integration was determined by the dominant detection mechanism used during inference: conventional (rule- or signature-based), lightweight ML (e.g., classical classifiers with handcrafted features), DL (end-to-end or representation-learning models), or hybrid systems combining multiple learning paradigms or stages. Throughout Section 4, claimed capabilities are distinguished from validated performance, in line with the operational definition in Section 3.2.

The studies also encompass **diverse datasets and deployment scenarios**, including traditional benchmark datasets (e.g., KDD Cup’99, NSL-KDD), more recent IoT-specific collections (e.g., BoT-IoT, ToN-IoT, IoT-23, NF-UQ-NIDS), and several custom or simulated datasets for MQTT, 6LoWPAN, or IoMT testbeds. Most studies still rely on offline evaluation, with only a limited number demonstrating deployment on fog/edge hardware or physical IoT testbeds. The following subsections (Sections 4.1–4.4) summarize the main findings for each detection approach.

This section compares IoT intrusion detection solutions against 10 criteria to identify strengths, limitations, and suitability for real-time IoT security in resource-constrained environments. The consolidated Yes/Partial/No scoring for all 32 studies across the 10 evaluation criteria is reported in Table 2, with paradigm- and deployment-level aggregations provided in Table 3 and Table 4, respectively.

Table 2. Summary of Evaluation Results for 32 IoT IDS Studies Across Ten Operational Criteria.

Ref.	Real-Time	Low Latency	Lightweight	High Accuracy	Mitigation	Integ. D & M	Adaptive	Soph. Attacks	Validated	Scalable
[57]	P	N	P	P	P	Y	N	P	P	N
[58]	Y	N	N	N	N	N	N	N	N	N
[59]	P	N	N	N	N	N	N	N	P	N
[60]	Y	N	N	P	Y	P	P	N	N	P
[61]	Y	N	N	Y	N	N	Y	P	N	N
[62]	N	N	N	Y	N	N	P	P	N	N
[63]	P	N	N	Y	N	N	N	P	N	N
[64]	P	N	N	N	N	N	N	N	P	N
[65]	N	N	N	P	N	N	N	N	P	N
[66]	Y	N	N	Y	P	P	P	Y	N	N

[67]	Y	P	N	Y	N	N	N	Y	P	N
[68]	N	N	N	P	N	N	P	P	N	N
[69]	N	N	N	Y	N	N	N	P	N	P
[70]	N	N	N	P	N	N	N	P	N	N
[71]	Y	N	N	Y	N	N	N	P	Y	N
[72]	N	N	P	Y	N	N	N	N	N	N
[73]	P	Y	Y	P	N	N	N	N	P	N
[74]	N	N	P	Y	N	N	N	P	N	N
[75]	N	N	N	Y	N	N	N	P	P	N
[76]	P	N	N	Y	N	N	P	N	N	P
[77]	P	N	N	Y	N	N	P	N	N	P
[59]	P	N	N	N	N	N	N	N	P	N
[69]	N	N	N	Y	N	N	N	P	N	P
[61]	Y	N	N	Y	N	N	Y	P	N	N
[62]	N	N	N	Y	N	N	P	P	N	N
[60]	Y	N	N	P	Y	P	P	N	N	P
[78]	P	N	P	Y	N	N	P	Y	P	N
[63]	P	N	N	Y	N	N	N	P	N	N
[64]	P	N	N	N	N	N	N	N	P	N
[70]	N	N	N	P	N	N	N	P	N	N
[71]	Y	N	N	Y	N	N	N	P	Y	N
[72]	N	N	P	Y	N	N	N	N	N	N
[73]	P	Y	Y	P	N	N	N	N	P	N
[65]	N	N	N	P	N	N	N	N	P	N
[79]	Y	N	N	Y	N	N	P	P	P	N
[80]	P	N	N	P	N	N	N	P	N	N
[81]	Y	N	N	P	Y	Y	N	N	N	N
[82]	N	N	N	Y	N	N	N	P	N	N
[83]	Y	N	N	Y	N	N	P	Y	N	N
[66]	Y	N	N	Y	P	P	P	Y	N	N
[67]	Y	P	N	Y	N	N	N	Y	P	N
[84]	N	N	N	P	N	N	N	P	N	N
[85]	P	Y	Y	P	N	N	Y	P	N	N
[68]	N	N	N	P	N	N	P	P	N	N
[74]	N	N	P	Y	N	N	N	P	N	N
[86]	N	N	N	Y	N	N	N	N	N	N
[87]	N	N	N	Y	N	N	N	P	N	N
[88]	N	N	N	Y	N	N	N	P	N	N
[75]	N	N	N	Y	N	N	N	P	P	N

Table 3. Summary of Evaluation Results Across Detection Paradigms.

Paradigm	Real-Time			Low La- tency			Light- weight			High Ac- curacy			Mitigation			Integrated D & M			Adaptive			Sophisti- cated At- tacks			Valid- dated			Scalable		
	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N
Convent.	2	2	0	0	0	4	0	1	3	0	2	2	1	1	2	1	1	2	0	1	3	0	1	3	0	2	2	0	1	3
ML	3	2	3	0	1	7	0	0	8	5	2	1	0	1	7	0	1	7	1	3	4	2	4	2	0	3	5	0	0	8
DL	1	1	5	1	0	6	1	2	4	5	2	0	0	0	7	0	0	7	0	0	7	0	5	2	1	2	4	0	1	6
Hybrid	3	5	5	1	0	12	1	1	11	8	5	0	1	0	12	1	0	12	1	5	7	2	7	4	0	2	11	0	2	11

Table 4. Cross-tabulation of intrusion detection studies by deployment layer (Device, Edge/Fog, Cloud, Hybrid) and evaluation outcomes across 10 criteria.

Deploy- ment Layer	Real-Time			Low La- tency			Light- weight			High Ac- curacy			Mitigation			Integrated D & M			Adaptive			Sophisti- cated At- tacks			Vali- dated			Scalable		
	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N	Y	P	N
Device	1	2	2	0	0	5	0	2	3	4	1	0	0	0	5	0	0	5	0	2	3	0	2	3	1	0	4	0	2	3
Edge/Fog	6	3	6	2	1	12	2	1	12	9	6	0	2	1	12	1	2	12	2	5	8	4	7	4	0	5	10	0	2	13
Cloud	0	0	2	0	0	2	0	0	2	1	1	0	0	0	2	0	0	2	0	0	2	0	2	0	0	0	2	0	0	2
Hybrid	2	5	3	0	0	10	0	1	9	4	3	3	0	1	9	1	0	9	0	2	8	0	6	4	0	4	6	0	0	10

The analysis highlights design trade-offs that limit operational readiness despite high accuracy. Three factors influence system performance: (i) model complexity and feature pipeline cost affecting latency and deployment; (ii) system placement, device, edge, or cloud, and centralization impacting responsiveness, robustness, and scalability; (iii) evaluation methods, such as offline testing versus streaming or testbed validation, which often overestimate accuracy and hide issues like runtime performance, data drift, and load handling. The following subsections discuss key studies, focusing on real-time challenges, lightweight design, mitigation, and scalability, and set the stage for the gap analysis in Section 5.

4.1. Conventional IDS Models

Conventional IDS techniques typically rely on predefined rules, signatures, or lightweight logic to flag suspicious behavior. Their main advantages are lightweight design, predictability, and ease of understanding, making them suitable for resource-limited IoT environments. These include signature- and rule-based systems, automata-based analysis, and lightweight event-driven mechanisms. Many approaches originate from traditional network security and remain vital for IoT intrusion detection.

Raza et al. (2013) [57] examined security in 6LoWPAN IoT networks, focusing on detection in RPL-enabled sensors. SVELTE detects spoofed data, sinkhole, and forwarding attacks by analyzing network inconsistencies with 6Mapper and data from the Border Router. Routing graph checks, supported by a mini-firewall, help identify attacks. 6Mapper requests data and analyzes every 2 min; this reflects periodic rather than continuous monitoring. Latency isn't reported; earliest detection is after 4 min, referring to analysis time post-setup, not per-event latency. SVELTE needs 1.76 KB ROM, with overhead increasing up to 4724 bytes (about 4.7 KB). SVELTE modules require an additional 0.365 KB of RAM. Energy use for 6Mapper response handling (0.1465 mJ), firewall processing (0.0478 mJ), and packet loss correction (0.0483 mJ) remains well within thresholds; computer requirements such as FLOPs are not reported. SVELTE achieves 90% TPR against sinkhole attacks and nearly 100% against selective forwarding, though false alarms affect TPR. It can correct faulty data or remove malicious nodes, deleting nodes after repeated detection. Detection uses a simple threshold without online or incremental learning, yet the system is extensible to new schemes. Experiments used Contiki's Cooja simulator, which is not a complete physical testbed, with node counts up to 16 for ROM calculations; larger network data isn't quantified. No throughput tests are reported.

Yin, Kang, and Kim (2016) [58] present a rule-based intrusion detection system using CEP via the Esper engine, with EPL rules to monitor IoT events across layers. It aims to identify real-time anomalies, distinguishing malicious attacks from benign issues like sensor faults or delays. The IDS offers real-time monitoring of event streams on the Esper engine, indicating continuous per-event detection. However, the study does not report latency, resource usage, or standard evaluation metrics. It mentions detecting abnormal

events and transmitting them to an alert processor, but it does not describe any automated actions or mitigation strategies. The model uses predefined EPL rules without online learning, drift detection, or automatic threshold adjustments. It does not specify the attack types it can handle or the datasets used, and the evaluation is based solely on datasets or simulations. The authors plan to test the model in real IoT environments in the future, indicating that current work is a concept without real-world validation. The paper provides no scalability or performance metrics, limiting the assessment of practicality. Its strengths are real-time processing and an event-driven architecture, making it well-suited to distributed IoT systems, but its lack of empirical validation limits its applicability. Further development and testing in realistic scenarios are suggested.

Fu et al. (2017) [59] developed a reliable IDS for resource-limited IoT networks. They expanded the Input-Output Labeled Transition System (IOLTS) to track communication deviations for anomaly detection. The framework includes an Event Monitor, Database, Analyzer, and Response Unit, which are mainly implemented within IoT gateways and detect jamming, false, and replay attacks, and is tested with simulated RADIUS traffic. The author claims the design can support real-time detection, although packet logging suggests a batch-oriented workflow. No latency, model size, RAM, FLOPs, or energy metrics are provided. The IDS detects three attack types and verifies them via a GUI, but provides no quantitative performance metrics. The Response Unit reports risks or triggers alarms, not automated mitigation. The architecture relies on a static automata model, with no mention of online learning or adaptive thresholds; future plans include improving fuzzy methods for library creation. The system does not handle zero-day threats, APTs, or complex attack patterns, and deployment details are limited to a lab environment. No scalability or load handling metrics are provided. Detection is visually verified, with limited attack coverage and manual event translation. Future work aims to expand protocol support, improve fuzzy matching, and develop better evaluation methods.

Haripriya and Kulothungan's 2019 paper [60] presents an MQTT IDS using fuzzy rule interpolation (FRI) to generate detection rules. The MQTT broker runs an anomaly detection algorithm that analyzes MQTT packets to decide whether to accept or reject them, responding immediately to attacks. While claiming to be lightweight, the paper lacks specific metrics on latency, model size, RAM, FLOPs, or energy per inference. Accuracy metrics are fragmented across scenarios, making overall performance difficult to interpret. The system includes autonomous mitigation: when an anomaly is detected, the broker automatically accepts or drops packets without human intervention, integrating detection and action via a fuzzy inference engine. The model detects DoS attacks characterized by high request rates, but it doesn't discuss zero-day threats, botnets, or adaptive learning. Experiments are simulated in IoT networks ranging from 60 to 500 nodes, with 10% malicious nodes, indicating low-to-moderate scalability. The authors suggest expanding the rule base and testing in broader environments. Overall, the setup indicates a move toward moderate-scale IoT IDS development.

