

Article

Evaluation Framework for Cross-Layer Intrusion Detection Based on ML Approaches

Dimitrios Tasiopoulos ^{1,*}, Apostolos Xenakis ^{1,*}, Alexios Lekidis ², Dimitrios Kosmanos ¹,
Costas Chaikalas ¹ and Vasileios Vlachos ³

¹ Department of Digital Systems, University of Thessaly, 41500 Larissa, Greece; dikosman@uth.gr (D.K.); kchaikalas@uth.gr (C.C.)

² Department of Energy Systems, University of Thessaly, 41500 Larissa, Greece; alekidis@uth.gr

³ Department of Economics, University of Thessaly, 38333 Volos, Greece; vsvlachos@uth.gr

* Correspondence: dimtasiopoulos@uth.gr (D.T.); axenakis@uth.gr (A.X.)

† These authors contributed equally to this work.

Abstract

Internet of Things networks evolve as a rapidly growing field for security threats, such as Denial-of-Service cross-layer attacks, due to their heterogeneous and resource-constrained environment. Intrusion detection systems (IDSs) serve as a vital defense mechanism in modern cybersecurity. However, the adoption of such a system, especially one that adopts a cross-layer strategy, requires a standardized, multifaceted evaluation framework that accounts for both detection capability and operational overhead. To address these challenges, we proposed a modular weight-based framework that evaluates cross-layer Machine Learning (ML) IDS across multiple dimensions, namely, detection effectiveness and generalizability, data quality, and attack coverage and practical deployability. We then applied this framework to the state-of-the-art cross-layer ML IDSs identified through the PRISMA framework. This proof-of-concept application illustrates how current evaluation practices generate disparate, fragmented results, while also highlighting the limitations inherent in retrospective literature-based scoring.

Keywords: cross-layer; Denial-of-Service; machine learning; Internet-of-Things; evaluation framework; intrusion detection systems

1. Introduction

Denial-of-Service (DoS) attacks aim to compromise node or network availability by depleting critical network and energy resources. Within this domain, cross-layer attacks represent a severe stretch. While conventional attacks are restricted to an isolated layer of the protocol stack, cross-layer strategies traverse multiple levels. This allows a seemingly minor disruption at one layer to propagate which results in magnified, cascading effects across the other layers.

One of the most effective technologies against such threats is an Intrusion Detection System (IDS). An IDS is a hardware- or software-based security solution that is used to monitor and identify unauthorized or malicious activities within a computer network or system. Its main use is detective, as it observes events and raises alarms without taking action towards addressing the suspicious event [1].

In case of cross-layer attacks, relying on a single layer IDS imposes various limitations. Conventional single-layer IDSs tend to detect only external attacks on isolated layers and overlook risks at system level and cross-layer disturbances [2]. In resource-constrained



Academic Editor: Hung-Yu Chien

Received: 1 June 2026

Revised: 2 July 2026

Accepted: 3 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

IoT environments, detection methods depending on specific data sources and single layers prove inadequate [3]. Traditionally, single-layer approaches, for example application-layer antivirus, network-layer firewalls, and middleware-layer hardware analysis, are not capable of identifying multilateral attack characteristics [4]. Such narrow observation often results in missed indicators or false positives, particularly in networks under energy and resource limitations [4]. Moreover, single-layer strategies cannot counter cross-layer attacks efficiently, where an action on one layer has effect on another [5].

State-of-the-art IDSs are starting to merge the versatility of Machine Learning (ML)-based detection with cross-layer strategies. Although these systems are highly effective at identifying attacks, the way they are evaluated remains fragmented. Current practice follows two separate approaches: metrics that capture detection performance (like accuracy) and those that measure efficiency (such as energy consumption or latency). This diversification creates a challenge because an IDS is never a one-size-fits-all solution. Due to the inherent heterogeneity and strict resource constraints of IoT networks, the adoption of such a system, especially one that adopts a cross-layer strategy, requires a standardized, multifaceted evaluation framework that accounts for both detection capability and operational overhead.

In our earlier work [6], we compared the detection performance of single-layer and cross-layer ML approaches against a specific set of attacks. We found that cross-layer methods hold a clear advantage on the subset of metrics that are most commonly reported. The aim of the present paper is to build a framework for cross-layer ML IDSs to evaluate the full spectrum of their performance. We then validated this framework by applying it to state-of-the-art cross-layer ML IDSs. Moreover, rather than limiting ourselves to a particular attack type, we incorporated the set of the most severe, state-of-the-art DoS attacks with cross-layer impact as a coverage factor. The scope of this paper is to identify the challenges in current evaluation practices and to provide a standardized framework that addresses them, serving as a practical reference for IDS evaluation. Both the attacks and the IDSs were systematically identified through the PRISMA process. Hence, the concrete contributions of the paper are the following:

- Identification of state-of-the-art DoS attacks with cross-layer impact and of ML-driven cross-layer IDSs, following the structured PRISMA methodology.
- Design of a framework for the multi-criteria evaluation of cross-layer ML IDSs.
- Application of the framework to the identified IDSs according to scenarios in prominent application domains, where AHP is integrated to resolve trade-offs and sensitivity analysis is utilized to examine the stability of the derived outcomes under varying weight assignments and missing data assumptions.

The remainder of the paper is organized as follows. Section 2 provides the necessary background. Section 3 provides the description of the overall methodology workflow, the related work in the field of evaluating IDSs, and the application of the PRISMA process. Furthermore, it maps attack types to the impacted network layers and examines cross-layer ML IDSs in detail, outlining their architectural approaches, the ML models they rely on, their data fusion and correlation strategies and the challenges that emerge when trying to evaluate them holistically. Section 4 presents the design of the evaluation framework, including its evaluation domains and the corresponding metrics, and introduces a method for deriving a unified score for each IDS. Section 5 validates the framework through a proof-of-concept scoring and ranking of the identified IDSs under a set of representative Use Cases. Finally, Section 6 concludes the paper and outlines the future work.

2. Background

2.1. IDS Systems Overview

IDSs serve as a vital “second line of defense” mechanism [7] in modern cybersecurity, especially within the heterogeneous and resource-constrained environments of the Internet of Things (IoT). While primary defenses like cryptography and security hardening techniques are essential, they struggle with the massive, diverse traffic of IoT networks. Consequently, an IDS, as the main event-source tool, is tasked with the continuous monitoring of system events to identify and alert administrators to malicious activities.

The architectural design of an IDS depends on the adopted strategy for data monitoring and analysis:

- Centralized Architecture: Processes sensor data at a single, central server [7,8].
- Distributed Architecture: Spreads the analytical workload across multiple network nodes [7,8].
- Hybrid Architecture: Integrates centralized management with distributed edge monitoring to balance high-level oversight with localized responsiveness [1,7,9].

For threat classification, IDSs choose from an array of detection mechanisms. These mechanisms fall into four categories:

- Signature-based (Knowledge-based): These systems function by comparing traffic against a database of known attack patterns [7,8].
- Anomaly-based (Behavior-based): This approach identifies deviations from a baseline of “normal” system behavior [1,7].
- Specification-based: This approach relies on defined rules that describe legitimate protocol behavior [7].
- Hybrid and Advanced Learning: Modern systems integrate multiple techniques, such as combining signature rules with anomaly detection engines, to maximize detection breadth while minimizing memory overhead [7,9]. Furthermore, the adoption of ML and Deep Learning (DL) has allowed systems to autonomously learn from complex network data [1,8,10].

The deployment strategy defines where the IDS sensing scope is integrated within the infrastructure:

- Host-based IDSs (HIDS) are installed on individual devices to monitor internal logs and memory.
- Network-based IDSs (NIDS) monitor traffic across entire network segments, providing broader visibility [1,7,8].
- Edge, Fog, and Cloud Deployment: For real-time response, IDS components may be deployed at the Edge or Fog layers to reduce latency [9,10]. High-complexity analysis and model training can be offloaded to the Cloud [9].

2.2. Challenges of Single-Layer IDS and the Cross-Layer Paradigm

Single-layer detection architectures exhibit critical weaknesses in modern networked systems, especially in resource-constrained IoT environments. Single-layer IDSs tend to detect only attacks on isolated layers and overlook system-level threats and cross-layer anomalies [2], while detection methods that depend on specific data sources and single layers prove inadequate [3]. Traditional single-layer approaches (e.g., application-layer antivirus, network-layer firewalls, middleware-layer hardware analysis) are not optimized for the identification of multilateral attack characteristics, and such narrow visibility often results in missed indicators or false positives, especially under energy and resource limitations [4].

Moreover, single-layer approaches cannot efficiently address cross-layer attacks, where an action on one layer affects another [5]. Their primary restriction is the difficulty in recognizing and countering cross-layer attacks that originate at one layer, and their impact traverses other layers [2,11,12]. Such attacks leverage the absence of cross-layer interaction (e.g., between MAC and network layers), leaving minor traces of activity on one layer, and inflict severe damage on another, bypassing detectors built for isolated threat models [5].

Additionally, this siloed approach fails to capture correlated, multi-vector attack signatures, where advanced threats like Distributed Denial-of-Service attacks generate simultaneous, subtle anomalies across various system sides [4] and a single-layer detector might flag an uncorrelated isolated event, leading to missed detections or false positives [4,5]. Consequently, metrics at a single layer (e.g., Packet Delivery Ratio) can be ineffective. For example, an attacker might use physical-layer jamming not as a primary goal but to impair MAC or network-layer functions, and a detector focused solely on physical-layer metrics would misinterpret the attack vector, yielding a low detection rate [5]. Ultimately, by ignoring risks at the system level and cross-layer dependencies, layer-isolated solutions appear as suboptimal solutions for the dynamic, coordinated threats facing IoT networks [2,3,13].

Cross-layer visibility is critical for boosting both the accuracy and the robustness of intrusion detection. Unlike single-layer detection, which underperforms against contemporary threats, cross-layer strategies offer a more holistic defense.

- **Comprehensive Coverage and Correlation:** By removing protocol layer boundaries, cross-layer designs facilitate anomaly and data correlation across layers [2,4,14]. Simultaneous monitoring of various layers makes cross-layer threats, which leave distinct, correlated traces, harder to miss [4]. Systems that fuse information across the stack can detect anomalies across layers, enhancing accuracy and minimizing the risk of false positives [4,15].
- **Countering Coordinated and Stealthy Attacks:** Emerging intelligent, stealthy attacks involve coordinated cross-layer actions that degrade performance while evading detection [11]. These cross-layer attacks exploit poor inter-layer collaboration (e.g., between MAC and routing) to maximize impact at minimal cost [11,12]. Single-layer detection fails at this point, as attackers can disrupt a target layer via minor activities on another [5]. Cross-layer visibility is thus essential for spotting such attack chains and low-intensity, multi-vector threats [5,16].
- **Dynamic Adaptation and Robust Defense:** An integrated cross-layer system may adapt detection parameters and response strategies dynamically, leveraging correlated threat signals from multiple layers [2]. This flexibility is vital in dynamic threat landscapes. Additionally, by cross-validating data integrity across layers and employing collaborative anomaly detection, cross-layer IDSs ensure stronger defense against data corruption and compromises at the system level [2]. This approach aims to improve not only detection efficacy but also overall security, adaptability, and service quality [14].

3. Methodology

3.1. Methodology Workflow

The objectives of this study, as illustrated in Figure 1, rest on two pillars. The first pillar is to identify the set of most severe, state-of-the-art cross-layer DoS attacks drawn from the pool of reported IoT attacks and to map them onto the network layers they affect. This attack set will then be used in the evaluation framework described later, where it serves as a coverage characteristic. We focus on DoS attacks because they aim at the most vulnerable element of the CIA triad [1] in IoT networks, i.e., availability. In resource-constrained

environments, availability is already fragile. Thus, DoS attacks can severely aggravate this fragility to the point that the availability pillar essentially collapses.

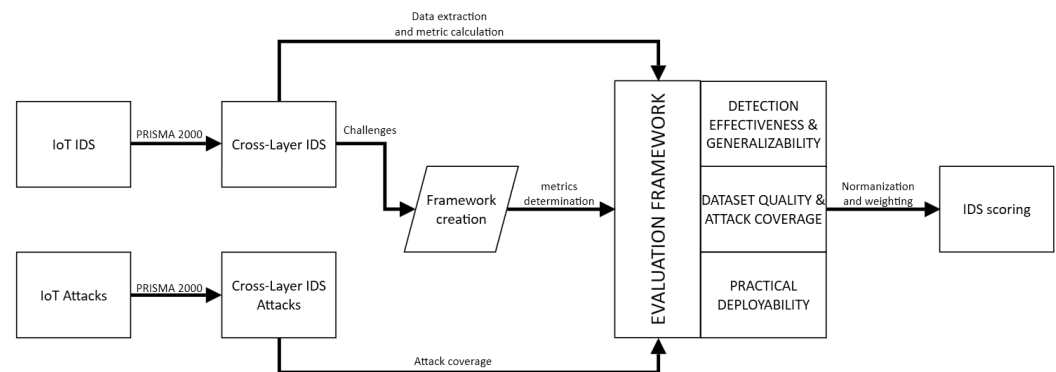


Figure 1. Methodology workflow.

The second pillar focuses on capturing the qualitative features of IDSs that rely on ML algorithms and perform cross-layer detection. In particular, we look at the architectural approaches they adopt, the ML models utilized by their inference mechanisms, and the data fusion/correlation strategies that give them their cross-layer trait. Finally, we examine the challenges that must be overcome to enable a comprehensive evaluation of these systems.

Building on the challenges that emerge, we will create an evaluation framework made up of metrics as a means of addressing these challenges. This framework is intended to assess existing IDSs and to act as a benchmark for future ones. Using it, the IDSs we identify will be scored according to the metrics they report.

To achieve our objectives, we followed the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology [17] to ensure a transparent and reproducible process. Based on the PRISMA approach, we derived the Review Questions shown in Table 1, guided by the following considerations:

- For both attacks and IDS, we limited our search to the period 2016–2026.
- The IoT DoS attacks examined for cross-layer impact in RQ3 were identified through a three-step process: first, we determined which IoT DoS attacks are covered in surveys and reviews from the 2016–2026 window (RQ1); next, we singled out those that had been evaluated in existing risk analyses and received the highest impact rating (RQ2); and finally, this subset was analyzed through the lens of RQ3.
- Although the risk analyses of RQ2 reference specific network protocols, the attacks themselves do not exploit protocol-specific vulnerabilities. Therefore, the attacks identified can be considered protocol-agnostic.

To determine the highest-severity attacks, we conducted a systematic review of existing risk analyses on IoT systems and cross-referenced findings from three distinct established qualitative risk-analysis methodologies: the EBIOS framework [18], the DREAD threat model [19], and the ETSI risk-assessment guidelines [20,21]. The targeted attacks, namely, blackhole, DDoS, flooding, jamming, selective forwarding, sinkhole and spoofing, were selected based on the following specific criteria and severity scales from these sources.

The EBIOS methodology adopts a four-level severity scale, comprising negligible, limited, important, and critical ratings. For an attack to meet the inclusion threshold within this framework, it had to be assessed as having a critical impact on the availability security objective in either of the examined domains (Things or Application).