Conventional IoT intrusion detection methods, such as rule-based systems, automata, fuzzy logic, and lightweight optimization, are useful in constrained environments due to their transparency, predictability, and low resource use, making them ideal for MCU devices and latency-sensitive control. Yet, reviews show these systems have limited attack coverage, struggle with zero-day/multi-stage threats, and lack adaptive learning. They often rely on periodic analysis and small testbeds, which limit real-world confidence. Scalability is rarely assessed, and metrics frequently fall short of standards. As IoT expands, these limitations highlight the need for more adaptive, learning-based models. While simple models suit MCU constraints, they reduce feature expressiveness and effectiveness against stealthier threats. 'Non-real-time' refers to periodic rather than continuous

detection, which affects response time. Scalability tests are rare, mainly evaluated in small simulations that don't reflect real complexities.

4.2. Machine Learning IDS Models

Building on traditional rule-based and heuristic IDS, lightweight ML models such as DT, RF, SVM, and Naïve Bayes offer greater flexibility and detection performance while remaining efficient for IoT devices. They have low computational overhead, enabling fast inference on edge hardware. This section reviews these ML models in terms of scalability, accuracy, latency, and suitability for IoT security. Claims of “real-time” or “adaptive” capabilities without specific metrics are considered tentative and are evaluated cautiously based on available evidence.

Prabavathy et al. (2018) [61] underscore the importance of rapid intrusion detection in large-scale IoT deployments, particularly when cloud systems encounter latency challenges. They advocate a fog computing framework that employs the Online Sequential Extreme Learning Machine (OS-ELM) across distributed fog nodes to classify high-speed IoT traffic. OS-ELM updates its weights in real-time by processing incoming packets and uses a recursive least-squares algorithm to compute the most recent hidden-layer outputs and output weights. Fog nodes can detect attacks approximately 25% faster than cloud systems, though specific latency metrics are not provided. The model attains accuracy rates of 97.36% (binary classification) and 96.54% (multi-class classification), surpassing established thresholds. The results are transmitted to the cloud for additional analysis, prediction, and response, indicating support for detection (L0) and possibly recommendation-level assistance (L1) but not autonomous mitigation. Detection is performed at fog nodes, with the cloud providing support for further analysis and predictive capabilities. Mitigation procedures are conducted separately and are not integrated into the system workflow. The online learning capability of OS-ELM facilitates swift adaptation to the dynamic nature of IoT environments. Evaluation was conducted using the NSL-KDD dataset, with the system capable of detecting various attacks, including DoS, Probe, R2L, and U2R, via distributed detection at fog nodes. Future research could explore predictive analysis of attacker movements and testing against zero-day exploits or botnet datasets. It is important to note that the system has been tested solely on the NSL-KDD dataset and not on real IoT devices or live traffic; specific scalability data remain unspecified.

In their 2019 study, Jiaqi Li and colleagues [62] introduce a hybrid AI-driven IDS for SD-IoT with two main phases: feature selection and classification. They use an improved Bat Algorithm (BA) with swarm division and binary differential mutation to reduce feature dimensionality while maintaining high discriminatory ability. The classification employs a modified RF with adaptive weighting and voting to robustly detect minority intrusions. Their experiments use a downsampled version of the [KDD Cup 1999 dataset]. The system is evaluated offline, and no latency measurements are provided. It discusses processing time and acceptable time consumption, but without quantitative data such as model size, RAM, FLOPs, or energy per inference. Performance is evaluated, achieving 96.03% accuracy, 99.51% precision, 95.17% recall, and 97.29% F-score. The false alarm rate is 0.98%, and the misclassification cost is 0.1302. Detection accuracy for minority intrusions improves. The paper does not detail automated responses, policy actions, or mitigation strategies. Integration with automated mitigation is conceptually mentioned, but not implemented or tested. The system aims to detect novel intrusions using self-learning, enhancing RF by adaptively adjusting sample weights and balancing the training set, thereby enabling online learning. No mechanisms for drift detection or auto-threshold adjustment are discussed. The authors claim the system can detect novel intrusions and handle unbalanced datasets with higher costs for minority classes, but do not evaluate against zero-day attacks, botnets, or long-term APT modeling. Evaluation relies on offline

experiments using the KDD Cup 1999 dataset, with no validation on real devices, complete testbeds, or diverse hardware. Scalability, real-time testing, and energy efficiency metrics are lacking. Limitations include reliance on outdated datasets and potential overfitting, with suggestions for deployment in real-world environments.

Zachos et al. (2021) [63] present a lightweight anomaly-based IDS for IoMT using ML classifiers (DT, RF, KNN) to reduce computation on edge devices. It combines host- and network-based monitoring but detects on collected data, not continuous inference, providing periodic rather than continuous real-time monitoring. Metrics such as latency, model size, RAM, FLOPs, and energy are not reported. Despite limitations, the IDS excels on the TON_IoT dataset, with models exceeding 99% accuracy, precision, recall, and F1-scores, meeting thresholds. However, it lacks autonomous mitigation, detection, and response, as well as adaptive learning. The dataset contains diverse IoT telemetry and threat data, but doesn't include evaluations of zero-day or botnet attacks. It relies on offline data without hardware testing or scalability analysis. Zachos et al. (2022) [64] extend this by developing a prototype with Monitoring and Data Acquisition (MDA) on devices and a Central Detection (CD) engine at the gateway, thereby improving the architecture but still enabling only partial real-time detection at fixed intervals. No latency or resource metrics are provided, nor is any ML accuracy data reported. Mitigation is alert-based, lacks autonomy or adaptive features, and has unspecified attack coverage. Validation is limited to a lab prototype, with no scalability assessment. Both studies show progress toward practical IDS for IoMT gateways, but they lack real-time detection, resource metrics, autonomous mitigation, adaptive learning, scalability, or real-world validation.

Vishwakarma and Kesswani (2023) [65] tackled the challenge of accurately detecting intrusions in IoT systems with diverse data types and imbalanced datasets. They propose a two-phase ML-based IDS that classifies data by type and employs variations of Naïve Bayes with majority voting for final classification. In the second phase, benign data are analyzed using an unsupervised elliptic envelope to identify anomalies, combining supervised and unsupervised methods to improve accuracy. The system currently operates offline with batch inference, and real-time capability remains future work. The model achieves 97% accuracy on NSL-KDD, 86.9% on UNSW_NB15, and 98.59% on CIC-IDS2017. It also mentions a "meager false positive rate," though no exact percentage is provided. F1-score, precision, and recall are not reported. The system did not evaluate detection latency or resource usage on constrained IoT hardware. Presently, it is detection-only, with mitigation noted as future work. The paper describes a static model trained on datasets with no mention of online or incremental learning, drift detection, or auto-threshold/policy adjustment. UNSW_NB15 is defined as a "comprehensive dataset for network intrusion detection systems" that may include sophisticated attacks. However, specific F1-scores for zero-day attacks, tests against botnets or APTs, or temporal and correlation modeling are not provided. Validation was conducted using these standard datasets; there is no mention of real devices, full testbeds, diverse hardware, or traffic generated from a deployed environment. The paper also fails to discuss scalability experiments, throughput tests, or stability under load with varying numbers of nodes or traffic volumes.

Alosaimi and Almutairi (2023) [66] propose an IDS for IoT using the BoT-IoT dataset to detect DoS, DDoS, and botnet attacks across TCP, UDP, and HTTP, employing supervised ML algorithms such as DTs, KNNs, SVMs, Linear Discriminant Analysis, and ensemble bagging. They state that monitoring network traffic with ML algorithms can quickly detect and respond to threats. The approach thoroughly examines IoT traffic to identify intrusions and abnormal behavior. The reported "Testing Time (s)" reflect total evaluation duration rather than inference latency, and metrics such as model size, RAM, FLOPs, and energy per inference are not provided. Performance metrics meet or exceed

thresholds, with DT, Ensemble Bag, and KNN achieving 100% across metrics and a 0% Error Rate. The system can isolate affected devices or block malicious traffic, aligning with policy (L2), but it's not described as a closed-loop with rollback (L3). Detection and mitigation are mentioned separately but not integrated into a single workflow. The classifiers are adaptive and can be updated continuously to counter emerging threats, implying online learning. The study does not mention drift detection or automatic threshold adjustments. It uses the BoT-IoT dataset, generated to simulate realistic network environments, including DDoS and DoS attacks and the Mozi botnet. However, the approach is based on simulation rather than real devices and lacks scalability or throughput testing. Overall, it advances ML-based IoT intrusion detection but needs further work for real-time deployment.

Fadhilla et al. (2023) [67] focus on detecting effective IoT BotNet attacks in resource-limited settings, such as small office/home office (SOHO) networks. They introduce a meta-learning ensemble that combines weak learners, RF, LR, DT, Naive Bayes, and MLP, with MLP serving as the meta-classifier. The deep packet inspection examines each packet, enabling continuous detection. Inference time is under 3.95 s on devices like the Raspberry Pi 4, which exceeds the accepted threshold for edge devices. Model size, RAM, FLOPs, and energy use aren't detailed. Models achieve over 97.9% accuracy with an FPR below 3.8%, with higher accuracy and lower FPR on datasets such as IoT-23 and KDD99. The architecture involves NIDS detection and relies on a firewall for mitigation, with these components not integrated into a single system. The model is static, deployed as pre-trained without online updates. It is tested on datasets including BotNet IoT-23, KDD99, and TON, covering various attack types. The model detects minor differences between normal and malicious flows. Evaluation was conducted on a Raspberry Pi 4 as an AIoT gateway, primarily using recent attack datasets rather than live traffic. Scalability and throughput are not reported, and the system's performance under increased load is not discussed. Limitations include tuning needs, limited multi-class testing, and dependence on firewalls.

Tahir et al. (2024) [68] examine IoT cybersecurity, focusing on ML-based anomaly detection with adaptive defenses. They propose a framework that uses ML algorithms such as RF, SVMs, DTs, and Gradient Boosting, mainly for supervised learning, to detect irregular patterns and security breaches in real-time. However, there's no detail on continuous, streaming detection, and resource efficiency isn't reported. Random Forest achieved 81% accuracy, 82% precision, and 81% recall, with some class-specific precision issues. SVM performed poorly with 54.72% accuracy and 30% precision. Decision Trees scored similarly to RF with 81.65% accuracy and 82% precision. Gradient Boosting showed the best results, with 89% accuracy, 91% precision, and 89% recall, slightly below the 90% threshold. FPR isn't reported. The paper aims at adaptive defenses but discusses only conceptual mechanisms, without detailing automated or closed-loop actions. There's no architecture or process explaining how detections trigger responses. The models are trained offline and tested on various attack types, but no datasets are named, and there is no real-world testing or scalability assessment. Reproducibility is limited due to the unavailability of datasets. The study highlights overfitting and suggests future work on ensemble learning, efficiency, and adaptive defenses in evolving IoT settings.

Lightweight ML models such as DTs, RFs, Naive Bayes, SVMs, and neural classifiers are popular for IoT intrusion detection due to their low cost, fast inference, and edge deployment. They often achieve high accuracy with limited features. Challenges include a lack of real-time detection, data scarcity, missing features such as autonomous mitigation and online learning, and limited testing with real data or against advanced threats like zero-days and APTs. Validation mainly relies on outdated or synthetic datasets, reducing reliability. These models need improvements in real-time operation, resource profiling,

adaptability, and scalability. The main barriers are missing system reporting (latency, RAM, energy) and offline validation, complicating deployment verification. When low latency isn't achieved, it's due to feature extraction and centralized processing, not the classifier. Adaptability is overstated; many studies describe the "self-learning" conceptually but use static offline training without drift monitoring. These factors explain why ML performs well on benchmarks but less in real-world applications.

4.3. Deep Learning IDS Models

Deep learning (DL) models offer greater representational capacity than conventional ML by enabling automatic feature learning and modeling of nonlinear and temporal patterns. In the reviewed literature, this capacity is often reflected in high detection accuracy and improved performance on complex attack patterns.