Table 1. Research Questions.

Index	Research Question	Domain	Goal
RQ1	What IoT DoS attacks are included in reviews and surveys of the examination period?	Cross-layer attacks	To find the state-of-the-art IoT DoS attacks.
RQ2	What attacks have been included in risk analysis papers and receive the highest level of severity?		To identify which state-of-the-art IoT DoS attacks are considered high-severity threats.
RQ3	From the high-severity attacks of RQ2, which attacks display cross-layer impact or can be detected in a cross-layer way and at what layers?		To determine whether and how each attack impacts or is detected on more than one network layer.
RQ4	What are the architectural approaches followed for the design of cross-layer ML IDSs?	Cross-layer IDS	To identify the architectural approaches used in cross-layer ML IDS design.
RQ5	What ML algorithms are utilized for the design of cross-layer ML IDSs?		To identify the ML algorithms used in cross-layer ML IDS development.
RQ6	What data fusion/correlation strategies are employed during the feature selection process?		To identify the data fusion/correlation strategies used to establish cross-layer detection.
RQ7	What evaluation challenges arise considering the architectural approaches, the ML algorithms, and the data fusion process?		To identify the evaluation challenges of existing cross-layer ML IDSs.

The DREAD algorithm employs a qualitative scale from 1 to 3, corresponding to Low, Medium, and High severity across five distinct categories. The inclusion criterion was that an attack receive a rating of 3 (High severity) specifically on the Damage potential category.

The ETSI guidelines classify attack impact as Minor, Moderate, or Significant. The inclusion threshold was set at an attack receiving a Significant impact rating specifically for the availability security objective.

As is evident from the above, each risk analysis methodology operates with a different scaling system. Our general inclusion criterion was therefore defined as an attack receiving the highest degree of severity according to its respective methodology. In cases where conflicts emerged, for instance, when a particular attack was rated with the highest severity in one study but only the second-highest in another, we adopted a conservative strategy where if an attack triggered the highest impact rating within any of the reviewed risk analysis sources, it was categorized as a highest-severity threat.

To locate suitable studies, we searched the IEEE Xplore, MDPI, ScienceDirect, and Springer digital libraries with formulated queries, as displayed in Table 2.

To ensure a rigorous and structured review process, the screening of the retrieved literature was divided into distinct, collaborative phases among the authors. The initial screening phase, which involved evaluating the titles and abstracts of the 3631 identified records, was conducted independently by the primary author. Following this baseline selection, the remaining 435 reports sought for retrieval were subjected to full-text eligibility assessments, a phase in which all authors actively participated to ensure thorough evaluation against the established Research Questions.

For the stages of full-text assessment and data extraction where multiple authors were involved, a consensus-driven approach was implemented to resolve any discrepancies. In instances where opinions differed regarding the eligibility of a specific study, the authors collectively reviewed the paper against the precise inclusion and exclusion criteria defined for each Research Question. Final inclusion was determined through open discussion until a unanimous agreement was reached among all co-authors.

In instances where a study's alignment with the established inclusion criteria remained ambiguous after the initial keyword and abstract screening, the study was advanced to the full-text eligibility phase for a more granular evaluation. These borderline cases were then collectively reviewed by all co-authors, and any discrepancies regarding their technical scope or systemic relevance were resolved through open discussion until a unanimous consensus was achieved.

The **inclusion criteria** for each Research Question were defined as follows:

- **Research Question 1:** Eligible studies must be reviews or surveys who focus on Denial-of-Service (DoS) attacks within Internet of Things (IoT) environments.
- **Research Question 2:** Publications must present a formal security risk analysis or severity assessment mapped to IoT network infrastructures.
- **Research Question 3:** Papers must investigate specific high-severity attack vectors, namely, blackhole, DDoS, flooding, jamming, selective forwarding, sinkhole, or spoofing, within the context of IoT, while describing their operational impact across multiple protocol layers.
- **Research Questions 4–7:** Included literature must propose an Intrusion Detection System that utilizes Machine Learning models as its primary inference engine and employs a cross-layer paradigm for threat detection.

The reduction in papers based on each Research Question's exclusion criteria is shown in Figure 2. Particularly, the **exclusion criteria** for each Research Question were defined as follows:

- **Research Question 1:** We excluded reports that did not describe or categorize specific attack types, as they lacked the necessary granularity to inform our attack taxonomy and impact analysis.
- **Research Question 2:** We excluded reports that did not provide a quantitative or qualitative severity analysis of DoS attacks in IoT, as they did not offer the necessary insights into attack criticality required for our impact assessment.
- **Research Question 3:** We excluded:
 - Studies that focused on a single network layer, as they failed to capture the cross-layer propagation characteristic of the attacks under investigation.
 - Studies that addressed general security topics without focusing on specific attack types, as they were not relevant to our attack-centric analysis.
 - Studies that lacked explicit IoT context, as they were not applicable to the IoT-specific attack landscape and its unique constraints.
 - Studies that examined multi-layer contexts unrelated to IoT, such as cloud infrastructures or cross-layer neural networks, as they did not align with our protocol stack focus.
- **Research Questions 4–7:** We excluded:
 - Survey or review articles, as they did not propose novel detection systems.
 - Studies that proposed single-layer detection approaches, as they did not align with our cross-layer focus.
 - Studies that addressed security topics other than attack detection (e.g., prevention, key management), as they were outside our scope.
 - Studies that did not employ Machine Learning as their core detection mechanism.
 - Conceptual frameworks without implementation or evaluation details, as they did not provide measurable outcomes.
 - PhD dissertations, as they were not peer-reviewed publications and fell outside our inclusion criteria.

Table 2. Search queries per Research Question and database.

RQ	Database	Query
RQ1	IEEE Xplore	("All Metadata":IoT OR "All Metadata": "Internet of Things") AND ("All Metadata":DoS OR "All Metadata": "Denial of Service") AND ("All Metadata":survey OR "All Metadata":review)
	MDPI	Keywords = (IoT OR "Internet of Things") AND (dos OR "denial of service") AND (survey OR review)
	SCIENCE DIRECT	Title, abstract, keywords: (IoT OR "Internet of Things") AND (dos OR "denial of service") AND (survey OR review)
	SPRINGER	[[All: iot] OR [All: "internet of things"]] AND [[All: dos] OR [All: "denial of service"]] AND [[All: survey] OR [All: review]]
RQ2	IEEE Xplore	("All Metadata":IoT OR "All Metadata": "Internet of Things") AND "All Metadata": "security risk analysis"
	MDPI	Keywords = (IoT OR "Internet of Things") AND "security risk analysis"
	SCIENCE DIRECT	Title, abstract, keywords: (IoT OR "Internet of Things") AND "security risk analysis"
	SPRINGER	[[All: iot] OR [All: "internet of things"]] AND [All: "security risk analysis"]
RQ3	IEEE Xplore	"All Metadata": "cross-layer" AND "All Metadata":blackhole AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata":ddos AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata":flooding AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata":jamming AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata": "selective forwarding" AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata":sinkhole AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
		"All Metadata": "cross-layer" AND "All Metadata":spoofing AND ("All Metadata": "IoT" OR "All Metadata": "internet of things")
	MDPI	Keywords = "cross-layer" AND blackhole AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND ddos AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND flooding AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND jamming AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND "selective forwarding" AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND sinkhole AND (IoT OR "internet of things")
		Keywords = "cross-layer" AND spoofing AND (IoT OR "internet of things")
	SCIENCE DIRECT	Title, abstract, keywords: "cross-layer" AND blackhole AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND ddos AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND flooding AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND jamming AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND "selective forwarding" AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND sinkhole AND (IoT OR "internet of things")
		Title, abstract, keywords: "cross-layer" AND spoofing AND (IoT OR "internet of things")
	SPRINGER	[All: "cross-layer"] AND [All: blackhole] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: ddos] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: flooding] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: jamming] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: "selective forwarding"] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: sinkhole] AND [[All: iot] OR [All: "internet of things"]]
		[All: "cross-layer"] AND [All: spoofing] AND [[All: iot] OR [All: "internet of things"]]
RQ4–7	IEEE Xplore	("All Metadata": "cross-layer" AND ("All Metadata":IoT OR "All Metadata": "Internet of Things") AND ("All Metadata": "intrusion detection" OR "All Metadata":IDS) AND ("All Metadata":ML OR "All Metadata": "Machine Learning"))
	MDPI	Keywords = "cross-layer" AND (iot OR "internet of things") AND ("intrusion detection" OR IDS) AND (ML OR "machine learning")
	SCIENCE DIRECT	Title, abstract, keywords: cross-layer AND (iot OR "internet of things") AND ("intrusion detection" OR IDS) AND (ML OR "machine learning")
	SPRINGER	[All: "cross-layer"] AND [[All: iot] OR [All: "internet of things"]] AND [[All: "intrusion detection"] OR [All: ids]] AND [[All: ml] OR [All: "machine learning"]]

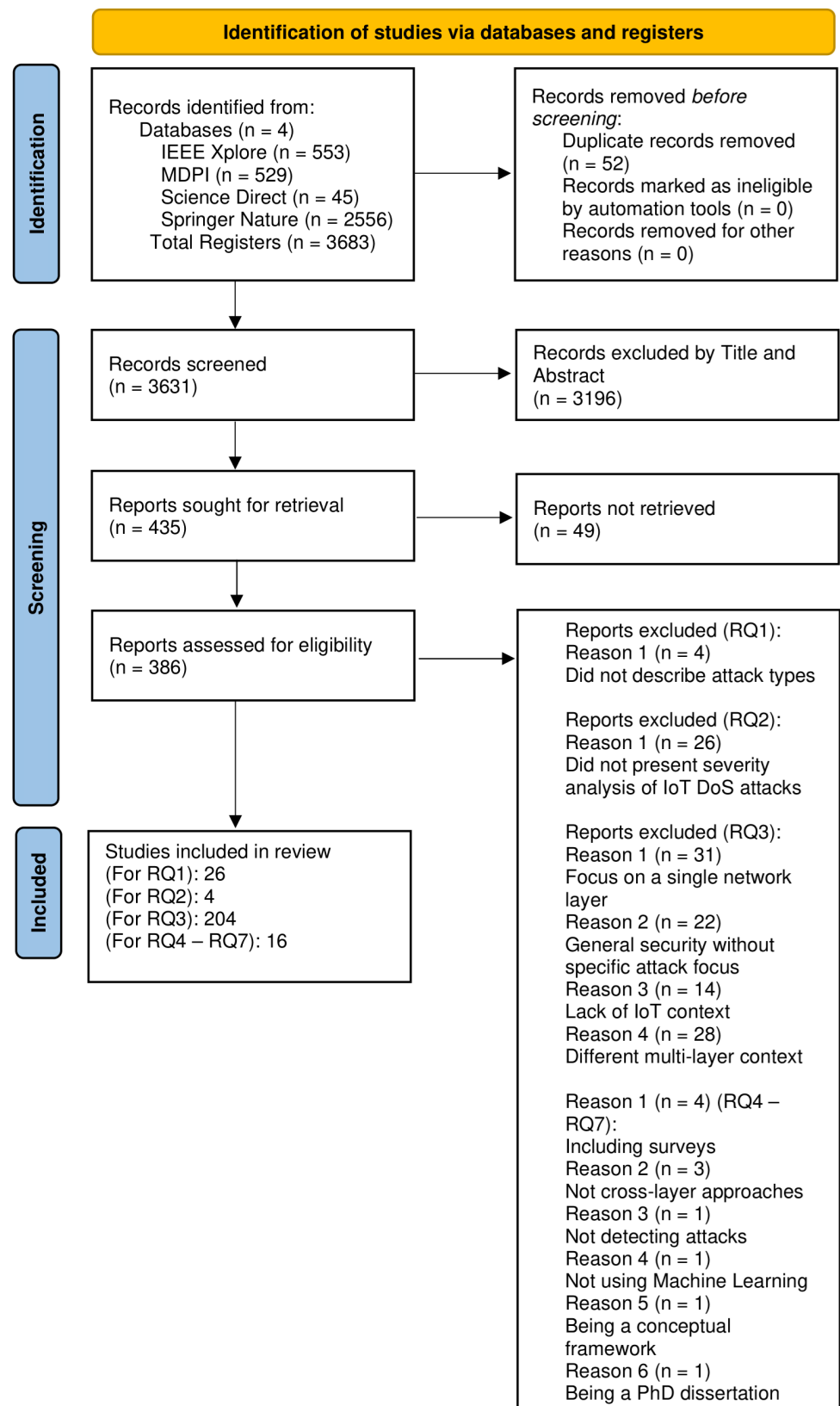


Figure 2. PRISMA 2020 checklist applied to Research Questions.

3.2. Related Work

The evaluation of IDSs, particularly those employing ML and DL techniques, has received considerable attention in the cybersecurity literature. Cheung [22] proposed an ML-based intrusion detection framework with emphasis on the Precision–Recall Area

Under Curve (PR-AUC) metric to evaluate model performance under class imbalance conditions. Their framework includes a unified preprocessing pipeline designed to prevent information leakage and employs SHAP-based feature contribution analysis to enhance model interpretability. The evaluation strategy focuses exclusively on detection performance metrics, specifically, PR-AUC, Recall, Precision, and Macro F1, using five-fold cross-validation and a two-stage hyperparameter optimization approach (Random Search followed by Grid Search) on the NSL-KDD dataset.

Hadri and Arabi [23] developed a cross-dataset evaluation framework that systematically compares eight ML techniques across ten validated datasets. Their evaluation strategy employs rigorous statistical methodology with automated dataset validation. The evaluation focuses on six essential security performance indicators: Accuracy, Precision, Recall, F1-Score, False Positive Rate, and ROC-AUC. They implement both intra-dataset (stratified cross-validation) and cross-dataset (train on one dataset, test on another with feature adaptation) validation protocols. Their evaluation strategy demonstrates exceptional generalization capability; however, the evaluation remains exclusively focused on algorithmic performance metrics.

Gharib et al. [24] introduced an evaluation framework specifically designed for assessing intrusion detection datasets rather than IDS models themselves. Their evaluation strategy focuses on eleven criteria: complete network configuration, complete traffic, labelled dataset, complete interaction, complete capture, available protocols, attack diversity, anonymity, heterogeneity, feature set and metadata. The framework employs a weighted scoring mechanism using a flexibility coefficient W (definable based on organization requirements) and subfactor coefficients V (based on empirical distributions). Attack categories are weighted according to the 2016 McAfee report distribution, while protocol distribution is based on observed traffic at their research center. Their evaluation strategy was applied to eleven available datasets (DARPA, KDD'99, DEFCON, CAIDA, LBNL, CDX, KYOTO, TWENTE, UMASS, ISCX2012, ADFA). The evaluation focuses on dataset quality assessment.

Praveen et al. [25] addressed the critical issue of model adaptability in the context of zero-day attacks, proposing an evaluation framework with four core stages: Ingestion & Storage, Pre-processing, Modeling, and Evaluation & Reporting. Their evaluation strategy focuses on Zero-Day attack evaluation (training on one dataset and testing on another to simulate unseen attack patterns) and cross-dataset generalization (testing models on multiple dataset pairs). Evaluation metrics include Accuracy, Precision, Recall, F1-score, AUROC, AUPRC, and calibration metrics (Expected Calibration Error). The evaluation focuses on quantifying performance degradation and adaptability.

Gungor et al. [26] investigated the resilience of IDS and developed a rigorous evaluation framework specifically targeting adversarial attacks against ML-based IDSs in IoT environments. Their evaluation strategy employs a wide range of adversarial attack techniques, including white-box (FGSM, BIM, PGD, DeepFool, JSMA, Elastic Net), gray-box (transfer attacks from DNN to other models), and black-box (ZOO) attacks across four realistic and recent IoT intrusion datasets (MQTTset, WUSTL-IIoT, X-IIoTID, Edge-IIoTset). The evaluation uses macro F1-score due to dataset imbalance, with the methodology including data pre-processing, model training with hyperparameter optimization, and adversarial attack generation using Adversarial Robustness Toolbox (ART). The evaluation focuses exclusively on IDS robustness without addressing generalizability, dataset quality, or resource efficiency.

Nguyen et al. [27] proposed an automated benchmarking framework designed to evaluate anomaly-based intrusion detection solutions. Their evaluation strategy comprises two main blocks: Dataloader (automating download and preparation of twelve popular

datasets) and Benchmarking (automating optimization of selected models with Optuna framework, providing baseline results of state-of-the-art models). The evaluation compares three machine learning models (XGBoost, LightGBM, KNN) and three deep learning models (SAINT, TabTransformer, TabNet) across five datasets (CIC-IoT2023, CIC-DDoS2019, UNSW-NB15, CIDD001, CIC-IDS2018). Evaluation metrics include Accuracy, TPR, FPR, F1, training time, and testing time. The evaluation strategy focuses on performance metrics and computational time.

The proposed work proposes a multi-dimensional evaluation framework designed for cross-layer machine learning-based IDS in IoT environments. The framework integrates three complementary dimensions that address detection effectiveness and generalizability, dataset quality and attack coverage and practical deployability. The framework offers flexible weighting allowing stakeholder-specific prioritization based on deployment requirements and provides a scalable scoring system that enables fair comparison across different IDS implementations, addressing the fragmentation identified across all reviewed frameworks.

Table 3 summarizes this related work in comparison with our work in terms of evaluation focus, weight-based scoring and cross-layer assessment.

Table 3. Summary and comparison of the related and proposed work.

Paper	Evaluation Focus	Adaptable Weight-Based Scoring	Cross-Layer Assessment
[22]	Detection Effectiveness	X	X
[23]	Detection Effectiveness	X	X
[24]	Dataset Quality	✓	X
[25]	Adaptability under Zero-Day Attacks	X	X
[26]	IDS Resilience against Adversarial Attacks	X	X
[27]	Detection Effectiveness and Training/Testing Time	X	X
Proposed work	Detection Effectiveness and Generalizability; Dataset Quality and Attack Coverage; Practical Deployability	✓	✓

3.3. Attack Type-to-Network Layer Mapping

The attacks that emerged from RQ2 were examined for their cross-layer impact using the queries formulated for RQ3. The findings on how these attacks affect multiple network layers are summarized below:

Blackhole targets the network layer [28,29]. Its impact crosses to the data link layer (CTS rushing, MAC timing), transport layer (ACK spoofing), and application layer (data integrity violation) [30–32].

DDoS targets the network, transport, and application layers [33–35]. Their cross-layer impact affects the physical layer (bandwidth exhaustion) [36].

Flooding targets the network, transport, and application layers [37,38]. Cross-layer impacts include data-link-layer congestion and increased collisions, transport-layer congestion and application-layer service unavailability [39,40].

Jamming primarily targets the physical and data link layers by transmitting high power or interfering signals [41–43]. Its cross-layer impact reaches the network layer (routing disruption), transport layer and application layer (increased end-to-end delay, mission disruption) [44,45].

Selective forwarding primarily targets the network layer [46,47]. The effects of this attack impact the application layer (QoS, throughput) and transport layer (end to end delay) [46,48,49].

Sinkhole targets the network layer [50,51]. Cross-layer impact includes increased delay and reduced throughput in transport layer and application-layer data integrity compromise [52].

Spoofing targets the data link and the network layer [53–55]. Cross-layer impact extends to the physical layer (spoofing causes cascading failures), transport layer (spoofing of TCP segments) and application layer (false data injection [55–58]).

3.4. Cross-Layer Machine Learning Intrusion Detection Systems

This section examines key qualitative characteristics of cross-layer IDSs that employ ML as their detection strategy. No single IDS fits all deployment scenarios, especially in resource-constrained IoT environments. Limitations in computing power, energy availability, and the inherently dynamic nature of IoT networks require carefully developed solutions that target on efficiency and minimal resource consumption.

Therefore, three design aspects deserve close attention: architectural approaches to deployment, the choice of machine learning models and strategies for correlating data across network layers. Each aspect carries significant implications. A centralized architecture may overload individual nodes, increasing both energy use and computational demand. Conversely, a decentralized approach can introduce considerable network overhead. Cross-layer data fusion is essential for realizing the benefits of cross-layer detection, yet the specific fusion method may generate additional computational overhead, thereby affecting power consumption. Meanwhile, the selected ML algorithm determines model size, which in turn influences system memory requirements. Beyond these design considerations, we also examine the evaluation challenges inherent in adopting cross-layer ML detection methodologies for IoT networks.

3.4.1. Architectural Approaches

Regarding system design, the three architectural patterns discussed in Section 2.1, i.e., centralized, distributed, and hybrid, also emerge from our review. Table 4 gives a comprehensive view of these approaches.

Centralized architectures dominate the landscape. The majority of the studies (56%) fall into this category. In these designs, a single controller, often an SDN controller, a border router, or a cloud node, collects data and makes the final detection decisions. For instance, the SDN-based framework in [59] uses a central control plane to manage the entire network. Others, like CrossSentry [4], take a different approach by focusing on the host side. CrossSentry collects metrics directly from individual devices across multiple layers. A few centralized systems also incorporate edge processing, as seen in [60], where information is pre-processed near the user before being sent to the cloud.

Distributed architectures appear in 38% of the studies. These designs spread detection responsibilities across multiple nodes, which is especially useful when no central infrastructure exists. For example, a wireless ad hoc network lacks a natural central point, so the method in [61] relies on collaborative behavior analysis among neighboring nodes. In [62,63], sniffer nodes capture traffic and forward it to a border router. A more sophisticated setup in [64] employs dedicated sniffers for local detection and a super node for global coordination. The system in [65] takes a different route by using federated learning, where nodes work together to train a global model while keeping their data private.

Hybrid architecture appears only once, but it offers an interesting insight. The lightweight cooperative IDS in [66] deploys agents that have been selected automatically across the network but also maintains a centralized border router. The agents handle local detection, while the router correlates alerts for a global view.

Table 4. Architectural approaches in cross-layer ML-IDS.

Paper	Approach	Distribution Method	Host/Network
[59]	Centralized	—	Network
[4]	Centralized	—	Host
[65]	Distributed	Federated Learning	Network
[67]	Centralized	—	Network
[62]	Distributed	Sniffer Nodes	Network
[68]	Centralized	—	Network
[60]	Distributed	Edge-based	Network
[69]	Centralized	—	Network
[70]	Centralized	—	Network
[71]	Centralized	—	Network
[61]	Distributed	Collaborative Node Analysis	Network
[72]	Centralized	—	Network
[73]	Centralized	—	Network
[66]	Hybrid	Distributed Agents	Network
[64]	Distributed	Dedicated Sniffers + Super Node	Network
[63]	Distributed	Sniffer Nodes	Network

3.4.2. ML Models and Correlation Strategies

The choice of learning algorithm varies widely across the reviewed systems. Table 5 summarizes the use of the variety of models.

Traditional machine learning is the prevalent choice for cross-layer detection. About 88% rely at least partly on classical ML algorithms. Among them, Random Forest appears in the majority of papers. Decision Trees and k-Nearest Neighbors follow closely. Support Vector Machines often appear in hybrid or ensemble settings.

Ensemble methods, which combine multiple base classifiers to improve detection performance, are also widely adopted in cross-layer IDS designs. The framework in [68] combines nine base classifiers with soft/hard voting, stacking, and bagging. In [60], a stack of four base learners (RF, Gradient Boosting, AdaBoost, LightGBM) was integrated with an XGBoost meta learner, all tuned by an Improved Grey Wolf Optimizer. In [72], a simpler stacking approach was followed with KNN, SVM, DT, and LR feeding into a logistic regression meta model. In the case of [4], four anomaly detectors were blended (Isolation Forest, OC SVM, LOF) with a statistical method (Robust Z Score) to fuse their outputs via a weighted voting scheme.

Deep learning methods appear less but still in a significant portion of studies. In [59], a wide array of models were evaluated, i.e., MLP, LSTM, GRU, CNN, and a hybrid CNN LSTM. The study in [65] utilized RNNs, LSTMs, Graph Neural Networks and Transformers under a federated learning framework. In [73], a four-layer fully connected network was used with ReLU activations and a softmax output. In [66], a shallow ANN was used with a single hidden layer, designed specifically to run on resource-constrained RPL devices. It used ReLU and Hardmax to keep computations cheap and stores only the trained weights and biases on the device's ROM.

Neural networks of various kinds appear in fewer cases than traditional ML models, typically as part of hybrid or deep learning architectures. MLPs show up in [62,67,69,70]. Study [67] also includes a generic ANN alongside classical ML models.

Table 6 categorizes the correlation strategies that have been observed through the review. Three broad families emerge: statistical correlation, decision-based fusion, and hybrid approaches.

Statistical correlation methods appear in 38% of the works. The basic idea is to measure how strongly each feature relates to the target class. In [67], both Pearson correlation and

Mutual Information were used to catch linear and nonlinear relationships. The work in [62] utilized multivariate skewness and kurtosis in combination with Random Forest importance scores. Pearson alone shows up in [71,72]. The work in [63] ranked features with Random Forest and then normalized them via Quantile Transformation and Min Max scaling.

Decision-based fusion strategies appear in 31% of the studies. The focus here is on combining outputs from multiple detectors or layers by using some logic scheme. The work in [4] used four single-layer algorithms run in parallel, and a cross-layer alarm fired only if at least three of them agreed. The model in [65] explicitly fused cross-layer features before classification. The work in [68] employed a feature selection scheme which was guided by RPL protocol semantics, reflecting the relevant attack mechanisms directly. The works in [60,64] were also guided by decision-based selection to fuse features in a cross-layer way. The work in [73] concatenated 7 data-related features, 16 topology-related features, and 4 link-layer features into a single vector for its neural network.

Hybrid approaches combine techniques of multiple types. The work in [69] used recursive feature elimination with cross-validation (RFECV) to pick an optimal subset of 12 features out of 29 and then leveraged CatBoost's built-in ability to model feature interactions. In [70], the Ward algorithm was used for hierarchical feature clustering to balance datasets, and PCA was used to compress correlated features into ten uncorrelated principal components. The work in [61] employed Particle Swarm Optimization (PSO) to select the best features, and used an inverse negative exponential distribution to weight classification parameters. In [66], flow-based features were extracted (packet counts, byte volumes, durations) and then chi-squared selection is applied.

Table 5. Machine learning, deep learning, and neural network models in cross-layer ML IDS.

Paper	ML Models	DL Models	NN Models
[59]	DT, RF, KNN, GNB, AdaBoost, SVM, LightGBM, CatBoost	MLP, LSTM, GRU, CNN, CNN-LSTM	—
[4]	Isolation Forest, OC-SVM, LOF	—	—
[65]	—	RNN, LSTM, GNN, Transformers (with Federated Learning)	—
[67]	RF, KNN, DT, NB	—	ANN
[62]	LR, SVM, DT, RF	—	MLP
[68]	LR, SVM, DT, KNN, RF, ET, GB, AdaBoost, XGB (ensembles)	—	—
[60]	RF, Gr Boost, AdaBoost, LightGBM (base); XGBoost (meta)	—	—
[69]	CatBoost, RF, XGBoost, KNN, DT	—	MLP
[70]	RF, DT, KNN	—	MLP
[71]	Stage 1: DT, KNN, SVM; Stage 2: DT, RF, LR, KNN, GB	—	—
[61]	Enhanced SVM (E-SVM)	—	—
[72]	KNN, SVM, DT, LR (base); LR (meta); stacking	—	—
[73]	—	—	4-layer NN (ReLU, Softmax)
[66]	—	—	Shallow ANN (1 hidden, ReLU, Hardmax)
[64]	C4.5 DT (local); Iterative Linear Regression (global)	—	—
[63]	Random Forest (RF)	—	—

Table 6. Data fusion and correlation strategies in cross-layer ML IDS.

Paper	Data Fusion/Feature Selection Strategy	Strategy Type
[59]	Temporal, statistical, and protocol-level feature combination	Hybrid approach
[4]	Weighted cross-layer decision: 3/4 single-layer algorithms must agree	Decision-based fusion
[65]	Selection of cross-layer features	Decision-based fusion
[67]	Pearson correlation, Mutual Information	Statistical correlation
[62]	Pearson correlation, multivariate skewness/kurtosis, RF importance	Statistical correlation
[68]	RPL-semantic guidance	Decision-based fusion
[60]	Base predictions, meta-learner	Decision-based fusion
[69]	RFECV, CatBoost feature interactions	Hybrid approach
[70]	Ward clustering, PCA transformation	Hybrid approach
[71]	Pearson Correlation	Statistical correlation
[61]	PSO optimization, inverse exponential distribution	Hybrid approach
[72]	Pearson Correlation	Statistical correlation
[73]	Direct fusion of data, topology, and link-layer features	Decision-based fusion
[66]	Flow-based features, Chi-squared statistic	Hybrid approach
[64]	Fusion into one feature vector	Decision-based fusion
[63]	RF feature ranking, Quantile Transformation; Min-Max scaling	Statistical correlation

3.4.3. Evaluation Challenges

A look at the evaluation practices in the reviewed literature reveals that most studies pay attention to detection performance metrics, such as accuracy, precision, recall and F1 score. These are signs of how well a detection system performs after its training. But they miss some aspects that are just as important for IDS deployments: the models' generalizability, the datasets' coverage and their practical deployability. From the conducted review, we have focused on the these challenges as follows:

Model generalizability: most studies evaluate their models under conditions similar to the training environment. Generalizability is rarely examined as an evaluation criterion. Only 12.5% of the reviewed works evaluate their models on datasets that are entirely different from the one used for training. Another 12.5% go a step further by testing on attacks that were not present during training as an attempt to evaluate how well the system detects unknown threats or zero-day attacks.

Dataset coverage: The methods do not adequately evaluate their training dataset, since only a minority prioritizes training their models to adapt to diverse conditions. Only 37.5% of the studies include topological variations, i.e., different area sizes or different numbers of nodes. Even fewer (12.5%) expose the model to varied network conditions during training, such as mixing static and mobile nodes or simulating nodes with different operational states.