Diro and Chilamkurti (2018) [69] examine IoT fog attack detection, where traditional ML struggles. They propose a DL-based distributed IDS that uses DNNs and stochastic gradient descent for parameter sharing across fog nodes. They reference real-time capability, but do not specify whether the system performs continuous packet-level inference. The paper discusses features, layers, neurons, batch sizes, and epochs, but not model size, RAM, compute, or energy use. The model achieves 99.20% accuracy for 2-class and 98.27% for 4-class detection, with high F1, precision, recall, and an FPR of 0.85%. It detects attacks without automation or mitigation functions. Parameters are shared to prevent overfitting, but no mention of online learning, drift detection, or adaptive thresholds; it appears static after training. Evaluation uses unseen NSL-KDD data to simulate zero-day attacks. The model's R2L and U2R classes get an F1 of 80%, but the impact on other attack types isn't detailed. The study highlights deep learning for feature extraction and sharing, but not temporal or correlation modeling. The dataset is offline, and the model outperforms centralized versions, with accuracy improving as more nodes are added, claiming better scalability but lacking detailed thresholds. Limitations include resource-analysis gaps, long training times, and the lack of autonomous mitigation, all of which pose security challenges.

Saba et al. (2022) [70] propose a CNN-based intrusion detection model trained on NID and BoT-IoT datasets to protect IoT ecosystems through anomaly detection. The model was trained and tested offline, lacking data on latency, model size, RAM, FLOPs, or energy consumption. While accuracy is high (NID: 99.51%, BoT-IoT: 95.55%), key metrics such as F1-score, precision, recall, and FPR are missing. The study focuses solely on detection and does not cover automated responses or mitigation. It uses fixed datasets without online or incremental learning, drift detection, or auto-thresholds. The model was tested on botnet datasets, including DDoS and DoS attacks. The approach does not incorporate temporal modeling or cross-flow correlation techniques. The CNN processes TCP/IP features without explicitly addressing advanced attack patterns. Despite claims of predicting new attacks with deep learning, no zero-day F1-scores are presented. The study does not include testing on real devices, live traffic, scalability, throughput, or hardware diversity. Future work will consist of hyperparameter tuning, diverse datasets, and variations on CNN architectures.

Vishwakarma and Kesswani (2022) [71] introduced DIDS, a real-time anomaly-based IDS for IoT networks using a DNN with batch normalization and dropout to enhance robustness and prevent overfitting. They state it can identify malicious packets and detect attacks in real-time, while capturing packets for analysis. While it mentions real-time detection, it lacks latency metrics for devices. The system considers IoT resource constraints and emphasizes lightweight, energy-efficient solutions, but provides no metrics for model size, RAM, FLOPs, or energy per inference. The model achieves high accuracy: 99.21% for binary classification and 99.08–99.48% for multiclass across multiple datasets, with F1-

scores from 99.02% to 99.79%, precision from 99.03% to 99.48%, and recall from 99.08% to 99.48%. The false positive rate is likely below 5%. The system focuses solely on detection, sending alarms without mitigation or response mechanisms, and lacks online learning or adaptation features. Evaluation on various datasets shows effectiveness against modern attacks, with tests on real IoT hardware demonstrating deployment feasibility. However, the study provides limited data on scalability, latency, and energy efficiency, and the system does not address zero-day threats or automatically adjust thresholds. Future work aims to improve threat coverage, online training, and scalability.

Khan et al. (2022) [72] focus on malware detection in IoT text data, where traditional DL methods require many resources. They introduce lightweight models such as RNNs, LSTMs, and Bi-LSTMs with simple architectures: a single dense layer and a few thousand parameters. The paper aims for real-time monitoring, but current work uses offline data without latency measurements. The models have a small footprint suitable for Node or Gateway devices, with low FLOPs and just a few thousand or hundreds of parameters. No specific FLOPs, RAM, or energy consumption data are provided. Test accuracy reaches 0.9946 (RNN), 0.9681 (BiLSTM), and 0.9631 (LSTM), with claims of up to 99.45% accuracy, surpassing baselines. The metrics, such as precision, recall, and F1-score, are summarized, and all meet the identified threshold. However, the study only focuses on malware detection, not on response or mitigation strategies. Evaluation is based on static models trained offline; online learning or concept drift detection are not discussed. They use MalwareTextDB for classification but do not test zero-day attacks, botnets, APTs, or model responses to temporal threats. The authors outline future deployment plans, but no real-world testing has been conducted yet. There is no information on scalability, throughput, latency, runtime, or energy consumption, which highlights limitations in responsiveness and deployment feasibility.

Idriss Idrissi et al. (2022) [73] created a Deep Learning-based Host Intrusion Detection System (DL-HIDS) for resource-limited IoT environments, using optimized CNNs with pruning and quantization. The system enables faster edge inference and near real-time traffic analysis via fog nodes. Traffic is pre-processed into an image-like representation for CNN-based analysis. Inference takes no more than 1 μ s, well below device thresholds. Model sizes include 343 KB for the original and 2704 bytes for the tflite version. Memory limitations, like those on Arduino UNOs, prevent some devices from running models. Inference on the Raspberry Pi and other devices is as low as 1–2 μ s. Hardware specs confirm resource availability. High accuracy (up to 99.74%) supports resource efficiency, though other metrics are absent. The system detects breaches but lacks automated mitigation. Models are tailored to device resources, with no online learning or adaptive policies. Evaluated on the MQTT-IoT-IDS2020 dataset, but does not address zero-day attacks or long-term adaptation. Deployment across various devices demonstrates real-world use; however, it is primarily with offline datasets. Scalability and heavy-load performance are not discussed, focusing on individual device deployment. The approach lowers costs by enabling local inference but introduces scalability and security challenges. Future work aims to optimize accuracy, latency, and size, emphasizing deployment feasibility on resource-limited devices while acknowledging current limits.

Fang et al. (2024) [74] introduce IRCNN-CBAM, a lightweight DL model for IoT intrusion detection. By combining Depthwise Separable Convolution, IRCNN, and CBAM, it reduces parameter count and computational cost, making it suitable for resource-limited devices. They describe training and testing on a pre-processed dataset to improve response time and network intrusion detection as proactive defenses. However, the paper evaluates only offline test performance and does not describe an online detection process. They report response time and mention that a slight increase in parameters to reduce detection time is worthwhile, providing training time per epoch, but not detection latency.

The model has 23.66k parameters and fits devices such as Node (≈ 94.64 KB) and Gateway (≤ 50 MB). It is designed for resource-constrained IoT devices that lack high-performance chips, while still meeting necessary thresholds. The paper does not report FLOPs, RAM usage, or energy consumption per inference. Metrics include Precision 0.9652, Recall 0.8810, and FPR 0.0020, all of which meet the specified thresholds. However, F1-score and overall accuracy are not reported. Although detection performance is evaluated, mitigation strategies and integrated response are not discussed. The model is trained offline on the CIC-IDS-2017 dataset, noting data imbalance and planning to use generative models in the future to address it. It shows improved detection for some attack types but not all, due to imbalance. No testing on zero-day attacks or temporal modeling is mentioned. Scalability and stability under load are not addressed. Despite faster training compared to DNNs, detecting minority attacks remains difficult.

Adel Binbusayyis (2025) [75] proposes a DL intrusion detection method that combines UN (Unify Net) and NSSN (Neutro Sequential Sense Net) to classify multi-class attacks in IoT networks, using CNNs and RNNs. UN models temporal dependencies like LSTM, while NSSN trains four sequential pathways for feature extraction. The dataset is split into 80% for training and 20% for testing, indicating an offline batch-processing approach rather than real-time detection. No latency metrics are provided. The paper notes computational resource needs but lacks quantitative metrics such as model size, RAM usage, FLOPs, or energy consumption. The model achieved 0.99% accuracy, precision, F1 Score, and recall, with AUC values near 1, indicating high performance; FPR is not explicitly reported but inferred to be low. Focus is solely on attack detection and classification, with no discussion of mitigation, online learning, drift detection, or adaptive thresholds. Tested on Kaggle IoT logs containing DDoS, probing, scanning, and Man-in-the-middle attacks, the model claims early threat detection and the handling of ambiguous data through deep feature extraction, but it does not address zero-day attacks or generalization. Data was collected from physical IoT devices using ultrasonic sensors, Arduino, and NodeMCU. The model is trained offline and not evaluated in continuous or real-device settings. Scalability, robustness under load, and resource impact are not discussed, though computational resource needs are acknowledged.

DL-based IDSs effectively detect attacks and analyze traffic, but often lack real-time guarantees and practical deployment in IoT environments. While they excel at detection, they need model compression, lighter pipelines, and hardware support for real-world edge applications. Their complexity leads to failures in meeting real-time and lightweight criteria, often due to offline testing rather than actual streaming. Studies use compression and simplified architectures, but many lack validation of energy and throughput, showing DL's strength in accuracy but weakness in deployment unless specifically designed for edge constraints.

4.4. Hybrid IDS Approaches

Hybrid IDSs combine traditional, ML, and DL methods to enhance IoT security, addressing the weaknesses of standalone techniques. In the studies reviewed, hybrid IDSs often combine multiple detection methods, such as signature-based and anomaly detection, with techniques including federated training, device clustering, SDN-powered control, and optimization strategies. These combinations aim to improve accuracy and overall effectiveness. Reinforcement learning is applied more selectively, typically for control or mitigation tasks in specific cases, rather than as a general part of hybrid systems. These systems manage diversity, real-time demands, and limited resources through multi-layered architectures, edge/cloud collaboration, and smart feature selection. They represent a step toward scalable and context-aware intrusion detection that better aligns with the complexity of modern IoT ecosystems.

Sedjelmaci et al. (2016, 2017) [76,77] propose a hybrid IDS designed for resource-limited IoT and WSN environments. It combines signature-based detection with anomaly detection, activating modules selectively using game-theoretic strategies, such as Nash equilibrium, to balance security and energy consumption. Detection is triggered periodically or when predefined conditions are met, rather than continuously. Neither study provides data on latency, model size, RAM usage, FLOPs, or energy per inference, making it challenging to assess resource efficiency. Reported accuracy exceeds 90%, with a false positive rate around 2%, but details on F1-score, precision, and recall are absent. The system can detect and “eject” malicious nodes, but this is limited to detection-only functions, lacking autonomous or integrated mitigation. Both studies include adaptive features via rule updates for new attack patterns, but neither implements formal drift detection or automated policy adjustments. The evaluation is solely conducted via simulation in TOSSIM, with no testing on real hardware, limited device variety, or realistic traffic conditions. Scalability is constrained to approximately 300 nodes, falling short of requirements for moderate or large-scale validation. While promising for low-energy IoT devices, the approach is still limited by the lack of real-time assurance, detailed resource metrics, adaptive mechanisms, advanced attack response, and real-world deployment.

Mudgerikar et al. (2020) [78] highlight the limitations of traditional network IDSs in detecting advanced IoT malware, such as fileless attacks. They introduce E-Spion, a hybrid, anomaly-based, system-level IDS that combines signature-like whitelisting, heuristic profiling, and ML-based anomaly detection, operating in a device-edge architecture. It uses system data and ML classifiers such as RF in a three-layer scheme for real-time detection, collecting logs locally and periodically transferring them to the edge server for window-based processing. While designed for resource-constrained IoT devices with low overhead (4.3% CPU, 2.6% RAM), specific latency and model size data are not provided. The system achieved high detection accuracy: 97.64–97.75% in the PBM layer, 100% in SBM, and overall detection above 78%, 97%, and 99% across layers. The top PBM classifiers had an F1-score of 0.97, with a false-positive rate of 3.32–8.5%, while SBM recorded zero false positives. The IDS detects anomalous behavior based on learned profiles stored in the cloud, but it does not incorporate online learning or mechanisms to adjust thresholds automatically. It was evaluated using 3973 malware samples across eight fileless attack types in a typical enterprise IoT setup, but details on hardware diversity and traffic realism are limited. Scalability assessments are absent, and limitations include assumptions about benign initial behavior, no real-time retraining, reliance on a stable internet connection, and a lack of public datasets. Future work suggests integrating with network IDS for better detection, though scalability challenges persist.