Practical deployability: Only a minority of the studies evaluate energy consumption or execution time. Table 7 summarizes the efficiency metrics of each study regarding their practical deployability, in contrast to the corresponding detection performance metrics. A few, like [4,68], report CPU utilization or training time. The work in [61] includes energy metrics (average and remaining energy) together with network performance indicators such as delay, jitter, and packet delivery ratio. The choice between a centralized or distributed IDS architecture has a direct energy impact on every participating node. A centralized scheme might save per node computation. However, it incurs higher communication overhead. A distributed scheme spreads the load but may increase latency and coordination cost. Without measuring energy and time, it is impossible to make an informed decision. Another aspect of this point is the model size. Only one study in our set [66] attempts to estimate memory requirements for a particular node with and without the IDS installed. No other work reports model size directly. Cross-layer detection, by its nature, fuses features from

multiple network layers, which tends to increase the number of input features. Some machine learning algorithms, e.g., deep neural networks or complex ensembles, also produce relatively large models. On the one hand, we gain detection performance and the ability to spot sophisticated attacks. On the other hand, a larger model may not fit on an extremely constrained node, or it may leave too little room for the application itself. The trade-off between cross-layer efficiency and model compactness should be quantified.

It is evident from the aforementioned challenges that an IDS is not a one-size-fits-all solution. IoT networks are diverse, and the nodes that make them up are in most cases heterogeneous. Thus, making the appropriate selection often comes down to the specific characteristics of each network. These very characteristics will determine the effectiveness and applicability of an IDS integrated within a particular network.

Table 7. Detection performance and efficiency metrics in cross-layer ML-IDS.

Paper	Detection Performance Metrics	Efficiency Metrics
[59]	Accuracy, Precision, Recall, F1-score, False Positive Rate (FPR) for DL: ROC-AUC ANOVA with FDR correction, PCA	—
[4]	Detection Rate, FPR	Average CPU Utilization (ms) and memory usage (RAM in MB)
[65]	Accuracy, Precision, Recall, F1-score, FPR	—
[67]	Accuracy	—
[62]	Accuracy, Precision, Recall, F1-score, AUC, FPR	Computational Time (model building time)
[68]	Accuracy, Weighted Precision, Weighted Recall, Weighted F1-score, k-fold CV accuracy, Overfitting Gap, ROC-AUC, Confusion Matrices	Computational Cost: training time, inference latency per sample, memory footprint
[60]	Accuracy, Macro/Micro Precision, Macro/Micro Recall, Macro/Micro F1-score, ROC Curve/AUC, Error Rate	—
[69]	Recall, FPR, Precision, NPV, Specificity, ROC-AUC	—
[70]	Accuracy, Precision, Recall, F1-score, ROC-AUC	Testing Time (TT); Memory Usage (MiB)
[71]	Accuracy, Weighted Average F1-score, FPR, Cross-validation Score	Testing/Execution Time
[61]	—	Energy Metrics: avg energy consumption, avg remaining energy. Traffic Metrics: PDR, Throughput, Goodput, Dropping ratio. Network Performance: Delay, Jitter, NRO, Collision rate. Utilization: Channel utilization ratio.
[72]	Accuracy, Cross-validation Score	—
[73]	Detection Rate, FPR, Accuracy, Confusion Matrix	—
[66]	Accuracy, Detection Rate, FPR, Specificity, FNR	Energy consumption (Cooja Energest), memory requirements (RAM/ROM on Tmote Sky)
[64]	Correctly Classified Instances, Accumulated Measure of Fluctuation	—
[63]	Accuracy, Precision, Recall, F-score	—

4. Proposed Evaluation Framework and Considered Dimensions

4.1. Evaluation Framework Description

To address the evaluation challenges that emerged from our review of cross-layer IDSs in Section 3 and to bring some standardization to multi-criteria assessment, we propose the following framework.

Our framework assigns a score based on a predefined set of metrics, which are grouped into three dimensions. Together, these dimensions cover the main aspects any IDS should be evaluated on: (1) detection effectiveness and generalizability, (2) dataset quality and coverage and (3) practical deployability. Table 8 lists each metric alongside its required inputs and shows how they are grouped into the three evaluation dimensions. Each dimension can be weighted independently, which lets evaluators express the trade-offs of their networks. A system that excels in one dimension may not be a good fit for a different network where that dimension matters less.

Table 8. Dimensions and metrics for evaluation framework.

Dimension	Subcategory	Metrics	Inputs
DETECTION EFFECTIVENESS & GENERALIZABILITY	Core Metrics	Accuracy	TP, TN, FP, FN
		Recall	TP, FN
		Precision	TP, FP
		F1-score	TP, FP, FN
	Advanced Metrics	Adaptability Score	Network scenarios, Core Metrics
		Novelty Detection Rate	Total number of novel attack instances, FP_{novel} , TP_{novel} , Novel attack class in test set
DATASET QUALITY & ATTACK COVERAGE		Attack Taxonomy Coverage Index	Number of attacks covered
		Dataset Diversity Requiremen	Protocol topology, Attack operation mode, Data origin, Network conditions
		Dataset Balance Index	Total number of dataset instances, Number of instances per class
		Layer Fusion Indicator	Number of correlated layers, Layer-fusion technique
PRACTICAL DEPLOYABILITY		Training Time Index	Coremark benchmark, Training time record
		Detection Time Index	Coremark benchmark, detection time record
		Memory Footprint Index	Available system memory, IDS process memory
		Model Size Index	Available persistent storage, Serialized model size
		Expanded Energy Index	Coremark benchmark, Active detection measurement, Idle measurement

The framework is modular. The developer of an IDS can evaluate their system on a subset of the metrics if some are not applicable; for example, accuracy is not ideal for heavily imbalanced datasets and a testbed may not support a wide variety of network setups. However, the more metrics are included, the more complete the picture. Each metric is scored on a scale from 0 to 1. Because the final score accounts for all metrics, using more of them gives a higher possible ceiling. Missing metrics receive zero points, which is equivalent to the worst possible performance on those metrics.

It should be noted that, while the quantification of metrics is inherently an experimental process, the proposed framework is intended as a descriptive tool that promotes evaluation standardization. In this context, its objective is to provide guidance for future work on how to assess and validate different aspects of IDS performance through the application of the metrics defined within the framework.

Dimension 1—Detection Effectiveness & Generalizability: This dimension brings together metrics that capture how well an IDS detects attacks, not only the ones it was trained on—in the same network environment—but also previously unseen attacks and in situations where the deployment conditions differ from the training environment.

a. Core Metrics

The core metrics subcategory comprises the standard metrics used to evaluate detection performance. As noted in [6], these metrics, or a subset of them, are widely adopted across the majority of studies employing an ML model for attack detection.

a.1 Accuracy

Accuracy is the ratio of all correct classifications ($TP + TN$), whether of normal or malicious traffic, to the total number of classifications ($TP + TN + FP + FN$). It functions as a coarse-grained metric of model quality. It is calculated by:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

a.2 Precision

Precision is the ratio of the model's correctly classified attack instances to all instances classified as attacks. A high precision value means few false alarms. It is measured by:

$$Pre = \frac{TP}{TP + FP} \quad (2)$$

a.3 Recall

Recall is the ratio of all correctly classified attack instances to the total number of attack instances. A high recall value means that most attack instances get detected. Hence,

$$Rec = \frac{TP}{TP + FN} \quad (3)$$

a.4 F1 score

The F1 score is the harmonic mean of precision and recall. It balances the importance of these metrics, as there is often an inverse relationship between them. Therefore,

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4)$$

b. Advanced Metrics

The metrics in this subcategory extend beyond standard detection performance to evaluate a model's ability to generalize across varying conditions and different attack environments.

b.1 Adaptability Score

Adaptability Score (**AdSc**) measures an IDS's capacity to maintain performance when evaluated in network environments different from its training conditions. This reflects its detection robustness to the dynamic nature of IoT networks. The purpose of this metric is to evaluate how well an IDS can generalize across different network topologies and conditions without the need for major training or fine-tuning processes. The following steps outline the calculation procedure

for this metric: (a) Initially, a set of S diverse network scenarios is defined. The S is not strictly defined and depends on the initial network implementation and its capabilities. These scenarios should align with some or all of the criteria of the following list:

- Different topological structures, such as different node density or network scale (topological adaptability);
- Communication through different protocols (protocol adaptability);
- Inclusion of diverse types of devices (device heterogeneity);
- Inclusion of a variety of traffic conditions, such as different traffic load, mobility of nodes or variable link quality (condition adaptability).

To calculate the $AdSc$, the model is firstly evaluated across the applicable Core Metrics (CM) for the set of S scenarios. As has been mentioned earlier, some (CM) may not be representative due to the training dataset characteristics, so the developers of the IDS omit them. For each (CM), the average performance drop ($\overline{PD}$) is determined for the new scenarios relative to the baseline ($BASE$) using the following equation:

$$\overline{PD}_{CM} = \frac{1}{S} \sum_{i=1}^S \frac{\delta_i}{CM_{BASE}} \quad (5)$$

$$\delta_i = \begin{cases} 0, & CM_{BASE} \leq CM_i \\ CM_{BASE} - CM_i, & CM_{BASE} > CM_i \end{cases}, CM = \{Acc, Pre, Rec, F1\}$$

After the calculation of performance drop for each Core Metric, the $AdSc$ is determined as follows:

$$AdSc = 1 - \frac{1}{n} (\overline{PD}_{Acc} + \overline{PD}_{Pre} + \overline{PD}_{Rec} + \overline{PD}_{F1}) \quad (6)$$

The number n represents the number of calculated Core Metrics (CM). This calculation averages the performance drop in each Core Metric. By subtracting this average from 1, the result is normalized so that a score of 1 represents perfect adaptability with no performance degradation.

In the calculation of PD , we cap the result to 0 when the CM of a scenario is greater than that of the baseline. The reason is that $AdSc$ is a measure of performance degradation. In such a case, we have an improvement in performance, which means no degradation relative to the baseline scenario.

b.2 Novelty Detection Rate

The Novelty Detection Rate (**NDR**) is a metric that assesses an IDS's capacity to detect previously unseen attack types that are not included in the training dataset. It measures a system's generalization capability and robustness in the dynamic threat landscape.

The purpose of this metric is to address the problem that most IDSs are evaluated on a closed set of known attack classes present in the training data. This fails to evaluate how they perform against zero-day or unknown attacks.

To measure this metric, the model's test set shall include a number of attacks (characterized as novel) that are not part of the training set. The IDS must classify the instances of these attacks as "novel". The score is determined using the following equation:

$$NDR = \frac{TP_{NOVEL}}{NovelAttackInstancesinTestSet} \quad (7)$$

where TP_{NOVEL} is the correctly classified instances of the novel attacks, and $NovelAttackInstancesinTestSet$ refers to the total number of novel attack instances in the testing set.

Dimension 2—Dataset Quality & Attack Coverage: This dimension looks at the quality of the dataset used for evaluation. It considers things like whether the training data covers a variety of network conditions and attack types, as well as how well the idea of cross-layer detection is actually reflected in each IDS under test.

a. Attack Taxonomy Coverage Index

The Attack Taxonomy Coverage Index (**ATCI**) measures the breadth of attack types an IDS under evaluation covers relative to the set of attacks identified and presented in Section 3.3. The purpose of this metric is to assess how effectively an IDS detects the most severe state-of-the-art cross-layer IoT DoS attacks. It is calculated as the ratio of the attacks the IDS can detect to the total number of attacks:

$$ATCI = \frac{\text{number of attacks detected}}{\text{total number of attacks}} \quad (8)$$

b. Dataset Diversity Requirement

The Dataset Diversity Requirement (**DDR**) is a metric that evaluates the diversity and generalizability of the dataset used to train and evaluate an IDS. The purpose is to assess to what degree the IDS models are trained in fixed, optimized conditions or approach the diverse nature of real-time deployment.

The IDS is evaluated by the rubric in Table 9, which is structured around four diversity facets. Each facet contains specific indicators that serve as criteria. For each diversity facet, a score of 1 (for high score indicators) or 0 (for low score indicators) is awarded based on the extent to which these indicators are present within the IDS's dataset. If a dataset contains both low and high score indicators for the same facet, then a score of 1 is awarded. The value of the metric is estimated by averaging the scores of the diversity facets:

$$DDR = \frac{1}{4} \sum_{i=1}^4 DiversityFacetScore_i \quad (9)$$

Table 9. Dataset diversity requirement rubric.

Diversity Facet	Low Score Indicators	High Score Indicators
Protocol & Topology Coverage	single protocol fixed topology	uses data from multiple protocols (e.g., RPL and 6LoWPAN) or diverse topologies
Attack Scenario Realism	attacks conducted in separation fixed number of attackers	attacks conducted in coordination varied number of attackers
Network Condition Variance	ideal network conditions (no background traffic, stable links, no mobility)	congested channels channel noise node mobility benign anomalies
Data Source Fidelity	synthetic data from simulators	physical testbeds hybrid emulation

c. Dataset Balance Index

The Dataset Balance Index (**DBI**) quantifies how balanced the samples are across different classes in a dataset. For the estimation of this metric, the Shannon entropy

is used. The concept of entropy is selected in various cases, for example, in [74,75], to address class imbalance issues. In this framework, it is utilized for the estimation of the existing imbalance in the datasets under evaluation. The index is computed as the normalized Shannon entropy of the class proportions. Its value ranges from 0 (the dataset is extremely imbalanced) to 1 (all classes have the same number of samples). In the following equation, K represents the number of the dataset classes, and p_i is the ratio of samples belonging to class i (n_i) to the total number N of samples ($p_i = \frac{n_i}{N}$). Therefore,

$$DBI = -\frac{1}{\log_2(K)} \sum_{i=1}^K p_i \log_2(p_i) \quad (10)$$

d. Layer Fusion Indicator

The Layer Fusion Indicator (**LFI**) is a metric designed to assess: (a) the extent of the interrelation among the network layers, (b) the sophistication of data fusion method utilized for attack detection within the evaluated IDS and (c) the attempt to harmonize disparate feature semantics into a unified representation for effective attack detection. It reflects both the number of network layers whose features are correlated in the trained model, the complexity of the applied feature selection method and the attempt to ensure semantic consistency and meaningful correlation among the fused cross-layer features. The LFI is designed as a descriptive metric that captures the complexity of cross-layer fusion, not its effectiveness. It complements the effectiveness metrics included in Dimension 1 (Detection Effectiveness and Generalizability) and contributes to the trade-off analysis among dataset quality, detection effectiveness, and practical deployment considerations. For the scoring of this metric, we initially categorize the models into the classes of the following Table 10:

Table 10. LFI classes.

LFI Level	Number of Fused Layers	Semantic Consistency	Fusion Process
LFI-0	0	—	—
LFI-1	≤ 2	no	decision-based
LFI-2	≤ 2	no	algorithm-based or hybrid fusion
LFI-3	≤ 2	yes	decision-based
LFI-4	≤ 2	yes	algorithm-based or hybrid fusion
LFI-5	> 2	no	decision-based
LFI-6	> 2	no	algorithm-based or hybrid fusion
LFI-7	> 2	yes	decision-based
LFI-8	> 2	yes	algorithm-based or hybrid fusion

The LFI-0 class is awarded a score of 0, while each subsequent class increases the score by 0.125, with a maximum value of 1 for the LFI-8 class.

Dimension 3—Practical Deployability: This dimension focuses on how well an IDS can support real-time deployment. It looks at timing, processing demands, and energy consumption.

a. Training Time Index

The Training Time Index (**TTI**) is a metric that quantifies the time required to train the detection model on a given dataset as a proportion of the time needed by the hardware platform hosting the evaluated IDS to complete a predefined number of iterations of the Coremark benchmark test [76]. This benchmark test is selected to account for the hardware setup parameters such that the TTI is hardware-independent and comparable among different IDSs. For the calculation of this metric, two separate timing sessions must be measured. Firstly, the Coremark benchmark

test must be executed for a count of 200,000 iterations to ensure that timer noise and startup overhead do not affect the measurement. The execution time is recorded as T_{BENCH} . Secondly, the measurement of the training procedure must be recorded as T_{TRAIN} . This includes data pre-processing, feature extraction and the actual learning phase. TTI is then calculated as:

$$TTI = \frac{1}{1 + \frac{T_{\text{TRAIN}}}{T_{\text{BENCH}}}} \quad (11)$$

The result of the equation lies within the range (0,1), where values closer to 1 indicate a training time lower than the benchmark time, whereas values closer to 0 indicate a training time exceeding the benchmark time.

b. Detection Time Index

The Detection Time Index (**DTI**) is a metric analogous to the TTI that is intended to quantify the time required to classify a batch of test samples during the detection phase. For the calculation of this metric, the Coremark benchmark test is employed in the same manner as in the TTI metric, using T_{BENCH} as the reference value. In addition, the time required to process a batch of 10,000 test samples, from the reception of the network packet to the generation of the classification result (T_{DETECT}), is recorded. The equation of DTI is:

$$DTI = \frac{1}{1 + \frac{T_{\text{DETECT}}}{T_{\text{BENCH}}}} \quad (12)$$

This metric also lies within the range (0,1), with values closer to 1 indicate a detection time lower than the benchmark time, whereas values closer to 0 indicate a detection exceeding the benchmark time.

c. Memory Footprint Index

The Memory Footprint Index (**MFI**) represents the proportion of processing memory utilized by the operation of IDS, while it actively monitors and classifies network traffic, relative to the available system memory. It is computed by measuring the IDS running memory process (mem_{IDS}) and the available system memory ($\text{mem}_{\text{SYSTEM}}$) and then calculating the equation as follows:

$$MFI = \frac{1}{1 + \frac{\text{mem}_{\text{IDS}}}{\text{mem}_{\text{SYSTEM}}}} \quad (13)$$

d. Model Size Index

The Model Size Index (**MSI**) measures the persistent storage space occupied by the trained model (stor_{IDS}) relative to the total storage available for the IDS on the system hosting the IDS ($\text{stor}_{\text{SYSTEM}}$). Therefore,

$$MSI = \frac{1}{1 + \frac{\text{stor}_{\text{IDS}}}{\text{stor}_{\text{SYSTEM}}}} \quad (14)$$

e. Expanded Energy Index

The Expanded Energy Index (**EI**) measures the proportion of energy an IDS consumes during the detection phase relative to the energy usage of the system when the IDS is inactive. To compute this metric, energy consumption is measured over a fixed duration of 10 s under three separate conditions: (a) the system in an idle state (E_{IDLE}), with the IDS inactive; (b) while the CoreMark benchmark test is running (E_{BENCH}); and (c) while the IDS is actively classifying incoming samples (E_{IDS}). The interval of 10 s is chosen to ensure the system reaches a steady state and to reduce measurement

noise. The idle-state measurement is used to isolate baseline power consumption unrelated to either the IDS or the CoreMark workload. Thus,

$$EEI = \frac{1}{1 + \frac{E_{IDS} - E_{IDLE}}{E_{BENCH} - E_{IDLE}}} \quad (15)$$

A note on the independence of framework metrics: The proposed framework is descriptive in nature and does not assume statistical independence among its constituent metrics. Nevertheless, we treat certain pairs of metrics as conceptually distinct because they capture different stages of the system lifecycle. Specifically, the Training Time Index (TTI) and Detection Time Index (DTI) are considered separately. The TTI reflects the cost of model training, whereas the DTI captures the real-time detection latency during real-time operation. Similarly, the Memory Footprint Index (MFI) and Model Size Index (MSI) address distinct deployment constraints; the MFI pertains to runtime memory overhead, while the MSI relates to persistent storage requirements. Although these pairs may exhibit empirical correlation on specific hardware platforms, the framework preserves them as separate metrics to enable fine-grained analysis across different phases of the IDS lifecycle.

4.2. Framework Scoring

To determine the total score of an IDS through the evaluation framework, the following equation is used:

$$FrameworkScore = w_{D1} * Avg_{D1} + w_{D2} * Avg_{D2} + w_{D3} * Avg_{D3}, \sum_{i=1}^3 w_{Di} = 1 \quad (16)$$

The values Avg_{D1} , Avg_{D2} and Avg_{D3} represent the average of the metrics of each Dimension. We chose to average the metrics within each dimension, because the dimensions contain different numbers of metrics. Had we simply summed them, a dimension with more metrics would carry more weight in the final score, regardless of whether that dimension was actually intended to be more important. Averaging avoids this imbalance and keeps the dimension weights as the only factor controlling importance.

The values w_{D1} , w_{D2} and w_{D3} represent the importance weights of each Dimension, introducing in this way the concept of trade-off between evaluation facets. The values of each w_{Di} ranges from 0 to 1, while the sum of them equals to 1. The relative importance of each Dimension is likely to vary across stakeholders. While the framework serves as a benchmark that evaluates IDSs on their multiple aspects in a neutral way, the final scoring is affected by the particular requirements of a network. Consequently, the choice of weights is left to the evaluator's judgment. For the selection of the appropriate values, the stakeholder must set the required priorities among the Dimensions of the framework for the installation environment of IDS. The higher the priority, the greater the value for this weight Dimension. Evaluators may benefit from employing an algorithmic approach for weight assignment, such as the Analytic Hierarchy Process (AHP) [77], the Entropy Weight method [78], or a multi-criteria decision-making process [79]. The adoption of such methods can improve the objectivity and consistency of the evaluation process, particularly when stakeholders consider the characteristics of their operational environment, the intended detection scope, deployment constraints, performance requirements, and the relative importance of the evaluated Dimensions.

The framework is not intended to classify an IDS as inherently good or poor in absolute terms. A system may demonstrate strong performance under a particular configuration and, consequently, within a specific operational environment, while exhibiting reduced suitability or performance under different conditions. The purpose of the framework is to

support environment-aware evaluation by capturing such trade-offs rather than producing a universally applicable ranking.

The scoring system of the evaluation framework is useful in two ways. For IDS developers, it provides a clear template of metrics, that points them towards the dimensions where their systems should be evaluated and optimized in a multifaceted manner. For a stakeholder who wants to deploy one of the available IDSs, the scoring can be adapted to reflect the trade-offs they need to make, enabling them to rank the candidate solutions accordingly.

5. Proof-of-Concept Validation

5.1. Use Case Analysis

For the proof-of-concept validation of our framework and the assignment of the Framework Score to each IDS, the relevant metrics were either computed or extracted from the corresponding studies. Certain metrics, such as Accuracy and Precision, were directly reported by the authors of the original works. Other metrics, including the Dataset Balance Index and Dataset Diversity Requirement, were derived based on the evidence and information available in those studies. The resulting partiality in scoring arises from the limited availability of evidence for some of the novel metrics introduced by the framework, such as the Memory Footprint Index and Expanded Energy Index, which did not facilitate their reliable calculation across all evaluated IDSs. The computed metric values are presented in Table 11.

To illustrate possible approaches on weight selection, we provide three distinct Use Cases with corresponding weight allocations:

- Use Case 1: As an example of this use case, we assume a network such as the one described in [59]. Particularly, this network which includes a centralized node, acting as the “brain” of the SDN and Edge interfaces, and has enough processing resources to support the operation of an IDS with large capabilities. In this case, the Detection Effectiveness & Generalizability Dimension is set as the highest priority, while the rest of them are evaluated with lower priority (and lower weights). For this use case, we define an Effectiveness-heavy weight profile.
- Use Case 2: As an example of this use case, we assume an IDS such the one that is described at [70]. For the development of this system the Dataset Quality & Attack Coverage metrics have achieved the highest score. A stakeholder that prioritizes this characteristic for its environment—with a compromise on detection performance—as a trade-off, can choose the Coverage-heavy weight profile.
- Use Case 3: As an example of this use case, we assume a network with resource-constrained devices [66]. In this case, the Practical Deployability of the IDS takes the highest priority, because the optimization of energy consumption and computational power are a critical factor for the operational use of the network. By compromising detection performance and dataset quality, we define a Deployment-heavy weight profile.

Although the examples provided make the highest-priority Dimension evident in each Use Case, the allocation of weights across all Dimensions depends on the selected priority determination approach. In these Use Cases, we will assign weights based on the AHP method.

Following the methodology established by Saaty [77], the intensity of importance between pairs of dimensions within the framework was evaluated using a standard 1–9 linear scale. On this scale, a value of 1 signifies equal importance, whereas a value of 9 represents the absolute dominance of one dimension over another. For clarity, the dimensions are designated as follows: D_1 for Detection Effectiveness & Generalizability, D_2 for Dataset Quality & Attack Coverage, and D_3 for Practical Deployability. The pairwise comparison

values were assigned based on expert judgment and empirical knowledge of the operational environment corresponding to each use case.

For Use Case 1, the relative importance relationships are defined as follows:

- D_1 vs. $D_2 = 3$ (D_1 is moderately more important than D_2);
- D_1 vs. $D_3 = 3$ (D_1 is moderately more important than D_3);
- D_2 vs. $D_3 = 1$ (D_2 and D_3 are equally important).

The resulting reciprocal pairwise comparison matrix A is structured as follows:

$$A = \begin{pmatrix} 1.0000 & 3.0000 & 3.0000 \\ 0.3333 & 1.0000 & 1.0000 \\ 0.3333 & 1.0000 & 1.0000 \end{pmatrix} \quad (17)$$

By calculating the geometric mean (GM) for each row and normalizing the values against the total sum of the GMs, the final criteria weights are derived as: $w_{D1} = 0.6$, $w_{D2} = 0.2$ and $w_{D3} = 0.2$. Given the perfect transitivity of the pairwise judgments, both the Consistency Index (CI) and the Consistency Ratio (CR) yield a value of 0.00, which indicates absolute logical consistency.

For Use Case 2, the specific relative importance relationships for this scenario are defined as follows:

- D_2 vs. $D_1 = 2$ (D_2 is weakly to moderately more important than D_1);
- D_1 vs. $D_3 = 3$ (D_1 is moderately more important than D_3);
- D_2 vs. $D_3 = 6$ (D_2 is strongly to strongly plus more important than D_3).

The resulting reciprocal pairwise comparison matrix A is structured as follows:

$$A = \begin{pmatrix} 1.0000 & 0.5000 & 3.0000 \\ 2.0000 & 1.0000 & 6.0000 \\ 0.3333 & 0.1667 & 1.0000 \end{pmatrix} \quad (18)$$

By calculating the geometric mean (GM) for each row and normalizing the values against the total sum of the GMs, the final criteria weights are derived as: $w_{D1} = 0.3$, $w_{D2} = 0.6$ and $w_{D3} = 0.1$. Because the mathematical layout perfectly satisfies transitive logic, both the Consistency Index (CI) and the Consistency Ratio (CR) yield a value of 0.00, confirming absolute logical consistency within the evaluation matrix.

For Use Case 3, the relative importance relationships for this configuration are defined as follows:

- D_1 vs. $D_2 = 3$ (D_1 is moderately more important than D_2);
- D_3 vs. $D_1 = 2$ (D_3 is weakly to moderately more important than D_1);
- D_3 vs. $D_2 = 6$ (D_3 is strongly to strongly plus more important than D_2)

The resulting reciprocal pairwise comparison matrix A is structured as follows:

$$A = \begin{pmatrix} 1.0000 & 3.0000 & 0.5000 \\ 0.3333 & 1.0000 & 0.1667 \\ 2.0000 & 6.0000 & 1.0000 \end{pmatrix} \quad (19)$$

By calculating the geometric mean (GM) for each row and normalizing the values against the total sum of the GMs, the final criteria weights are derived as: $w_{D1} = 0.3$, $w_{D2} = 0.1$, and $w_{D3} = 0.6$. Because the mathematical judgments perfectly maintain transitive alignment, both the Consistency Index (CI) and the Consistency Ratio (CR) yield a value of 0.00, indicating absolute logical consistency.

Table 11. Scoring of IDSs across the evaluation framework metrics.

Dimension	Subcategory	Metrics	[59]	[4]	[65]	[67]	[62]	[68]	[60]	[69]	[70]	[71]	[61]	[72]	[73]	[66]	[64]	[63]
DETECTION EFFECTIVENESS & GENERALIZ- ABILITY	Core Metrics	Accuracy	0.9328	0	0	0.9987	0.9866	0.9947	0.9944	0	0.9960	0.9597	0	0.9498	0	0.9959	0	0.9932
		Recall	0.9368	1	0	0	0.9876	0	0.9806	0.9830	0.9930	0	0	0	0.9711	0.9829	0	0.9938
		Precision	0.9328	0	0	0	0.9884	0	0.9841	0.9920	0.9980	0	0	0	0	0	0	0.9923
		F1-score	0.9327	0	0	0	0.9816	0.9947	0.9824	0	0.9960	0.9599	0	0	0	0	0	0.9931
	Advanced Metrics	Adaptability Score	0	0	0	0	0	0	0.9390	0.9951	0	0	0	0	0.9464	0.9907	0	0
		Novelty Detection Rate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DATASET QUALITY & ATTACK COVERAGE		Attack Taxonomy Coverage Index	0.1429	0	0	0.2857	0.5714	0.2857	0.4286	0.1429	0.4286	0.1429	0.1429	0.1429	0.1429	0.4286	0.1429	0.5714
		Dataset Diversity Requirement	0.7500	0	0.7500	0.7500	0.5000	0.7500	0.2500	0.5000	1	0.2500	0.5000	0.2500	0.5000	0.7500	0	0.5000
		Dataset Balance Index	0.6649	0	0	0	0.9995	1	0.9965	0.9086	1	0	0	0	0	0.7983	1	0.9995
		Layer Fusion Indicator	1	0.3750	0.8750	1	1	0.3750	0.3750	0.5000	1	1	1	0.2500	0.3750	1	0.3750	1
PRACTICAL DEPLOYABILITY		Training Time Index	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Detection Time Index	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Memory Footprint Index	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Model Size Index	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8970	0	0
		Expanded Energy Index	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

After computing all available metric values and applying the aforementioned Use Case weights, each IDS received a score per Use Case, as shown in Table 12. It is noted that for each IDS, we extracted the peak performance metrics reported. This approach ensures a fair representation of each methodology's potential. What stands out is how the rankings and final scores shift depending on the Use Case.