Holubenko and Silva (2023) [79] propose an IoT intrusion detection system that uses system call analysis, ML, and FL, with classifiers such as RF, KNN, and MLP, for resource-constrained devices. It gathers system call traces, classifies inputs in real-time, and issues alerts for intrusions. The system claims low detection latency but does not specify metrics, nor metrics for model size, RAM, FLOPs, or energy consumption. The best model (RF) achieved 99% accuracy, precision, and recall, a 1% false-positive rate, and a 99% F1-score. It detects intrusions and issues alerts, but lacks autonomous mitigation. Models require frequent updates, suggesting online learning, but there are no details on drift detection or threshold adjustments. Focusing on system call classification can help detect zero-day threats, but the paper does not report performance for zero-day or botnet detection. Experiments on three VMs emulate target-device resources using ARM64 and QEMU 6.1, but this emulated environment does not fully capture the diversity or constraints of physical IoT devices. Although migration to monitoring servers is suggested, no scalability or throughput metrics are provided. Future work will explore deep learning models, including CNNs, RNNs, and LSTMs.

Talpini et al. (2023) [80] combine ML with FL for IoT intrusion detection, employing entropy- and similarity-based clustering techniques. Their three-tier architecture categorizes IoT devices by entropy similarity, with data processed “per flow” in near real-time; however, specific latency and resource metrics are not provided. The proposed approach enhances detection performance, yielding up to a 17% increase in F1-score compared to conventional FL, with results approaching those of centralized methods. Nonetheless, explicit F1-score values or other performance metrics are not disclosed, and the study does not address automated responses or adaptive models. Models are developed through federated learning combined with clustering, aiming to generalize to new devices, although online learning and concept drift detection are not incorporated. The methodology was tested on the CIC-ToN-IoT dataset, which encompasses nine attack types, suggesting potential robustness against unknown threats; however, its ability to detect zero-day attacks has not been demonstrated. The experiments were confined to simulations, with limited details on scalability, and clustering was performed randomly. Future research aims to enhance the methodologies, validate findings across additional datasets, and refine federated learning aggregation procedures.

Grigoriadou et al. (2023) [81] developed a hybrid AI system to detect and mitigate cyberattacks, including flooding and RTSP brute-force attacks. It used DL with an MLP (2–4 hidden layers) for intrusion detection and an SDN controlled by a Q-Learning agent for mitigation through rate limiting and rerouting. The system detects malicious activity in real-time by analyzing network traffic data and flow statistics using pre-trained AI models, enabling continuous online inference. However, the study provides minimal information on latency or resource use. The MLP with 4 hidden layers achieved high accuracy, TPR, F1-score, and low FPR, although precision was not specified. The Mitigation and Response Module uses detection results to direct SDN actions, relying on an offline-trained Q-Learning agent for policy-driven mitigation. Detection employs DNNs, SDN, and Q-Learning within a single IDPS framework that directly initiates mitigation actions. The paper mentions pre-trained models but does not discuss online learning or adaptation to evolving threats. It focused on flood and brute-force attacks and trained on the CIC IoT Dataset 2022, but makes no claims about zero-day attacks or generalization. Evaluation was based on this dataset and flow statistics, with no mention of real testbeds, diverse hardware, or extensive scalability testing. Future work aims at improved generalization and real-time adaptive mitigation, but robustness across different networks remains unexplored.

Fenanir and Semchedine (2023) [82] introduce an edge-FL smart intrusion detection system for IoT using DL models, DNN, CNN, and LSTM, trained locally with FedAvg. Evaluated on IoTID20, IoT-23, and N-BaIoT datasets, the system performs offline analysis rather than real-time detection. Although claims are made that federated learning reduces latency and communication, no specific metrics are provided. Details on model size, resource use, or energy consumption are absent, but LSTM achieved up to 99% accuracy. While high precision, recall, and F1-scores are noted, the highest F1 isn’t specified, though similar systems on IoT-23 score up to 0.99. The focus is on attack detection and classification, with no mention of automated responses or mitigation. The architecture emphasizes federated training without real-time adaptation or online learning. The evaluation relied on dataset-driven experiments with limited hardware diversity. Future work plans include additional models and the integration of blockchain.

Majjaru and Senthilkumar (2023) [83] introduce HOPNET, a hyperparameter-optimized neural network for IoT intrusion detection. It combines a hybrid DL method with a frog fitness algorithm to improve feature extraction and attack classification (DoS, R2L, U2R, probes) on the NSL-KDD dataset. The paper highlights HOPNET’s low time consumption, indicating its efficiency for real-time IoT intrusion detection, with a detection

time of 18,457 ms, which is too slow for edge or node-level real-time deployment. Metrics like model size, RAM, FLOPs, and energy are not reported. HOPNET achieves a detection rate of 97.38% with high precision above 0.90, outperforming other models (DBM, RNM, DNM, CNM). Sensitivity (recall) is also high, though F1-score and FPR are not reported. The focus is solely on intrusion detection; no automation or response mechanisms are discussed. HOPNET's optimization, guided by Frog fitness, improves prediction accuracy, though details on online learning or auto-thresholding are not provided. On the NSL-KDD dataset, it detects various attack types, including sophisticated U2R and R2L attacks. Its adaptability suggests it is effective against evolving threats, but no real-world deployments or hardware tests are mentioned. Scalability and throughput data are absent. Overall, the model improves speed and tuning while accounting for some resource considerations, but it faces deployment challenges.

Abusitta et al. (2023) [84] focus on detecting unusual activities in noisy, incomplete IoT data using a hybrid DL and ML framework: a denoising autoencoder extracts features, which are processed by a neural layer and then classified by SVMs or LR. When evaluated on datasets, it supports offline batch processing rather than real-time detection. The model achieved 94.9% accuracy on the BoTNeTIoT-L01 dataset, and 94.6% on DS2OS traffic traces. Other metrics, such as F1, Precision, Recall, or FPR, are not detailed. Latency and resource efficiency are also not discussed. The focus is solely on anomaly detection, with no mention of automated actions or mitigation strategies, nor of an integrated detection-mitigation workflow. The model operates as a static pre-trained system without adaptation capabilities. Evaluation used IoT datasets, likely containing botnet traffic, with plans to improve robustness against adversarial attacks. No claims are made regarding handling zero-day attacks or APTs. Testing was limited to datasets; testing on real devices or in live environments is not reported. Scalability is not addressed.

Alalhareth and Hong (2024) [85] address the security of resource-limited IoMT devices that are vulnerable to cyber threats. They propose a meta-learning-based ensemble IDS combining CNNs, RNNs, and autoencoders, with a meta-learner that dynamically assigns weights based on accuracy, loss, and confidence. This system enhances the detection of known and new threats, trained on WUSTL-EHMS-2020. It processes data in micro-batches with inference times of 20–50 ms, making it suitable for resource-constrained environments at the node or gateway level. The ME-IDS model reduces CPU utilization (41–57% during training, 20–30% during inference) and memory use (45–49% of RAM), outperforming other models such as Stack-IDS, DIS-IoT, and EDL-IDS in resource efficiency. It achieves high accuracy (up to 0.980), detection rate (up to 0.970), and F1-score (up to 0.996), although the reported FPR (0.101) is higher than desirable for an operational environment. Focused solely on intrusion detection, the system uses a self-optimizing ensemble that adapts to changing data and threats, combining multiple detection techniques to improve identification of novel attacks. It employs anomaly and signature-based detection to handle zero-day attacks, though it lacks specific testing on botnet datasets or for network scalability beyond dataset simulation. The evaluation shows superior detection performance in offline experiments but provides no information on real-device testing or stability under heavy traffic.

Deng's 2024 study [86] presents an industrial IoT intrusion detection system combining LightGBM with an MLP to improve accuracy and efficiency. The hybrid model combines LightGBM's fast feature selection with MLP's nonlinear modeling. It trains and tests on an 80/20 split, indicating offline batch evaluation rather than continuous detection. The detection time for LightGBM is 1.01 s, exceeding the thresholds for node (≤ 10 ms) and gateway/edge (≤ 50 ms) devices, placing it outside the acceptable range for real-time IoT detection. No model size, RAM, FLOPs, or energy info is provided. Performance metrics show 96.2% accuracy, 97.3% intrusion prediction, 96.5% F1 Score, and 95.6% recall, with

a high implied precision. The false positive rate isn't reported. The study focuses only on intrusion detection without automation or responses. The approach relies on static training on fixed datasets, with no online learning or adaptive measures. It doesn't specify attack types or test against sophisticated threats such as zero-day exploits, botnets, or APTs. Datasets are generic IDS benchmarks, with no real-device testing or scalability metrics. While effective on benchmark data, the model lacks real-world validation.

Thiruvankatasamy et al. (2024) [87] introduce BAFWO-MLIDS, a blockchain-based IoT healthcare intrusion detection system combining Fireworks Optimization (FWO), an Elman Neural Network, and Bayesian Optimization to provide secure, tamper-proof logs. Tested on the BoT-IoT Database, it supports offline, batch, or retrospective analysis, but does not mention online inference or streaming. However, no latency or resource-efficiency data are provided. The system achieved high metrics: for an 80:20 split, accuracy 99.86%, precision 99.02%, recall 99.38%, and F-Score 99.19%. For a 70:30 split, accuracy 99.62%, precision 98.24%, recall 97.61%, F-Score 97.91%. All metrics surpass thresholds, though FPR is not explicitly reported. The paper highlights "enhanced security" via "secure data transmission" and "intrusion detection" but lacks details on automated mitigation or response. Labeled as an IDS, it lacks an integrated mitigation component. The methodology involves a three-stage process: feature selection, detection, and parameter optimization, with no discussion of online learning or adaptive thresholds. The evaluation used the BoT-IoT Database, which includes attack types such as DoS, DDoS, scanning, fingerprinting, and keylogging, but does not mention zero-day attacks or advanced persistent threats. Since the assessment was offline, there's no evidence of deployment on real IoT devices or testing with real-device traffic. The study used 33,673 samples but doesn't address scalability or performance under different loads. Although blockchain increases security, it raises concerns about overhead and scalability in resource-limited IoT environments.

Adekunle et al. (2024) [88] propose a hybrid intrusion detection framework for IoT that combines deep learning, metaheuristics, and generative modeling, and they test it on three datasets. The paper discusses IDS and its effectiveness, but does not mention continuous, online, per-packet processing. Evaluation on datasets such as Bot-IoT, CICIDS2017, and CICIDS2019 indicates batch analysis, with no reported metrics regarding low latency or resource efficiency. The models used (DenseNet201, RAPNet, CGAN) are resource-intensive; however, no specific details are provided. The feature selection strategy achieves high accuracy levels: BoT-IoT 97.95%, CICIDS2017 97.98%, CICIDS2019 99.11%. Future research plans include developing an end-to-end system with intrusion prevention and mitigation. Still, the current models are static and have not been tested for deployment on real devices. Although the dataset includes multiple attack categories, no evaluation of generalization or zero-day detection is conducted. The evaluation is limited to dataset analysis, with no assessments of scalability or throughput.

Hybrid IDSs enhance detection and communication via layered cooperation and modular architecture, tackling IoT heterogeneity and non-IID data. They achieve high accuracy and faster processing but are mainly tested offline on small setups, with limited real-time, scalability, and cost evidence. Mitigation remains conceptual primarily, with few empirical demonstrations. Responses fall into two: (i) mitigation actions within workflows, like detection triggering SDN policies; and (ii) detection-only, with mitigation through alerts, external controls, or future work. Future research should focus on safe, co-designed detection and response, lightweight models, benchmarks, and open testbeds. Hybrid systems are promising for task distribution and complementary detectors, but they inherit the weaknesses of their individual components and face integration challenges. Scalability issues often result from reliance on central coordination without thorough testing of communication costs. Mitigation is mainly through alerts and external

actions, not tightly integrated, safety-aware loops. Though conceptually solid, hybrid systems lack sufficient real-world evidence for widespread deployment.

Across all paradigms, four main factors often explain why outcomes are only partial or nonexistent. First, evaluations mainly rely on offline datasets, which can improve reported accuracy but overlook essential factors such as latency, drift, and load behavior. Second, pipeline costs are frequently underestimated; preprocessing and feature extraction usually have a greater impact on runtime than the classifier itself. Third, architectural centralization, such as through gateways or cloud aggregation, can enhance accuracy but also increase delays and reduce resilience when scaled up. Fourth, responses tend to be under-specified; mitigation efforts are often missing or only loosely integrated, making them less effective against active attacks. These patterns explain why high benchmark accuracy usually doesn't translate into deployment readiness. They also guide key research areas discussed in Section 5, including scalability, real-time edge feasibility, integrated mitigation, and reproducible testbeds.

5. Discussion

A comprehensive review of the 32 surveyed IoT IDS studies was conducted using 10 operational criteria: real-time capability, latency, lightweight design, detection accuracy, mitigation, integrated detection-and-mitigation workflows, adaptability, resilience to sophisticated attacks, real-world validation, and scalability. The combined results highlight apparent differences in how these systems meet real-world IoT requirements.

The analysis reveals a gap between detection accuracy and real-world readiness. Many studies report classification accuracies above 95%, but practical deployment remains limited. Key operational factors like runtime efficiency, low latency, lightweight operation, autonomous response, scalability, and real-world testing are often overlooked. Few IDSs are suitable for resource-constrained devices or distributed networks. These findings highlight the need for lightweight, resilient, and adaptable IDSs that are well-suited to real-world IoT deployments. Despite high detection accuracy, many systems fail to deliver real-time, low-latency performance, especially at the device and edge levels, as they are resource-intensive on MCU- and SBC-based devices. Integrated detection-and-mitigation workflows and autonomous responses are rare, leaving systems reliant on passive detection. Real-world validation and scalability tests are infrequent, showing a gap between experiments and practical solutions. These patterns are summarized in Table 2 and Figure 2.

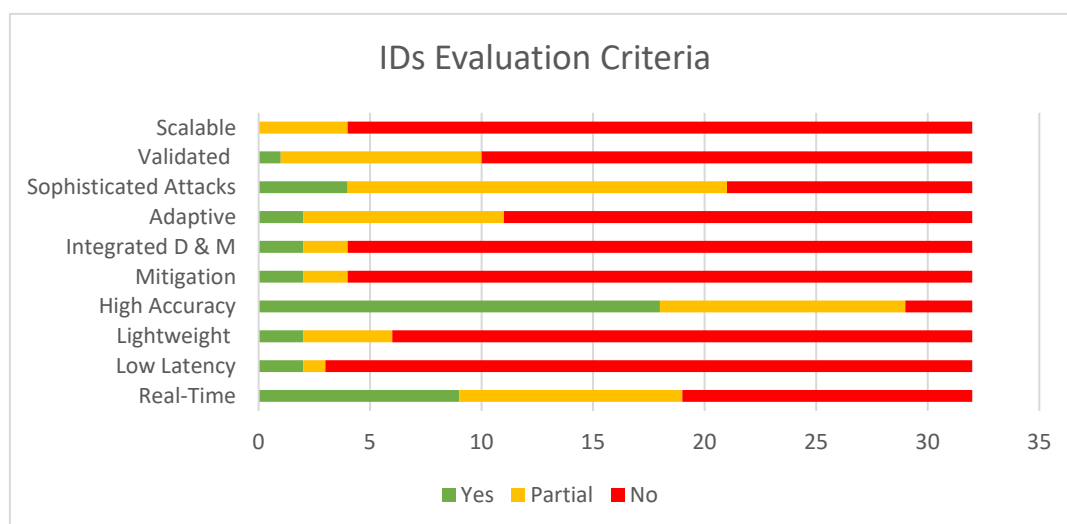


Figure 2. Evaluation of IoT IDSs against key performance and architectural criteria. The stacked bars indicate how many surveyed studies satisfy each criterion fully (Yes), partially (Partial), or not at all (No).

As shown in Table 2 and Figure 2, the surveyed IDSs demonstrate uneven alignment with the operational criteria, with notable weaknesses in latency, lightweight operation, mitigation, and scalability. These deficiencies appear consistently across all methodological families, suggesting field-wide gaps rather than limitations of individual approaches.

To enhance understanding of performance patterns, the systems were categorized into four paradigms: conventional, machine learning (ML), deep learning (DL), and hybrid approaches. Table 3 presents the counts of Yes, Partial, and No ratings for each paradigm across the ten criteria. Conventional IDSs demonstrate modest real-time capability but remain deficient in adaptability, mitigation, and scalability. ML-based models often achieve strong detection accuracy but struggle to satisfy latency and lightweight constraints. DL models provide high accuracy but typically fall short of real-time and lightweight requirements due to their computational demands. Hybrid approaches show the greatest architectural flexibility, balancing accuracy, partial adaptability, and modularity, but remain weak in real-world validation.

To evaluate architectural suitability, the IDSs were also categorized by deployment location: Device, Edge/Fog, Cloud, and Hybrid. As shown in Table 4, device-level IDSs perform poorly, with only 1 meeting real-time requirements; 4 have high accuracy but lack integration of detection and mitigation or scalability. Edge/Fog devices perform better: 6 provide real-time operation; 9 achieve high accuracy, but they still face issues such as a lack of lightweight design, limited mitigation support, and weak scalability. Cloud-based IDSs typically fall short on real-time and low-latency requirements and rarely address lightweight operation or mitigation. Hybrid deployments also produce mixed results, while some achieve high accuracy, lightweight operation, and scalability remain significant challenges.

Across different paradigms and deployment layers, the evaluation reveals a common problem: high offline accuracy does not necessarily imply that the system is well-suited to real-world IoT environments. Most systems lack real-time responsiveness, are not lightweight, do not provide autonomous mitigation, and have not been validated at a large scale, indicating that current IoT IDS research remains largely algorithm-centric, with limited emphasis on deployment readiness.

A focused quantitative analysis was conducted on the subset of studies reporting numerical latency data, allowing comparison of accuracy and latency efficiency across studies. Only 5 of 32 studies report numerical latency, so the observed tension between accuracy and latency should be seen as indicative of reporting and design trade-offs, not a definitive field-wide relationship. Comparability is limited because “latency” may refer to different measurement points, such as model inference alone or end-to-end detection delay that includes preprocessing, buffering, and communication, across diverse hardware and software setups. Table 5 shows the range and variability of delays and emphasizes the need for standardized runtime reporting rather than a universal latency benchmark across IoT IDS. All reported latencies were normalized to milliseconds for consistent comparison. To address the highly skewed distribution caused by extremely high latency values, both a 10% trimmed mean and the median were calculated. The trimmed-mean latency was 1257.5 ms, while the median latency was much lower at 530 ms, indicating the presence of outliers and showing that inference delays among IoT IDSs can vary widely. Despite this variation, detection accuracy remained high (trimmed mean = 97.76%, median = 97.9%), confirming that most IDS models prioritize classification accuracy over runtime performance. The accuracy-latency efficiency ratio was 0.18% per ms

(median-based) and 0.078% per ms (trimmed-mean-based). We introduce the efficiency ratio as a straightforward way to show the balance between accuracy and runtime, focusing on the small set of studies that report both metrics. Because latency can vary greatly, we calculate both a median-based ratio and a 10% trimmed-mean ratio to minimize the influence of extreme values. This combined measure isn't meant to be a universal benchmark across different devices or testing conditions; instead, it offers a clear, auditable summary that supports the main conclusion: that many systems achieve high accuracy without necessarily emphasizing runtime efficiency. These low ratios demonstrate that improvements in accuracy come at the expense of substantial latency increases, making the evaluated IDSs unsuitable for latency-critical IoT environments. Overall, these results emphasize a key point of this review: high accuracy does not guarantee operational practicality. Most IDSs do not meet the stringent timing requirements of resource-constrained IoT devices (≤ 1 – 10 ms at the node level, ≤ 20 – 50 ms at the gateway), underscoring the need for holistic optimization across accuracy, inference speed, model size, and resource usage.

Table 5. Reported Latency and Detection Performance for IoT IDS Models.

Ref	Latency (Reported)	Converted to ms	Accuracy
[73]	1 μ s	0.001 ms	99.74%
[83]	18,457 ms	18,457 ms	97.38%
[67]	3.95 s	3950 ms	97.9%
[85]	0.02–0.05 s	20–50 ms	98.0%
[86]	1.01 s	1010 ms	96.2%

5.1. Thematic Gap Analysis

The comparative assessment highlights several recurring thematic gaps that hinder the practical readiness of current IoT IDS solutions. These gaps remain consistent across different methodologies and deployment levels.

- (a) **Real-time and Latency:** Real-time intrusion detection remains one of the weakest dimensions across all studies. Only a minority meet IoT latency thresholds (≤ 3 of 32; see Table 2), and quantitative reporting remains uncommon. Even when latency is reported, values vary widely, reflecting poor runtime optimization.
- (b) **Lightweight constraints:** Lightweight operation is essential for constrained devices; however, few systems include resource profiling or hardware-aware design (≤ 6 of 32; see Table 2). Many proposals rely on computationally demanding models that cannot operate on MCU-class or battery-powered devices.
- (c) **Real-world validation:** Validation is predominantly dataset-based (>20 of 32; Table 2). This limits generalizability and can disguise performance degradation under real-world conditions.
- (d) **Mitigation:** Mitigation remains largely unaddressed, with the vast majority focusing solely on detection (>28 of 32; Table 2). Without response capabilities, IDSs cannot contain active threats or support rapid recovery after an intrusion.
- (e) **Adaptability:** Only a few systems demonstrate adaptive learning (≤ 11 of 32; Table 2). Most remain static after deployment, making them susceptible to concept drift and evolving attack behaviors.
- (f) **Scalability:** This criterion is the most deficient across all 10 categories because it has the fewest “Yes” ratings (0/32) and the highest combined number of “No” or unreported outcomes (28/32) (Table 2), indicating both limited achievement and systematic under-reporting. Most evaluations rely on small-scale or tightly controlled setups that do not reflect realistic multi-device or high-traffic IoT environments.

5.2. Structural Causes of Recurrent Gaps in IoT IDS Research

While the gaps identified in Section 5.1 are consistently observed across various studies, they are not just isolated issues. Instead, they stem from broader structural design and evaluation choices that shape current IoT IDS research.

First, **model-centric design priorities** often prioritize detection accuracy over practical usability. Many IDS proposals rely on deep or ensemble learning architectures optimized for classification on benchmark datasets, which can result in high computational demands, large models, and slower inference times. These factors can make real-time operation and lightweight deployment challenging, especially on devices like MCUs and SBCs.

Secondly, **centralized or cloud-based systems** remain common, even in studies primarily examining edge or IoT setups. Relying on centralized data collection, batch processing, and offline inference inherently introduces latency and limits scalability, particularly during periods of high traffic or multi-node operation. This preference for architecture explains why many systems claim real-time capabilities but often omit per-packet latency or continuous-throughput measurements.

Thirdly, **dataset-driven evaluation practices** heavily influence reported results. Using static, pre-assembled datasets, often tested offline with random train-test splits, can lead to overlooking real-world challenges such as timing issues, energy consumption, and communication delays. Because of this, systems might perform well in controlled environments but often don't reflect the streaming data, resource conflicts, or long-term changes that happen in real-world IoT setups.

Fourthly, **separating detection and response** in system design results in a lack of integrated mitigation. Many studies treat mitigation as an external component (such as firewall rules or manual actions), rather than as a co-designed part of the IDS architecture. This separation discourages closed-loop evaluation and limits the use of autonomous or policy-based response mechanisms.