This scoring and ranking variation is illustrated by the work of [66]. In Use Case 3, it achieves a score of 0.3305 and takes first place. However, in Use Cases 1 and 2, its scores are 0.4817 and 0.6129, which place it sixth and fourth, respectively.

Figures 3–5 gives a complete picture of the rankings and how each IDS shifts across the three use cases. Specifically, they present the scores in descending order for the aforementioned Use Cases. This descending arrangement makes it easy to follow how each IDS moves up or down from one Use Case to another. From these Figures, we can draw the following conclusions:

- The way scores and rankings shift across the various Use Cases confirms the value of weight-based scoring in differentiating the priorities set for each Use Case.
- The notably lower scores in the Use Case 3 (Deployment-heavy scenario) are explained by the fact that most metrics in Dimension 3 (Practical Deployability) are novel and, as a result, do not appear in existing studies, reinforcing the need for practical evaluation tied to the deployment environment in which these systems operate.

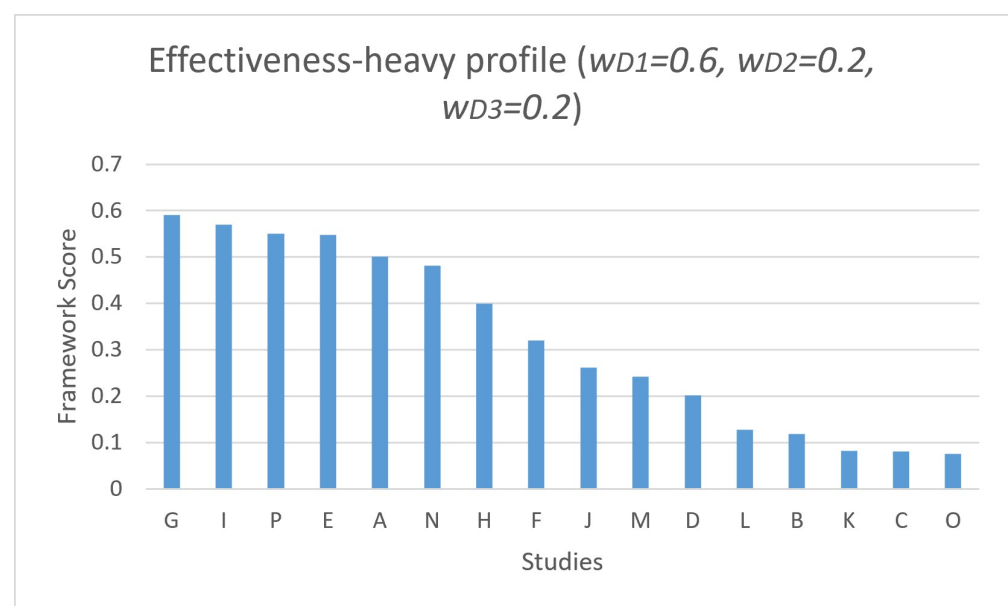


Figure 3. Use Case 1 ranking profile.

Table 12. Framework scoring evaluations across different Use Case.

Paper (Figure Index)	Use Case 1	Use Case 2	Use Case 3
[59] (A)	0.5014	0.5704	0.2507
[4] (B)	0.1188	0.1063	0.0594
[65] (C)	0.0813	0.2438	0.0406
[67] (D)	0.2017	0.3553	0.1008
[62] (E)	0.5480	0.6578	0.2740
[68] (F)	0.3195	0.4611	0.1597
[60] (G)	0.5906	0.5515	0.2953

Table 12. Cont.

Paper (Figure Index)	Use Case 1	Use Case 2	Use Case 3
[69] (H)	0.3996	0.4562	0.1998
[70] (I)	0.5697	0.7134	0.2849
[71] (J)	0.2616	0.3049	0.1308
[61] (K)	0.0821	0.2464	0.0411
[72] (L)	0.1271	0.1439	0.0636
[73] (M)	0.2426	0.2486	0.1213
[66] (N)	0.4817	0.6129	0.3305
[64] (O)	0.0759	0.2277	0.0379
[63] (P)	0.5508	0.6593	0.2754

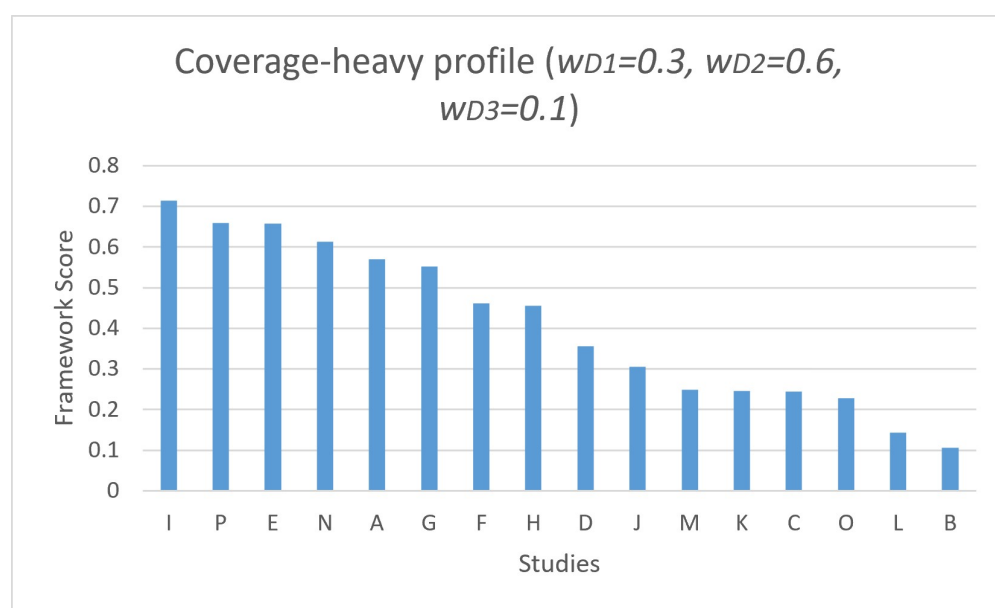


Figure 4. Use Case 2 ranking profile.

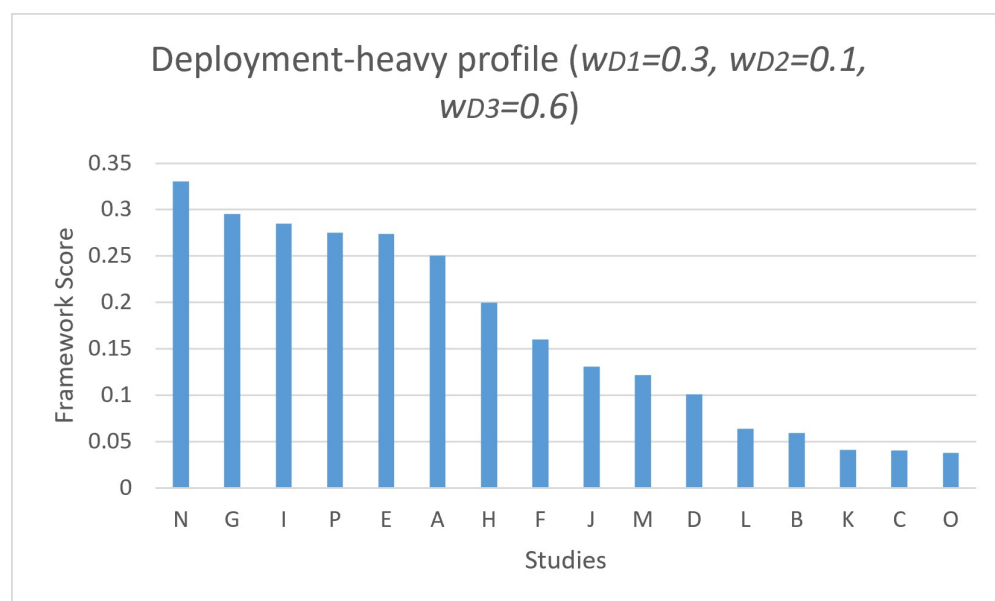


Figure 5. Use Case 3 ranking profile.

5.2. Sensitivity Analysis

5.2.1. Weight Calibration

While the weights assigned to each Dimension through AHP process reflect a reasonable design to prioritize specific operational criteria, any subjective weight imputation introduces an inherent layer of uncertainty. To ensure that the final rankings of the evaluated IDSs do not merely depend on these specific numbers, a rigorous sensitivity analysis was conducted. The objective of this evaluation is to test the structural stability of the model against realistic calibration margins of error.

To accomplish this, we utilized a localized fixed-percentage perturbation method [80]. By focusing the analysis directly around the intentional “anchor” values, we can simulate minor real-world variations or subjective estimation errors. The primary, heavily-weighted dimension in each Use Case was independently varied across a controlled interval of $\pm 5\%$ and $\pm 10\%$. To maintain strict multi-criteria normalization, any change applied to the target anchor weight was dynamically absorbed by the remaining two criteria in exact proportion to their original baseline ratios, satisfying the fundamental constraint that $\sum w_i = 1.0$. To ground the analysis, the highest-priority Dimension within each use case was selected as an empirical anchor, given its dominant influence on the overall scoring trends.

The specific weight distributions resulting from this systematic perturbation method are structured across three distinct weight configurations. Table 13 provides a complete breakdown of these experimental shifts. Under the weight profile for Use Case 1 (Table 13a), the primary emphasis on Detection Effectiveness & Generalizability ($w_{D1} = 0.60$) is scaled from 0.5400 up to 0.6600, with the supporting criteria splitting the remaining weight evenly.

In the weight profile for Use Case 2 (Table 13b) and Use Case 3 (Table 13c), the localized shifts target towards Dataset Quality & Attack Coverage ($w_{D2} = 0.60$) and Practical Deployability ($w_{D3} = 0.60$), respectively. Because their supporting criteria are asymmetrical (0.30 and 0.10 in the baseline), the dynamic scaling preserves a strict 3:1 ratio during reallocation. For example, in Use Case 3 pushing the Practical Deployability anchor up by 10% to 0.6600 causes Detection Effectiveness & Generalizability to scale down to 0.2550 and Dataset Quality & Attack Coverage to drop to 0.0850, keeping their relative proportions intact.

Table 13. Weight configuration profiles under local variations ($\pm 5\%$ and $\pm 10\%$) for all three Use Cases. (a) Weight configurations for Use Case 1. (b) Weight configurations for Use Case 2. (c) Weight configurations for Use Case 3.

(a)				(b)			
Variation	w_{D1}	w_{D2}	w_{D3}	Variation	w_{D1}	w_{D2}	w_{D3}
−10%	0.5400	0.2300	0.2300	−10%	0.3450	0.5400	0.1150
−5%	0.5700	0.2150	0.2150	−5%	0.3225	0.5700	0.1075
+5%	0.6300	0.1850	0.1850	+5%	0.2775	0.6300	0.0925
+10%	0.6600	0.1700	0.1700	+10%	0.2550	0.6600	0.0850
(c)							
	Variation	w_{D1}	w_{D2}	w_{D3}			
	−10%	0.3450	0.1150	0.5400			
	−5%	0.3225	0.1075	0.5700			
	+5%	0.2775	0.0925	0.6300			
	+10%	0.2550	0.0850	0.6600			

The percentage and ranking changes relatively to the baseline weight configurations of all 16 evaluated IDSs under these variations are comprehensively mapped in Tables 14–16. An examination of these metrics yields a highly compelling finding regarding

the core integrity of the evaluation model, particularly the scoring framework demonstrates remarkable robustness, displaying an almost entirely unaltered ranking across the tested intervals.

For both the Use Cases 1 and 3 matrices, the ranking is identical to the baseline weights. Despite individual paper scores shifting by up to $\pm 15.00\%$, these fluctuations are not enough to cause a rank reversal.

The sole weight profile where localized rank reversals manifest is within the Use Case 2 (Table 15). Far from indicating a weakness in the model, this localized volatility reveals that the scores of these IDSs are so densely clustered that even minor weight adjustments are enough to alter their relative positions.

This behavior confirms that the framework possesses excellent discriminatory power, because it remains sensitive where IDSs scores are dense. This proves that the final evaluations are stable and entirely decoupled from minor expert calibration variances.

Table 14. Sensitivity analysis results for Use Case 1 under local weight perturbations.

Paper	−10%		−5%		+5%		+10%	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	−3.62%	0	−1.81%	0	1.81%	0	3.62%	0
[4]	−6.05%	0	−3.03%	0	3.03%	0	6.05%	0
[65]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[67]	2.62%	0	1.31%	0	−1.31%	0	−2.62%	0
[62]	−2.99%	0	−1.50%	0	1.50%	0	2.99%	0
[68]	−0.57%	0	−0.28%	0	0.28%	0	0.57%	0
[60]	−5.66%	0	−2.83%	0	2.83%	0	5.66%	0
[69]	−3.58%	0	−1.79%	0	1.79%	0	3.58%	0
[70]	−2.48%	0	−1.24%	0	1.24%	0	2.48%	0
[71]	−3.34%	0	−1.67%	0	1.67%	0	3.34%	0
[61]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[72]	−3.68%	0	−1.84%	0	1.84%	0	3.68%	0
[73]	−4.76%	0	−2.38%	0	2.38%	0	4.76%	0
[66]	−0.41%	0	−0.21%	0	0.21%	0	0.41%	0
[64]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[63]	−3.03%	0	−1.52%	0	1.52%	0	3.03%	0

Table 15. Sensitivity analysis results for Use Case 2 under local weight perturbations.

Paper	−10%		−5%		+5%		+10%	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	−1.82%	0	−0.91%	0	0.91%	0	1.82%	0
[4]	1.76%	0	0.88%	0	−0.88%	0	−1.76%	0
[65]	−10.00%	0	−5.00%	0	5.00%	+1	10.00%	+1
[67]	−6.49%	0	−3.24%	0	3.24%	0	6.49%	0
[62]	−2.51%	0	−1.25%	0	1.25%	0	2.51%	0
[68]	−4.61%	−1	−2.30%	−1	2.30%	0	4.61%	0
[60]	1.06%	0	0.53%	0	−0.53%	0	−1.06%	0
[69]	−1.86%	+1	−0.93%	+1	0.93%	0	1.86%	0
[70]	−3.02%	0	−1.51%	0	1.51%	0	3.02%	0
[71]	−2.13%	0	−1.07%	0	1.07%	0	2.13%	0
[61]	−10.00%	0	−5.00%	0	5.00%	+1	10.00%	+1
[72]	−1.75%	0	−0.88%	0	0.88%	0	1.75%	0
[73]	−0.36%	0	−0.18%	0	0.18%	−2	0.36%	−3
[66]	−3.21%	0	−1.61%	0	1.61%	0	3.21%	0
[64]	−10.00%	0	−5.00%	0	5.00%	0	10.00%	+1
[63]	−2.47%	0	−1.23%	0	1.23%	0	2.47%	0

Table 16. Sensitivity analysis results for Use Case 3 under local weight perturbations.