Ultimately, **limited system-level validation**, often confined to single testbeds or simulations, makes it hard to assess real-world scalability. Without stress tests involving more nodes, higher traffic, or diverse devices, it's difficult to gauge how IDS solutions will perform in the field.

These underlying factors help explain why challenges related to real-time performance, scalability, adaptability, and mitigation persist across various methodological approaches. To effectively tackle these core issues, it's essential to move from simply optimizing algorithms toward a more holistic, deployment-aware system-level co-design.

5.3. Areas for Future Research

Building on the gaps identified in Section 5.1 and focusing on those with the most significant impact on real-world deployment, several specific research directions emerge. Scalability and real-time performance are the primary challenges, with lightweight operation, validation fidelity, and autonomous response following closely behind.

- (a) **Scalable Distributed IDS Architecture:** Scalability is the most deficient criterion across all evaluated dimensions, with 0/32 studies receiving a "Yes" rating and 28/32 either failing to demonstrate scalability or not evaluating it at all (Table 2). In most cases, scalability was not evaluated, but rather empirically disproven, due to reliance on small simulations, single gateways, or tightly controlled testbeds. None of the reviewed studies demonstrated stable operation under large-scale, multi-node, or high-throughput IoT conditions representative of real deployments. Future research should therefore prioritize distributed IDS architectures that explicitly assess synchronization overhead, communication costs, fault tolerance, and throughput stability under large-scale conditions, with evaluation protocols that move beyond single-testbed validation.

- (b) **Real-time Optimization:** Real-time responsiveness is only weakly supported in the literature. Only 1/32 studies satisfy node-level latency constraints (≤ 10 ms), and only 5/32 report any numerical latency measurements (Tables 2 and 5). In many cases, real-time capability is claimed but not demonstrated, with latency either unreported or measured only in offline or batch settings. Closing this gap requires architectural redesign rather than incremental classifier tuning, including optimization of feature extraction, buffering, inference, and decision pipelines. End-to-end latency must be evaluated under sustained streaming conditions to close the real-time gap identified in Section 5.1(a).
- (c) **Lightweight, Energy-Aware Design:** Resource efficiency is critical for MCU- and SBC-class devices, yet 27/32 studies report no hardware-level metrics such as RAM usage, model size, FLOPs, or energy per inference (Table 2). In most cases, lightweight suitability is asserted rather than empirically verified. Future IDS designs should treat lightweight operation as a primary design constraint, supported by explicit hardware-aware profiling and co-optimization of model architecture, feature pipelines, and deployment targets.
- (d) **Robust Validation and Reproducibility:** Validation practices remain heavily dataset-centric. Approximately 90% of the reviewed IDSs rely exclusively on offline datasets, with no physical testbed deployment, long-duration evaluation, or drift-aware testing (Table 2). This reflects a validation gap largely due to non-evaluation rather than documented failure in real-world settings. Stronger validation through physical testbeds, long-running experiments, and concept-drift-sensitive evaluation is therefore required to assess stability, reliability, and lifecycle robustness and to close the observed validation gap affecting the majority of studies.
- (e) **Integrated Detection-Mitigation Pipelines:** Only 2/32 studies implement any form of automated mitigation, and even fewer integrate mitigation into a closed-loop detection–response workflow (Table 2). In most cases, mitigation is not implemented, but rather unsuccessfully demonstrated, and is limited to alerts, external firewalls, or future-work statements. Future research should focus on policy-bounded, closed-loop mitigation pipelines with rollback mechanisms, particularly for mission-critical IoT environments where passive detection alone is insufficient.
- (f) **Adaptive and Continual Learning:** Adaptability remains limited across the surveyed literature. 30/32 systems remain static after training, with no support for online learning, concept drift detection, or continual adaptation (Table 2). While several studies conceptually reference adaptability, few demonstrate adaptive behavior empirically. Long-lived IoT IDS deployments therefore require integration of streaming learners, drift detection, adversarial robustness, and safe model update mechanisms to maintain effectiveness under evolving threat conditions.
- (g) **Hardware-Software Co-Design:** Comparison across IDS proposals is hindered by inconsistent evaluation practices across MCU-class, SBC-class, and cloud environments, with thresholds and metrics often undefined or incomparable. This limitation reflects systematic under-reporting rather than negative results. Future IDS research should adopt hardware–software co-design principles and standardized, tier-aware evaluation practices to ensure that reported performance is meaningful and transferable across deployment contexts.

Collectively, these research directions are directly derived from the quantified operational gaps in Section 5.1 and provide a deployment-oriented roadmap for advancing IoT IDS research beyond algorithmic accuracy toward real-time, scalable, validated, and deployable security solutions.

5.4. Broader Impact and Significance

The Advancement of IoT IDS research requires a transition from algorithm-centric evaluation toward deployment-aware, safety-aligned, and reproducible system architectures. This shift is necessary because high accuracy alone, reported by nearly all studies, did not consistently translate into acceptable latency, resource use, or adaptability in our evaluation.

- (a) **Hardware-Tiered Benchmark Suite for IoT IDS Evaluation:** Our review reveals broad inconsistencies in latency reporting, with only 5 studies quantifying inference time and even fewer quantifying hardware usage. A unified benchmark suite is essential for enabling consistent, comparable evaluation across diverse IoT platforms. Such a suite should define: (i) *MCU-class benchmarks* (e.g., ≤ 10 ms inference, ≤ 100 KB RAM, ≤ 1 mJ energy), (ii) *SBC/edge benchmarks* (e.g., ≤ 50 ms inference, ≤ 500 MB working memory, moderate energy profiles), (iii) *Cloud benchmarks* emphasizing throughput, long-horizon analytics, and multi-tenant scalability. Each benchmark tier should include standardized workloads, datasets, device profiles, and latency/energy reporting requirements to enable transparent, repeatable, and comparable evaluation across studies. These thresholds serve as reference envelopes rather than fixed standards. They should be adjusted based on the application context (e.g., control-loop criticality, device duty cycle, network topology) using representative hardware benchmarks and deployment constraints.
- (b) **Composite Scoring Rubric for Deployment Readiness:** A deployment-readiness score would allow objective comparison of competing IDS approaches. Such a rubric is needed because accuracy metrics alone can obscure substantial weakness in real-time capability, mitigation, and scalability. Detection accuracy and minority-class performance, Real-time latency, Lightweight resource usage (CPU, RAM, model size, FLOPs, energy), Adaptability to drift and evolving threats, Autonomous mitigation capability, Real-world validation fidelity, and Scalability under load. Such a rubric would allow researchers to compare system-level performance rather than relying solely on accuracy. A single deployment-readiness score derived from these components would enable objective comparison between IDS designs. Weighting can depend on context, such as prioritizing latency and energy in MCU-class deployments or focusing on scalability and mitigation in infrastructure-scale settings, using transparent, normalized scoring rather than opaque composite optimization.
- (c) **Governance and Safety Principles for Autonomous Mitigation:** Autonomous mitigation must remain bounded by safety, policy compliance, and human oversight. As mitigation actions become automated, incorrect or overly aggressive responses may introduce operational risk, especially since most IDSs today lack robust handling of false positives. These safeguards are critical in safety-critical domains such as healthcare, industrial IoT, and smart mobility systems. Given the operational risks associated with autonomous mitigation, especially in the presence of false positives, transparent human-in-the-loop governance remains essential.
- (d) **Public Reproducibility Artifacts for Transparent Evaluation:** None of the reviewed studies provided complete replication packages, and only a minority released pre-processing code. Reproducibility is essential for scientific trust and practical adoption. To combat fragmentation and improve reproducibility, future IDS research should release: Open-source evaluation scripts, Standardized preprocessing pipelines, Lightweight deployable models (e.g., TFLite, ONNX, Edge-optimized binaries), Testbed configuration files, and network emulation scenarios. Such artifacts would support cross-laboratory validation and accelerate the maturation of IoT IDS research. A shared repository would also support continual updates as new datasets, device types, and attack patterns emerge.

- (e) **Ethical, Privacy, and Adversarial Considerations:** Complementing the governance safeguards outlined in (c), large-scale IDS deployment introduces additional ethical and adversarial risks beyond just mitigation control. These include: *Privacy concerns* in monitoring device behavior and network flows; *Exposure to adversarial* examples or poisoning attacks that exploit model vulnerabilities; *Fairness and bias* across heterogeneous device classes; and Potential harm from *false-positive-driven* automated actions. Addressing these concerns is essential to ensuring trustworthy and responsible adoption of IDS, particularly in safety-critical environments such as healthcare, industrial IoT, and smart cities.

Finally, this evaluation shows that despite advances in algorithms, IoT intrusion detection still faces significant operational challenges. High accuracy has not translated into solutions that meet real-world needs, with few providing real-time responses, lightweight operation, or edge performance. The overwhelming reliance on offline datasets further widens the gap between academic benchmarks and deployable security. The lack of lifecycle evaluation, including drift handling, long-term robustness testing, and multi-day operation, further prevents assessing whether models remain reliable after deployment. This gap aligns closely with earlier thematic findings and is a critical barrier to trustworthy, field-ready IDS solutions. Scalability, adaptability, and mitigation are often absent. The findings suggest that future IoT IDS research should shift from developing algorithms to creating deployment-aware architectures, emphasizing lightweight, adaptive, explainable, and real-time solutions integrated with automated mitigation and supported by standardized evaluation. Addressing these challenges is essential to building robust IoT defenses.

6. Conclusions

This study offers a comprehensive review of recent IoT intrusion detection systems (IDSs), spanning traditional, machine-learning, deep-learning, and hybrid approaches. The findings show that although detection accuracy continues to advance, critical operational dimensions, real-time performance, lightweight design, adaptability, mitigation capability, and scalability, remain significantly underdeveloped. Out of the 32 studies reviewed, over 75% are detection-only prototypes tested on static datasets. Only 5 studies report latency, fewer than 25% include any mitigation, and in more than 85%, scalability is either unmet or not reported. These findings underscore the need for next-generation IDS designs that combine lightweight architecture, real-time analytics, and AI-driven autonomous capabilities. Future systems should incorporate energy- and latency-efficient benchmarks, standardized validation environments, and explainable, agent-based intelligence to support resilient and transparent operation in large-scale IoT ecosystems. Overall, this review outlines the foundational requirements for developing self-healing, intelligent, and verifiably autonomous IoT intrusion detection and mitigation systems that can adapt to continuously evolving threats.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/iot7010016/s1>, Table S1: Per-study risk-of-bias assessment across six dimensions.

Author Contributions: S.S. conceptualized, designed, and conducted the study, including data analysis, visualization, and manuscript drafting. The research was supervised and administratively supported by M.E.B. and N.L., with funding obtained by M.E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any external funding. The University of Wollongong in Dubai paid the Article Processing Charge (APC).

Data Availability Statement: No new data were generated or analyzed in this study. Data sharing does not apply to this article because it relies entirely on previously published research and publicly available datasets referenced within the manuscript.

Acknowledgments: This paper is part of the first author's Ph.D. thesis at the School of Computing and Information Technology, University of Wollongong. The work was financially supported by the University of Wollongong in Dubai (UOWD). The author sincerely thanks May El Barachi and Nan Li for their academic guidance and supervision throughout this research. During the preparation of this manuscript, generative artificial intelligence tools were only used for language refinement and structural editing. The authors reviewed and verified all content and take full responsibility for the publication's integrity.

Conflicts of Interest: The authors declare no conflicts of interest. The funder had no role in designing the study, collecting, analyzing, or interpreting data, writing the manuscript, or deciding to publish the results.

Abbreviations

AE	Autoencoder	IRS	Intrusion Response System
AI	Artificial Intelligence	KNN	K-Nearest Neighbors
APT	Advanced Persistent Threat	LSTM	Long Short-Term Memory Network
BA	Bat Algorithm	LR	Logistic Regression
CBAM	Convolutional Block Attention Module	MACs	Multiply-Accumulate Operations
CEP	Complex Event Processing	MCU	Microcontroller Unit
CNN	Convolutional Neural Network	ML	Machine Learning
DDoS	Distributed Denial of Service	MLP	Multilayer Perceptron
DL	Deep Learning	MQTT	Message Queuing Telemetry Transport
DNN	Deep Neural Network	NFV	Network Function Virtualization
DoS	Denial of Service	NIDS	Network Intrusion Detection System
DT	Decision Tree	OS-ELM	Online Sequential Extreme Learning Machine
SBC	Single-Board Computer (Edge-Level Device)	R2L	Remote-to-Local Attack Category
EPL	Event Processing Language	RF	Random Forest
FedAvg	Federated Averaging Algorithm	RNN	Recurrent Neural Network
FL	Federated Learning	RPL	Routing Protocol for Low-Power and Lossy Networks
FLOPs	Floating-Point Operations	SD-IoT	Software-Defined Internet of Things
FPR	False Positive Rate	SDN	Software-Defined Networking
FWO	Fireworks Optimization	SOHO	Small Office/Home Office Network
GAN	Generative Adversarial Network	SVM	Support Vector Machine
HIDS	Host-Based Intrusion Detection System	TFLite	TensorFlow Lite
IDS	Intrusion Detection System	TPR	True Positive Rate
IDPS	Intrusion Detection and Prevention System	U2R	User-to-Root Attack Category
IoMT	Internet of Medical Things	UN	Unify Net (DL model component)
IoT	Internet of Things	VM	Virtual Machine
IOLTS	Input-Output Labeled Transition System	WSN	Wireless Sensor Network

References

1. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>.
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>.
3. Rejeb, A.; Rejeb, K.; Appolloni, A.; Jagtap, S.; Iranmanesh, M.; Alghamdi, S.; Alhasawi, Y.; Kayikci, Y. Unleashing the power of internet of things and blockchain: A comprehensive analysis and future directions. *Internet Things Cyber-Physical Syst.* **2024**, *4*, 1–18. <https://doi.org/10.1016/j.iotcps.2023.06.003>.
4. Alloui, H.; Mourdi, Y. Exploring the Full Potentials of IoT for Better Financial Growth and Stability: A Comprehensive Survey. *Sensors* **2023**, *23*, 8015. <https://doi.org/10.3390/s23198015>.

5. H.J., F.B.; S., S. A Survey on IoT Security: Attacks, Challenges and Countermeasures. *Webology* **2022**, *19*, 3741–3763. <https://doi.org/10.14704/web/v19i1/web19246>.
6. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. <https://doi.org/10.1109/ACCESS.2019.2924045>.
7. Baldini, G.; Botterman, M.; Neisse, R.; Tallacchini, M. Ethical Design in the Internet of Things. *Sci. Eng. Ethics* **2018**, *24*, 905–925. <https://doi.org/10.1007/s11948-016-9754-5>.
8. Callebaut, G.; Leenders, G.; Van Mulders, J.; Ottoy, G.; De Strycker, L.; Van der Perre, L. The art of designing remote iot devices—Technologies and strategies for a long battery life. *Sensors* **2021**, *21*, 913. <https://doi.org/10.3390/s21030913>.
9. Dritsas, E.; Trigka, M. A Survey on Cybersecurity in IoT. *Future Internet* **2025**, *17*, 30. <https://doi.org/10.3390/fi17010030>.
10. Sam, M.F.M.; Ismail, A.F.M.F.; Bakar, K.A.; Ahamat, A.; Qureshi, M.I. The Effectiveness of IoT Based Wearable Devices and Potential Cybersecurity Risks: A Systematic Literature Review from the Last Decade. *Int. J. Online Biomed. Eng.* **2022**, *18*, 56–73. <https://doi.org/10.3991/ijoe.v18i09.32255>.
11. Madanian, S.; Chinbat, T.; Subasinghage, M.; Airehrour, D.; Hassandoust, F.; Yongchareon, S. Health IoT Threats: Survey of Risks and Vulnerabilities. *Future Internet* **2024**, *16*, 389. <https://doi.org/10.3390/fi16110389>.
12. Oliha, J.S.; Biu, P.W.; Obi, O.C. Securing the smart city: A review of cybersecurity challenges and strategies. *Open Access Res. J. Multidiscip. Stud.* **2024**, *7*, 094–101. <https://doi.org/10.53022/oarjms.2024.7.1.0013>.
13. Gharaibeh, A.; Salahuddin, M.A.; Hussini, S.J.; Khreishah, A.; Khalil, I.; Guizani, M.; Al-Fuqaha, A. Smart Cities: A Survey on Data Management, Security, and Enabling Technologies. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 2456–2501. <https://doi.org/10.1109/COMST.2017.2736886>.
14. Alotaibi, B. A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities. *Sensors* **2023**, *23*, 7470. <https://doi.org/10.3390/s23177470>.
15. Vardakis, G.; Hatzivasilis, G.; Koutsaki, E.; Papadakis, N. Review of Smart-Home Security Using the Internet of Things. *Electronics* **2024**, *13*, 3343. <https://doi.org/10.3390/electronics13163343>.
16. Khan, R.; Maynard, P.; McLaughlin, K.; Lavery, D.; Sezer, S. *Threat Analysis of BlackEnergy Malware for Synchrophasor Based Real-time Control and Monitoring in Smart Grid*; BCS Learning & Development Ltd.: Swindon, UK, 2016.
17. Martin, G.; Martin, P.; Hankin, C.; Darzi, A.; Kinross, J. Cybersecurity and healthcare: How safe are we? *BMJ (Online)* **2017**, *358*, j3179. <https://doi.org/10.1136/bmj.j3179>.
18. Anagnostopoulos, M.; Spathoulas, G.; Viaño, B.; Augusto-Gonzalez, J. Tracing your smart-home devices conversations: A real world iot traffic data-set. *Sensors* **2020**, *20*, 6600. <https://doi.org/10.3390/s20226600>.
19. Maghrabi, L.A.; Shabanah, S.; Althaqafi, T.; Alsalman, D.; Algarni, S.; Al-Ghamdi, A.A.-M.; Ragab, M. Enhancing Cybersecurity in the Internet of Things Environment Using Bald Eagle Search Optimization With Hybrid Deep Learning. *IEEE Access* **2024**, *12*, 8337–8345. <https://doi.org/10.1109/ACCESS.2024.3352568>.
20. Heidari, A.; Jamali, M.A.J. Internet of Things intrusion detection systems: A comprehensive review and future directions. *Clust. Comput* **2023**, *26*, 3753–3780. <https://doi.org/10.1007/s10586-022-03776-z>.
21. Mishra, N.; Pandya, S. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access* **2021**, *9*, 59353–59377. <https://doi.org/10.1109/ACCESS.2021.3073408>.
22. Arshad, J.; Azad, M.A.; Abdeltaif, M.M.; Salah, K. An intrusion detection framework for energy constrained IoT devices. *Mech. Syst. Signal Process.* **2020**, *136*, 106436. <https://doi.org/10.1016/j.ymssp.2019.106436>.
23. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. <https://doi.org/10.1109/COMST.2020.2988293>.
24. Kaur, B.; Dadkhah, S.; Shoeleh, F.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Lamontagne, P.; Ray, S.; Ghorbani, A.A. Internet of Things (IoT) security dataset evolution: Challenges and future directions. *Internet Things* **2023**, *22*, 100780. <https://doi.org/10.1016/j.iot.2023.100780>.
25. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT privacy and security: Challenges and solutions. *Appl. Sci.* **2020**, *10*, 4102. <https://doi.org/10.3390/AP10124102>.
26. Benkhelifa, E.; Welsh, T.; Hamouda, W. A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE Commun. Surv. Tutorials* **2018**, *20*, 3496–3509. <https://doi.org/10.1109/COMST.2018.2844742>.
27. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. <https://doi.org/10.1186/s42400-021-00077-7>.

28. Esmaeili, M.; Rahimi, M.; Pishdast, H.; Farahmandazad, D.; Khajavi, M.; Saray, H.J. Machine Learning-Assisted Intrusion Detection for Enhancing Internet of Things Security. *arXiv* **2024**, <https://doi.org/10.48550/arXiv.2410.01016>.
29. Vitorino, J.; Andrade, R.; Praça, I.; Sousa, O.; Maia, E. *A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection*; Springer: Cham, Switzerland, 2022. https://doi.org/10.1007/978-3-031-08147-7_13.
30. Paul, E.C.; Amrita. A Review of Intrusion Detection for Internet of Things Using Machine Learning. In Proceedings of the 2024 International Conference on Cybernation and Computation, CYBERCOM 2024, Dehradun, India, 15–16 November 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, pp. 781–785. <https://doi.org/10.1109/CYBERCOM63683.2024.10803135>.
31. Al-Haija, Q.A.; Droos, A. A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT). *Expert Syst.* **2025**, *42*, e13726. <https://doi.org/10.1111/exsy.13726>.
32. Almowsawi, A.P.A.A.H.D. Deep Guard-IoT: A Systematic Review of AI-Based Anomaly Detection Frameworks for Next-Generation IoT Security (2020–2024). *Wasit J. Pure Sci.* **2024**, *3*, 70–77. <https://doi.org/10.31185/wjps.598>.
33. Rajesh, L.T.; Das, T.; Shukla, R.M.; Sengupta, S. Give and Take: Federated Transfer Learning for Industrial IoT Network Intrusion Detection. *ArXiv*, **2023**, <https://doi.org/10.48550/arXiv.2310.07354>.
34. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. <https://doi.org/10.1016/j.jnca.2017.02.009>.
35. Elrawy, M.F.; Awad, A.I.; Hamed, H.F.A. Intrusion detection systems for IoT-based smart environments: A survey. *J. Cloud Comput.* **2018**, *7*, 21. <https://doi.org/10.1186/s13677-018-0123-6>.
36. Hajiheidari, S.; Wakil, K.; Badri, M.; Navimipour, N.J. Intrusion detection systems in the Internet of things: A comprehensive investigation. *Comput. Networks* **2019**, *160*, 165–191. <https://doi.org/10.1016/j.comnet.2019.05.014>.
37. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. <https://doi.org/10.1109/COMST.2019.2896380>.
38. da Costa, K.A.P.; Papa, J.P.; Lisboa, C.O.; Munoz, R.; de Albuquerque, V.H.C. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Comput. Netw.* **2019**, *151*, 147–157. <https://doi.org/10.1016/j.comnet.2019.01.023>.
39. Naithani, K. AI-based Intrusion Detection System for Internet of Things (IoT) Networks. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2019**, *10*, 1095–1100. <https://doi.org/10.17762/turcomat.v10i2.13631>.
40. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. <https://doi.org/10.1109/COMST.2020.2986444>.
41. Kamaldeep; Dutta, M.; Granjal, J. Towards a Secure Internet of Things: A Comprehensive Study of Second Line Defense Mechanisms. *IEEE Access* **2020**, *8*, 127272–127312. <https://doi.org/10.1109/ACCESS.2020.3005643>.
42. Mazhar, N.; Salleh, R.; Hossain, M.A.; Zeeshan, M. SDN based Intrusion Detection and Prevention Systems using Manufacturer Usage Description: A Survey. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 717–737. <https://doi.org/10.14569/IJACSA.2020.0111283>.
43. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. Iot intrusion detection taxonomy, reference architecture, and analyses. *Sensors* **2021**, *21*, 6432. <https://doi.org/10.3390/s21196432>.
44. Alsoufi, M.A.; Razak, S.; Siraj, M.M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. *Appl. Sci.* **2021**, *11*, 8383. <https://doi.org/10.3390/app11188383>.
45. Inayat, U.; Zia, M.F.; Mahmood, S.; Khalid, H.M.; Benbouzid, M. Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects. *Electronics* **2022**, *11*, 1502. <https://doi.org/10.3390/electronics11091502>.
46. Kumar, S.V.N.S.; Selvi, M.; Kannan, A. A Comprehensive Survey on Machine Learning-Based Intrusion Detection Systems for Secure Communication in Internet of Things. *Comput. Intell. Neurosci.* **2023**, *2023*, 8981988. <https://doi.org/10.1155/2023/8981988>.
47. Aldhaheri, A.; Alwahedi, F.; Ferrag, M.A.; Battah, A. Deep learning for cyber threat detection in IoT networks: A review. *Internet Things Cyber-Physical Syst.* **2024**, *4*, 110–128. <https://doi.org/10.1016/j.iotcps.2023.09.003>.
48. Meziane, H.; Ouerdi, N. A survey on performance evaluation of artificial intelligence algorithms for improving IoT security systems. *Sci. Rep.* **2023**, *13*, 21255. <https://doi.org/10.1038/s41598-023-46640-9>.
49. Kumari, P.; Mangat, V.; Singh, A. Comparative Analysis of State-of-the-Art Attack Detection Models. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023, Delhi, India, 6–8 July 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA. <https://doi.org/10.1109/ICCCNT56998.2023.10306428>.
50. Rafique, S.H.; Abdallah, A.; Musa, N.S.; Murugan, T. Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection—Current Research Trends. *Sensors* **2024**, *24*, 1968. <https://doi.org/10.3390/s24061968>.