Paper	−10%		−5%		+5%		+10%	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[4]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[65]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[67]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[62]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[68]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[60]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[69]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[70]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[71]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[61]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[72]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[73]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[66]	6.86%	0	3.43%	0	−3.43%	0	−6.86%	0
[64]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0
[63]	15.00%	0	7.50%	0	−7.50%	0	−15.00%	0

5.2.2. Handling Missing Values

To explore the structural impact of missing values on our evaluation framework proof of concept, a comprehensive missing-value sensitivity analysis was performed [81]. When dealing with complex systems like Intrusion Detection Systems (IDS), multi-criteria evaluation frameworks often encounter incomplete data. In this study, the primary driver for these missing entries is the integration of novel evaluation metrics. Since these parameters did not exist when the examined IDSs were designed and documented, literature naturally lacks the data required to populate every metric uniformly.

To evaluate how our model addresses these gaps without distorting the final rankings, we compared the baseline scenario, where missing values are treated as 0, to three distinct statistical perturbation tracks:

- **Minimum Imputation:** Gaps are filled with the lowest reported value for that specific metric row. This serves as a cautious, middle-ground buffer that avoids a zero-penalty but assumes below-average performance.
- **Average and Median Imputations:** Gaps are supplemented with the row's observed mean or median, acting under the assumption that missing entries would likely mirror the industry standard of their peers.

The ranking reactions to these adjustments are compiled in the three evaluation tables (Tables 17–19). These tables capture the percentage shift in final scores (Δ Score) and the corresponding movement in rankings (Δ Pos.) across all 16 evaluated studies for each of the Use Cases.

Analyzing these shifts yields critical insights into the framework's overall robustness. The studies exhibit two distinct behaviors: high stability (robustness) or high volatility (sensitivity). Studies such as [62,63,66,70] show minimal changes in score and completely static ranking positions across all imputation methods. This consistency indicates high data density.

Conversely, studies like [4,61,64,65] display long ranking swings, in some instances climbing up to +9 positions under average imputation or sliding under the baseline. These fluctuations indicate a high concentration of missing values, making their final standing highly sensitive to the chosen imputation strategy.

Table 17. Sensitivity analysis results for Use Case 1 under missing value imputation strategies.

Paper	Minimum		Average		Median	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	25.88%	−2	26.46%	−6	0.00%	0
[4]	389.47%	−2	434.41%	+3	127.28%	+2
[65]	669.14%	+6	720.48%	+9	155.26%	+2
[67]	219.81%	+6	234.60%	+6	9.90%	−1
[62]	23.68%	+1	24.21%	+1	0.00%	0
[68]	99.14%	+2	102.95%	0	0.00%	−1
[60]	6.08%	−7	6.08%	−13	0.00%	0
[69]	55.67%	−3	57.96%	−5	23.89%	+1
[70]	22.78%	+1	23.28%	+1	0.00%	0
[71]	133.78%	−3	143.49%	0	7.63%	−1
[61]	653.17%	+3	695.26%	+7	140.53%	0
[72]	348.68%	−4	372.15%	−4	15.70%	−4
[73]	143.81%	−4	155.05%	−5	47.57%	+2
[66]	38.73%	+2	40.66%	+2	0.00%	−1
[64]	679.63%	+3	728.91%	+3	158.74%	+1
[63]	23.56%	+1	24.09%	+1	0.00%	0

Table 18. Sensitivity analysis results for Use Case 2 under missing value imputation strategies.

Paper	Minimum		Average		Median	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	11.38%	−1	11.63%	−3	0.00%	0
[4]	342.09%	+1	452.98%	+5	202.12%	+1
[65]	152.95%	+5	182.41%	+7	57.33%	+3
[67]	85.77%	+4	99.29%	+4	16.85%	0
[62]	9.86%	0	10.08%	0	0.00%	0
[68]	34.35%	0	35.66%	−2	0.00%	−1
[60]	3.25%	−4	3.25%	−8	0.00%	0
[69]	24.38%	−3	25.38%	−5	10.46%	+1
[70]	9.10%	0	9.30%	0	0.00%	0
[71]	84.65%	−2	99.67%	0	19.64%	−1
[61]	142.59%	+3	163.04%	+5	43.67%	−1
[72]	211.74%	−1	245.11%	−1	41.60%	−1
[73]	103.64%	−3	122.44%	−4	43.29%	−1
[66]	15.22%	0	15.98%	0	0.00%	0
[64]	127.00%	+1	152.86%	+2	53.91%	0
[63]	9.84%	0	10.06%	0	0.00%	0

This high sensitivity does not necessarily indicate a flaw in the methodology; rather, it demonstrates that the novel metrics have the potential to influence rankings when data is available. If the new parameters lacked influence, altering the missing data strategy would not trigger such a structural realignment of the ranking. However, this also underscores that the current proof-of-concept results should be interpreted with caution, as the observed ranking shifts are partly an artifact of missing data rather than inherent system performance. Furthermore, documenting these wide shifts allows us to quantify the “strategic omission gap.” For instance, if missing metrics were instead assigned a value greater than zero, such as the median performance observed across a representative set of IDSs, it could introduce unintended incentives for selective reporting. Specifically, developers whose IDS achieved a measured value within the interval [0, default missing value) might prefer not to report that metric, as omitting the result would lead to a higher evaluation outcome than disclosing the

actual performance. Assigning the minimum possible score to missing values discourages such behavior and encourages complete reporting across all evaluation metrics.

This conservative strategy may appear to introduce bias against systems that do not report a complete set of metrics. However, it is intended to promote confidence among evaluators who rely on these results to select an IDS for their operational environment by encouraging the reporting of accurate and verifiable performance values. At the same time, it supports transparency across IDS developers and discourages attempts to gain competitive advantage through incomplete reporting or selectively disclosed metric values.

It should be noted that rankings derived from incomplete data are provisional. A system with missing metrics might achieve a higher score if those metrics were measured and reported and the framework is not designed to penalize systems for the novelty of its own metrics.

Table 19. Sensitivity analysis results for Use Case 3 under missing value imputation strategies.

Paper	Minimum		Average		Median	
	Δ Score	Δ Pos.	Δ Score	Δ Pos.	Δ Score	Δ Pos.
[59]	61.66%	−1	62.24%	−5	0.00%	0
[4]	540.55%	−2	585.49%	+3	127.28%	+2
[65]	889.94%	+6	941.28%	+9	155.26%	+2
[67]	308.77%	+6	323.57%	+6	9.90%	−1
[62]	56.42%	+2	56.95%	+2	0.00%	0
[68]	155.30%	+2	159.10%	0	0.00%	−1
[60]	36.45%	−6	36.45%	−12	0.00%	0
[69]	100.56%	−3	102.86%	−5	23.89%	0
[70]	54.27%	+2	54.77%	+2	0.00%	0
[71]	202.36%	−3	212.07%	0	7.63%	−1
[61]	871.57%	+3	913.66%	+7	140.53%	0
[72]	489.81%	−4	513.27%	−4	15.70%	−4
[73]	217.75%	−4	228.99%	−5	47.57%	+2
[66]	28.22%	−3	29.63%	−3	0.00%	0
[64]	916.02%	+3	965.30%	+3	158.74%	+1
[63]	56.13%	+2	56.66%	+2	0.00%	0

6. Conclusions and Future Work

In this study, we began by applying the PRISMA framework to investigate the cross-layer impact of the most severe, state-of-the-art IoT DoS attacks. Our findings show that each of these attacks exerts an influence that traverses across several network layers. Using the same systematic approach, we then identified state-of-the-art cross-layer IDSs that rely on ML techniques for detection. We examined their architectural patterns, the data fusion strategies that give them their cross-layer character, and the ML algorithms used to train their models. At the same time, we surfaced the existing challenges for a comprehensive evaluation.

To address these challenges, we proposed an evaluation framework that assesses an IDS across multiple dimensions, namely, detection performance, generalization ability, training data quality, and practical deployability. As a proof of concept, we applied this framework to the IDSs identified earlier in our workflow. The results illustrate that IDS evaluation remains fragmented across disparate sets of metrics, with detection performance dominating the evaluation efforts. However, due to the retrospective nature of our scoring and the significant number of missing values, particularly for the novel metrics introduced in this framework, these results should be interpreted as illustrative rather than as definitive rankings of the evaluated systems.

Our future work will assess the practical deployability of the ML models and cross-layer datasets most used in the literature for training IDSs. As highlighted throughout this paper, the majority of existing studies emphasize on performance oriented metrics. We intend to shift the research focus towards resource-aware evaluation by benchmarking the best performing ML models alongside the datasets on which they were trained. This benchmarking process will then be combined with the findings that emerge from the proposed evaluation framework. This shift will allow us to classify which solutions are holistically affordable for different classes of IoT devices, given that many IoT networks present diversified requirements.

A key part of this effort will be the computation of metrics that have not been widely available due to their novelty, particularly those belonging to the Practical Deployability dimension. Building on this, future work could extend the validation of the framework by correlating its scores with independent testbed evaluations or by conducting meta-analyses to assess whether high-scoring systems indeed perform and generalize better in practice. Together, these directions will bridge the gap between theoretical evaluation and feasible deployment in real-world IoT environments.

Author Contributions: Conceptualization, D.T. and A.X.; methodology, D.T. and A.L.; validation, D.T., D.K. and C.C.; formal analysis, D.T.; investigation, D.T.; resources, D.T.; writing—original draft preparation, D.T.; writing—review and editing, D.T., A.X., and A.L.; visualization, D.T.; supervision, A.X. and A.L.; project administration, V.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

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

References

1. 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. [[CrossRef](#)]
2. Ioannou, C.; Chrysostomou, C. Can Cross-Layer Intrusion Detection Secure Agriculture 4.0 Systems? In Proceedings of the 2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Lucca, Italy, 9–11 June 2025; pp. 984–990. [[CrossRef](#)]
3. Kharkwal, A.; Mishra, S.; Paul, A. Cross-Layer DoS Attack Detection Technique for Internet of Things. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; pp. 368–372. [[CrossRef](#)]
4. Mofidi, F.; Hounsinnou, S.G.; Bloom, G. CrossSentry: A Cross-Layer Approach to Ransomware Detection in IoT. In Proceedings of the 2025 13th International Symposium on Digital Forensics and Security (ISDFS), Boston, MA, USA, 24–25 April 2025; pp. 1–6. [[CrossRef](#)]
5. Zhang, L.; Restuccia, F.; Melodia, T.; Pudlewski, S.M. Learning to detect and mitigate cross-layer attacks in wireless networks: Framework and applications. In Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 9–11 October 2017; pp. 1–9. [[CrossRef](#)]
6. Tasiopoulos, D.; Xenakis, A.; Lekidis, A.; Kosmanos, D.; Chaikalis, C.; Vlachos, V. Towards DoS Attack Detection for IoT Systems: A Cross-Layer Oriented Approach Based on Machine Learning Techniques. In Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR), Crete, Greece, 4–6 August 2025; pp. 42–49. [[CrossRef](#)]
7. 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. [[CrossRef](#)]
8. Khacha, A.; Aliouat, Z.; Harbi, Y.; Gherbi, C.; Saadouni, R.; Harous, S. Landscape of learning techniques for intrusion detection system in IoT: A systematic literature review. *Comput. Electr. Eng.* **2024**, *120*, 109725. [[CrossRef](#)]

9. Ferrag, M.A.; Shu, L.; Friha, O.; Yang, X. Cyber Security Intrusion Detection for Agriculture 4.0: Machine Learning-Based Solutions, Datasets, and Future Directions. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 407–436. [\[CrossRef\]](#)
10. Bankó, M.B.; Dyszewski, S.; Králová, M.; Limpek, M.B.; Papaioannou, M.; Choudhary, G.; Dragoni, N. Advancements in Machine Learning-Based Intrusion Detection in IoT: Research Trends and Challenges. *Algorithms* **2025**, *18*, 209. [\[CrossRef\]](#)
11. Asati, V.K.; Pilli, E.S.; Vipparthi, S.K.; Garg, S.; Singhal, S.; Pancholi, S. RMDD: Cross Layer Attack in Internet of Things. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 172–178. [\[CrossRef\]](#)
12. Radosavac, S.; Benammar, N.; Baras, J. Cross-layer attacks in wireless ad hoc networks. In Proceedings of the Conference on Information Sciences and Systems, Princeton, NJ, USA, 17–19 March 2004.
13. Wang, A.; Mohaisen, A.; Chen, S. XLF: A Cross-layer Framework to Secure the Internet of Things (IoT). In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 1830–1839. [\[CrossRef\]](#)
14. Saranya, K.; Valarmathi, D.A. A Comparative Study on Machine Learning based Cross Layer Security in Internet of Things (IoT). In Proceedings of the 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 13–15 December 2022; pp. 267–273. [\[CrossRef\]](#)
15. Wang, J.; Fapojuwo, A.O.; Zhang, C.; Tan, H. UML Modeling of Cross-Layer Attack in Wireless Sensor Networks. In *Proceedings of the Interoperability, Safety and Security in IoT*; Mitton, N., Chaouchi, H., Noel, T., Watteyne, T., Gabillon, A., Capolsini, P., Eds.; Springer: Cham, Switzerland, 2017; pp. 104–115.
16. Fang, Z.; Fu, H.; Gu, T.; Qian, Z.; Jaeger, T.; Mohapatra, P. ForeSee: A Cross-Layer Vulnerability Detection Framework for the Internet of Things. In Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Monterey, CA, USA, 4–7 November 2019; pp. 236–244. [\[CrossRef\]](#)
17. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Zahra, B.F.; Abdelhamid, B. Risk analysis in Internet of Things using EBIOS. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–7. [\[CrossRef\]](#)
19. Jimo, S.; Abdullah, T.; Jamal, A. IoE Security Risk Analysis in a Modern Hospital Ecosystem. In *Proceedings of the Cybersecurity in the Age of Smart Societies*; Jahankhani, H., Ed.; Springer: Cham, Switzerland, 2023; pp. 451–467.
20. Plosz, S.; Farshad, A.; Tauber, M.; Lesjak, C.; Rupprechter, T.; Pereira, N. Security vulnerabilities and risks in industrial usage of wireless communication. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; pp. 1–8. [\[CrossRef\]](#)
21. Butun, I.; Pereira, N.; Gidlund, M. Security Risk Analysis of LoRaWAN and Future Directions. *Future Internet* **2019**, *11*, 3. [\[CrossRef\]](#)
22. Cheung, K.L. Enhancing Intrusion Detection Systems with Machine Learning: A Comprehensive Framework for Improved Detection and Interpretability. In Proceedings of the 2025 7th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Hangzhou, China, 24–26 October 2025; pp. 270–274. [\[CrossRef\]](#)
23. Hadri, A.; Arabi, S. Comprehensive Cross-Dataset Evaluation Framework for Machine Learning-Based Intrusion Detection Systems: A Robust Multi-Metric Validation Approach. In Proceedings of the 2025 8th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco, 3–5 December 2025; pp. 1–8. [\[CrossRef\]](#)
24. Gharib, A.; Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. An Evaluation Framework for Intrusion Detection Dataset. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 19–22 December 2016; pp. 1–6. [\[CrossRef\]](#)
25. Praveen, L.; Vatsa, D.; Lalithaditya, P.; Gupta, M. Adaptability Over Accuracy: Strengthening Intrusion Detection Systems for Zero-Day Attacks. In Proceedings of the 2026 International Conference on Intelligent Systems in Engineering, Secured Systems and Cybersecurity (ICISESSC), Greater Noida, India, 9–10 April 2026; pp. 442–446. [\[CrossRef\]](#)
26. Gungor, O.; Li, E.; Shang, Z.; Guo, Y.; Chen, J.; Davis, J.; Rosing, T. Rigorous Evaluation of Machine Learning-Based Intrusion Detection Against Adversarial Attacks. In Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience (CSR), London, UK, 2–4 September 2024; pp. 152–158. [\[CrossRef\]](#)
27. Nguyen, H.C.T.; Nguyen, X.H.; Le, K.H. An Automated Benchmarking Framework for Anomaly-based Intrusion Detection Systems. In Proceedings of the 2024 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Da Nang, Vietnam, 15–16 August 2024; pp. 1–6. [\[CrossRef\]](#)
28. Gite, P.; Chouhan, K.; Murali Krishna, K.; Kumar Nayak, C.; Soni, M.; Shrivastava, A. ML Based Intrusion Detection Scheme for various types of attacks in a WSN using C4.5 and CART classifiers. *Mater. Today Proc.* **2023**, *80*, 3769–3776. [\[CrossRef\]](#)
29. Dener, M.; Okur, C.; Al, S.; Orman, A. WSN-BFSF: A New Data Set for Attacks Detection in Wireless Sensor Networks. *IEEE Internet Things J.* **2024**, *11*, 2109–2125. [\[CrossRef\]](#)