51. Isma'ila, U.A.; Danyaro, K.U.; Muazu, A.A.; Maiwada, U.D. Corrections to "Review on Approaches of Federated Modeling in Anomaly-Based Intrusion Detection for IoT Devices". *IEEE Access* **2024**, *12*, 30941–30961. <https://doi.org/10.1109/ACCESS.2024.3486411>.
52. Hassan, H.A.A.; Zolfy, M. Exploring Lightweight Deep Learning Techniques for Intrusion Detection Systems in IoT Networks: A Survey. **2024**, *20*, 1944–1958. <https://doi.org/10.52783/jes.2292>
53. Blali, A.; Dargaoui, S.; Azrou, M.; Guezzaz, A.; Amounas, F.; Alabdulatif, A. Analysis of deep learning-based intrusion detection systems in IoT environments. *EDPACS* **2025**, *70*, 18–52. <https://doi.org/10.1080/07366981.2025.2498222>.
54. Fatima, M.; Rehman, O.; Rahman, I.M.H.; Ajmal, A.; Park, S.J. Towards Ensemble Feature Selection for Lightweight Intrusion Detection in Resource-Constrained IoT Devices. *Future Internet* **2024**, *16*, 368. <https://doi.org/10.3390/fi16100368>.
55. Sarhan, M.; Layeghy, S.; Portmann, M. Feature Analysis for Machine Learning-based IoT Intrusion Detection. *arXiv* **2022**, <https://doi.org/10.48550/arXiv.2108.12732>.
56. Ge, M.; Syed, N.F.; Fu, X.; Baig, Z.; Robles-Kelly, A. Toward a Deep Learning-Driven Intrusion Detection Approach for Internet of Things. *arXiv* **2020**, <https://doi.org/10.48550/arXiv.2007.09342>.
57. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. <https://doi.org/10.1016/j.adhoc.2013.04.014>.
58. Yin, S.-N.; Kang, H.-S.; Kim, S.-R. Complex Event Processing for Object Tracking and Intrusion Detection in Internet of Things Environments. *Res. Briefs Inf. Commun. Technol. Evol.* **2016**, *2*, 74–81.
59. Fu, Y.; Yan, Z.; Cao, J.; Koné, O.; Cao, X. An Automata Based Intrusion Detection Method for Internet of Things. *Mob. Inf. Syst.* **2017**, *2017*, 1750637. <https://doi.org/10.1155/2017/1750637>.
60. Haripriya, A.P.; Kulothungan, K. Secure-MQTT: An efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 90. <https://doi.org/10.1186/s13638-019-1402-8>.
61. Prabavathy, S.; Sundarakantham, K.; Shalinie, S.M. Design of cognitive fog computing for intrusion detection in Internet of Things. *J. Commun. Netw.* **2018**, *20*, 291–298. <https://doi.org/10.1109/JCN.2018.000041>.
62. Li, J.; Zhao, Z.; Li, R.; Zhang, H. AI-based two-stage intrusion detection for software defined IoT networks. *IEEE Internet Things J* **2019**, *6*, 2093–2102. <https://doi.org/10.1109/JIOT.2018.2883344>.
63. Zachos, G.; Essop, I.; Mantas, G.; Porfyakis, K.; Ribeiro, J.C.; Rodriguez, J. An anomaly-based intrusion detection system for internet of medical things networks. *Electronics* **2021**, *10*, 2562. <https://doi.org/10.3390/electronics10212562>.
64. Zachos, G.; Mantas, G.; Essop, I.; Porfyakis, K.; Ribeiro, J.C.; Rodriguez, J. Prototyping an Anomaly-Based Intrusion Detection System for Internet of Medical Things Networks. In Proceedings of the IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, Paris, France, 2–4 November 2022, Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, pp. 179–183. <https://doi.org/10.1109/CAMAD55695.2022.9966912>.
65. Vishwakarma, M.; Kesswani, N. A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection. *Decis. Anal. J.* **2023**, *7*, 100233. <https://doi.org/10.1016/j.dajour.2023.100233>.
66. Alosaimi, S.; Almutairi, S.M. An Intrusion Detection System Using BoT-IoT. *Appl. Sci.* **2023**, *13*, 5427. <https://doi.org/10.3390/app13095427>.
67. Fadhilla, C.A.; Alfikri, M.D.; Kaliski, R. Lightweight Meta-Learning BotNet Attack Detection. *IEEE Internet Things J.* **2023**, *10*, 8455–8466. <https://doi.org/10.1109/JIOT.2022.3229463>.
68. Tahir, U.; Abid, M.K.; Fuzail, M.; Aslam, N. Enhancing IoT Security through Machine Learning-Driven Anomaly Detection. *VFAST Trans. Softw. Eng.* **2024**, *12*, 1–13. <https://doi.org/10.21015/otse.v12i1.1766>.
69. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. <https://doi.org/10.1016/j.future.2017.08.043>.
70. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. <https://doi.org/10.1016/j.compeleceng.2022.107810>.
71. Vishwakarma, M.; Kesswani, N. DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. *Decis. Anal. J.* **2022**, *5*, 100142. <https://doi.org/10.1016/j.dajour.2022.100142>.
72. Khan, A.R.; Yasin, A.; Usman, S.M.; Hussain, S.; Khalid, S.; Ullah, S.S. Exploring Lightweight Deep Learning Solution for Malware Detection in IoT Constraint Environment. *Electronics* **2022**, *11*, 4147. <https://doi.org/10.3390/electronics11244147>.
73. Idrissi, I.; Azizi, M.; Moussaoui, O. A Lightweight Optimized Deep Learning-based Host-Intrusion Detection System Deployed on the Edge for IoT. *Int. J. Comput. Digit. Syst.* **2022**, *11*, 209–216. <https://doi.org/10.12785/ijcds/110117>.
74. Fang, Z.; Liu, Y.; Yuan, S.; Ye, T. A lightweight network intrusion detection model based on convolution and attention mechanisms. In Proceedings of the Third International Conference on Electronic Information Engineering, Big Data, and Computer

- Technology (EIBDCT 2024), Beijing, China, 26–28 January 2024; SPIE-The International Society for Optical Engineering: Bellingham, WA, USA, 2024; p. 32. <https://doi.org/10.1117/12.3031017>.
75. Binbusayyis, A. Innovative Defense: Deep Learning-Powered Intrusion Detection for IoT Networks. *IEEE Access* **2025**, *13*, 31105–31120. <https://doi.org/10.1109/ACCESS.2025.3542275>.
 76. Sedjelmaci, H.; Senouci, S.M.; Taleb, T. An accurate security game for low-resource iot devices. *IEEE Trans. Veh. Technol.* **2017**, *66*, 9381–9393. <https://doi.org/10.1109/TVT.2017.2701551>.
 77. Sedjelmaci, H.; Senouci, S.M.; Al-Bahri, M. A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; IEEE: New York, NY, USA; pp. 1–6. <https://doi.org/10.1109/ICC.2016.7510811>.
 78. Mudgerikar, A.; Sharma, P.; Bertino, E. Edge-Based Intrusion Detection for IoT devices. *ACM Trans. Manag. Inf. Syst.* **2020**, *11*, 1–21. <https://doi.org/10.1145/3382159>.
 79. Holubenkov, V.; Silva, P. An Intelligent Mechanism for Monitoring and Detecting Intrusions in IoT Devices. In Proceedings of the 2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2023, Boston, MA, USA, 12–15 June 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 470–479. <https://doi.org/10.1109/WoWMoM57956.2023.00082>.
 80. Talpini, J.; Sartori, F.; Savi, M. A Clustering Strategy for Enhanced FL-Based Intrusion Detection in IoT Networks. In Proceedings of the International Conference on Agents and Artificial Intelligence, Lisbon, Portugal, 22–24 February 2023; Science and Technology Publications, Lda: Setúbal, Portugal, 2023; pp. 152–160. <https://doi.org/10.5220/0011627500003393>.
 81. Grigoriadou, S.; Radoglou-Grammatikis, P.; Sarigiannidis, P.; Makris, I.; Lagkas, T.; Argyriou, V.; Lytos, A.; Fountoukidis, E. Hunting IoT Cyberattacks with AI—Powered Intrusion Detection. In Proceedings of the 2023 IEEE International Conference on Cyber Security and Resilience, CSR 2023, Venice, Italy, 31 July–2 August 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 142–147. <https://doi.org/10.1109/CSR57506.2023.10224981>.
 82. Fenanir, S.; Semchedine, F. Smart Intrusion Detection in IoT Edge Computing Using Federated Learning. *Rev. D'intelligence Artif.* **2023**, *37*, 1133–1145. <https://doi.org/10.18280/ria.370505>.
 83. Majjaru, C.; Senthilkumar, K. Strengthening IoT Intrusion Detection through the HOPNET Model. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2023**, *14*, 89–102. <https://doi.org/10.58346/JOWUA.2023.I3.007>.
 84. Abusitta, A.; de Carvalho, G.H.S.; Wahab, O.A.; Halabi, T.; Fung, B.C.M.; Al Mamoori, S. Deep learning-enabled anomaly detection for IoT systems. *Internet Things* **2023**, *21*, 100656. <https://doi.org/10.1016/j.iot.2022.100656>.
 85. Alalhareth, M.; Hong, S.C. Enhancing the Internet of Medical Things (IoMT) Security with Meta-Learning: A Performance-Driven Approach for Ensemble Intrusion Detection Systems. *Sensors* **2024**, *24*, 3519. <https://doi.org/10.3390/s24113519>.
 86. Deng, Y. Design of Industrial IoT Intrusion Security Detection System Based on LightGBM Feature Algorithm and Multi-layer Perception Network. *J. Cyber Secur. Mobil.* **2024**, *13*, 327–348. <https://doi.org/10.13052/jcsm2245-1439.1327>.
 87. Thiruvengkatasamy, S.; Sivaraj, R.; Vijayakumar, M. Blockchain Assisted Fireworks Optimization with Machine Learning based Intrusion Detection System (IDS). *Teh. Vjesn.* **2024**, *31*, 596–603. <https://doi.org/10.17559/TV-20230712000798>.
 88. Adekunle, T.S.; Alabi, O.O.; Lawrence, M.O.; Adeleke, T.A.; Afolabi, O.S.; Ebong, G.N.; Egbedokun, G.O.; Bamisaye, T.A. An Intrusion System for Internet of Things Security Breaches Using Machine Learning Techniques. *Artif. Intell. Appl.* **2024**, *2*, 165–171. <https://doi.org/10.47852/bonviewAIA42021780>.

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