30. Baiad, R.; Alhussein, O.; Otrouk, H.; Muhaidat, S. Novel cross layer detection schemes to detect blackhole attack against QoS-OLSR protocol in VANET. *Veh. Commun.* **2016**, *5*, 9–17. [\[CrossRef\]](#)
31. Umar, I.A.; Hanapi, Z.M.; Sali, A.; Zulkarnain, Z.A. TruFiX: A Configurable Trust-Based Cross-Layer Protocol for Wireless Sensor Networks. *IEEE Access* **2017**, *5*, 2550–2562. [\[CrossRef\]](#)
32. Devaraju, B.M.; Raju, G.T. Cross Layer and Management Plane Integration Approach for Detection and Prevention of Malicious Activities in WSN. In Proceedings of the 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 14–15 December 2018; pp. 1831–1838. [\[CrossRef\]](#)
33. Liu, Y.; Tsang, K.F.; Wu, C.K.; Wei, Y.; Wang, H.; Zhu, H. IEEE P2668-Compliant Multi-Layer IoT-DDoS Defense System Using Deep Reinforcement Learning. *IEEE Trans. Consum. Electron.* **2023**, *69*, 49–64. [\[CrossRef\]](#)
34. Wei, Y.; Jang-Jaccard, J.; Sabrina, F.; Singh, A.; Xu, W.; Camtepe, S. AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification. *IEEE Access* **2021**, *9*, 146810–146821. [\[CrossRef\]](#)
35. Praseed, A.; Thilagam, P.S. HTTP request pattern based signatures for early application layer DDoS detection: A firewall agnostic approach. *J. Inf. Secur. Appl.* **2022**, *65*, 103090. [\[CrossRef\]](#)
36. Kalambe, D.; Sharma, D.; Kadam, P.; Surati, S. A comprehensive plane-wise review of DDoS attacks in SDN: Leveraging detection and mitigation through machine learning and deep learning. *J. Netw. Comput. Appl.* **2025**, *235*, 104081. [\[CrossRef\]](#)
37. Aditya Sai Srinivas, T.; Manivannan, S. Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm. *Comput. Commun.* **2020**, *163*, 162–175. [\[CrossRef\]](#)
38. Li, D.C.; Tu, H.H.; Chou, L.D. Cross-layer detection and defence mechanism against DDoS and DRDoS attacks in software-defined networks using P4 switches. *Comput. Electr. Eng.* **2024**, *118*, 109307. [\[CrossRef\]](#)
39. Al-Anzi, F.S. Design and analysis of intrusion detection systems for wireless mesh networks. *Digit. Commun. Netw.* **2022**, *8*, 1068–1076. [\[CrossRef\]](#)
40. Cakir, S.; Toklu, S.; Yalcin, N. RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning. *IEEE Access* **2020**, *8*, 183678–183689. [\[CrossRef\]](#)
41. Gunduz, M.Z.; Das, R. Cyber-security on smart grid: Threats and potential solutions. *Comput. Netw.* **2020**, *169*, 107094. [\[CrossRef\]](#)
42. Hu, C.; Zhang, R.; Li, B.; Jiang, X.; Zhao, N.; Di Renzo, M.; Niyato, D.; Nallanathan, A.; Karagiannis, G.K. Generative AI-Empowered Secure Communications in Space–Air–Ground Integrated Networks: A Survey and Tutorial. *IEEE Commun. Surv. Tutor.* **2026**, *28*, 4156–4194. [\[CrossRef\]](#)
43. López, M.; Peinado, A.; Ortiz, A. An extensive validation of a SIR epidemic model to study the propagation of jamming attacks against IoT wireless networks. *Comput. Netw.* **2019**, *165*, 106945. [\[CrossRef\]](#)
44. Lall, S.; Maharaj, B.; van Vuuren, P.J. Null-frequency jamming of a proactive routing protocol in wireless mesh networks. *J. Netw. Comput. Appl.* **2016**, *61*, 133–141. [\[CrossRef\]](#)
45. Alaimo, R.; Tinnirello, I.; Garlisi, D. Exploiting DCI Leakage: A Stealthy 5G Uplink Jamming Attack Using Compromised UE. In Proceedings of the 2025 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Madison, WI, USA, 4–6 August 2025; pp. 1–6. [\[CrossRef\]](#)
46. Ma, Y.; Wu, Y. The DPC-LSTM-MD scheme for detecting selective forwarding attack under variable environment in event-driven wireless sensor networks. *J. Netw. Comput. Appl.* **2026**, *247*, 104411. [\[CrossRef\]](#)
47. Sinha, S.; B, S. Impact of DoS attack in IoT system and identifying the attacker location for interference attacks. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 8–10 July 2021; pp. 657–662. [\[CrossRef\]](#)
48. Wang, R.; Zhang, Z.; Zhang, Z.; Jia, Z. ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs. *Comput. Netw.* **2018**, *139*, 119–135. [\[CrossRef\]](#)
49. Kaur, A.; Saiyed, M.F.; Al-Anbagi, I.; Shamim Hossain, M. A QoS-aware adaptive cross-layer cyber-attack detection algorithm for vehicular networks. *Veh. Commun.* **2025**, *54*, 100929. [\[CrossRef\]](#)
50. Al-chikh Omar, A.A.R.; Soudan, B.; Ala' Altaweel. A comprehensive survey on detection of sinkhole attack in routing over low power and Lossy network for internet of things. *Internet Things* **2023**, *22*, 100750. [\[CrossRef\]](#)
51. Prathapchandran, K.; Janani, T. A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest – RFTRUST. *Comput. Netw.* **2021**, *198*, 108413. [\[CrossRef\]](#)
52. Yang, Z.; Li, L.; Gu, F.; Ling, X.; Hajjee, M. TADR-EAODV: A trust-aware dynamic routing algorithm based on extended AODV protocol for secure communications in wireless sensor networks. *Internet Things* **2022**, *20*, 100627. [\[CrossRef\]](#)
53. Nagarajan, G.; Simpson, S.V.; Minu, R. Chapter Thirteen—Edge computing security: Layered classification of attacks and possible countermeasures. In *Edge/Fog Computing Paradigm: The Concept Platforms and Applications*; Raj, P., Saini, K., Surianarayanan, C., Eds.; Advances in Computers; Elsevier: Amsterdam, The Netherlands, 2022; Volume 127, pp. 359–377. [\[CrossRef\]](#)
54. Toe, E.; Jaafar, F.; Charles André Ferrier, L. Securing LoRaWAN in the AIoT Era: A Systematic Mapping Study and an MITRE-Based Threat Matrix. *IEEE Internet Things J.* **2026**, *13*, 3827–3849. [\[CrossRef\]](#)

55. She, D.; Chen, Z.; Chang, M.; Feng, B.; Bai, J. Resilience Evaluation and Recovery Strategies for IoT-Enabled Unmanned System-of-Systems under GNSS Spoofing Attacks. *IEEE Internet Things J.* **2026**, *13*, 29708–29722. [\[CrossRef\]](#)
56. Kumar, M. Oversee effective DDoS detection based on dual frameworks. *Comput. Netw.* **2026**, *275*, 111863. [\[CrossRef\]](#)
57. Singh, J.; Singh, A.; Manchanda, K.; Kaur, A.; Kaur, K.; Sethi, R. Secure Framework for Privacy and Risk Mitigation in IoT-Enabled Health Insurance. In Proceedings of the 2025 2nd International Conference on Integration of Computational Intelligent System (ICICIS), Lohegaon, India, 9–12 September 2025; pp. 1–6. [\[CrossRef\]](#)
58. Arvind, S.; Narayanan, V.A. An Overview of Security in CoAP: Attack and Analysis. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 655–660. [\[CrossRef\]](#)
59. Lakshan Yasarathna, T.; Le-Khac, N.A. ASEADOS-SDN-IoT: A novel SDN-IoT network intrusion detection dataset and framework. *Internet Things* **2026**, *36*, 101891. [\[CrossRef\]](#)
60. Manokaran, J.; Vairavel, G. IGWO-SoE: Improved Grey Wolf Optimization Based Stack of Ensemble Learning Algorithm for Anomaly Detection in Internet of Things Edge Computing. *IEEE Access* **2023**, *11*, 106934–106953. [\[CrossRef\]](#)
61. Srinivasan, J. Innovative cross-layer defense mechanisms for blackhole and wormhole attacks in wireless ad-hoc networks. *Sci. Rep.* **2025**, *15*, 14747. [\[CrossRef\]](#) [\[PubMed\]](#)
62. Kamaldeep; Malik, M.; Dutta, M. Feature Engineering and Machine Learning Framework for DDoS Attack Detection in the Standardized Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 8658–8669. [\[CrossRef\]](#)
63. Kamaldeep; Malik, M.; Dutta, M.; Granjal, J. IoT-Sentry: A Cross-Layer-Based Intrusion Detection System in Standardized Internet of Things. *IEEE Sens. J.* **2021**, *21*, 28066–28076. [\[CrossRef\]](#)
64. Amouri, A.; Alaparthi, V.T.; Morgera, S.D. Cross layer-based intrusion detection based on network behavior for IoT. In Proceedings of the 2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON), Sand Key, FL, USA, 9–10 April 2018; pp. 1–4. [\[CrossRef\]](#)
65. Singhatia, S.; Dhingra, M. Developing an Advanced Cross layer Distributed Model for Enhanced Attack Detection Model. In Proceedings of the 2025 2nd International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT), Warangal, India, 30–31 October 2025; pp. 1–6. [\[CrossRef\]](#)
66. Azzaoui, H.; Boukhamla, A.Z.E.; Perazzo, P.; Alazab, M.; Ravi, V. A Lightweight Cooperative Intrusion Detection System for RPL-based IoT. *Wirel. Pers. Commun.* **2024**, *134*, 2235–2258. [\[CrossRef\]](#)
67. Al Sukhni, B.; Manna, S.K.; Dave, J.M.; Zhang, L. Exploring Optimal Set of Features in Machine Learning for Improving IoT Multilayer Security. In Proceedings of the 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), Aveiro, Portugal, 12–27 October 2023; pp. 1–6. [\[CrossRef\]](#)
68. Panda, N.; Muthuraman, S. A Multiclass Machine Learning Framework for Detecting Routing Attacks in RPL-Based IoT Networks Using a Novel Simulation-Driven Dataset. *Future Internet* **2026**, *18*, 35. [\[CrossRef\]](#)
69. Hafsa, N.; Alzoubi, H.; Imran, S. Machine learning-based intrusion detection and prevention using cross-layer features in Internet of Things (IoT) networks. *J. Commun. Netw.* **2025**, *27*, 345–358. [\[CrossRef\]](#)
70. Shwayat, D.; Abu Al-Haija, Q.; Alma'aitah, A. Resource-aware ML framework for multi-level cross-layer and cross-protocol attack detection in IoMT. *J. Supercomput.* **2026**, *82*, 291. [\[CrossRef\]](#)
71. Paul, A.; Sinha, S.; Mishra, S. A Hybrid Intrusion Detection System for detecting Cross-layer DoS attacks in IoT. *Wirel. Pers. Commun.* **2025**, *145*, 717–740. [\[CrossRef\]](#)
72. Bajaj, P.; Mishra, S.; Paul, A. Comparative Analysis of Stack-Ensemble-Based Intrusion Detection System for Single-Layer and Cross-layer DoS Attack Detection in IoT. *SN Comput. Sci.* **2023**, *4*, 562. [\[CrossRef\]](#)
73. Canbalaban, E.; Sen, S. A Cross-Layer Intrusion Detection System for RPL-Based Internet of Things. In *Proceedings of the Ad-Hoc, Mobile, and Wireless Networks*; Grieco, L.A., Boggia, G., Piro, G., Jararweh, Y., Campolo, C., Eds.; Springer: Cham, Switzerland, 2020; pp. 214–227.
74. Ren, J.; Liu, Y.; Liu, J. EWGAN: Entropy-Based Wasserstein GAN for Imbalanced Learning. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 10011–10012. [\[CrossRef\]](#)
75. Gong, J.; Dong, M. Entropy-based dynamic ensemble classification algorithm for imbalanced data stream with concept drift. *PLoS ONE* **2024**, *19*, e0311133. [\[CrossRef\]](#) [\[PubMed\]](#)
76. Jin, Z.; Hu, T.; Jie, Z.; Wang, P. Research and Implementation of Performance Optimization Methods for RISC-V Level-5 Processors. *Appl. Sci.* **2025**, *15*, 11634. [\[CrossRef\]](#)
77. Saaty, T.L. A scaling method for priorities in hierarchical structures. *J. Math. Psychol.* **1977**, *15*, 234–281. [\[CrossRef\]](#)
78. Li, T.; Zhang, Y. Adaptive Trust Evaluation Model Based on Entropy Weight Method for Sensing Terminal Process. *Entropy* **2025**, *27*, 200. [\[CrossRef\]](#) [\[PubMed\]](#)
79. Ayan, B.; Abacıoğlu, S.; Basilio, M.P. A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision-Making. *Information* **2023**, *14*, 285. [\[CrossRef\]](#)

80. Canepa, M. XAI-MCDA-HoDEM: An Explainable Multi-Criteria Decision Framework for Maritime and Port Decarbonization. *Gases* **2026**, *6*, 25. [[CrossRef](#)]
81. Zhang, X.; Sun, X.; Xia, L.; Tao, S.; Xiang, S. A Matrix Completion Method for Imputing Missing Values of Process Data. *Processes* **2024**, *12*, 659. [[CrossRef](#)]

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.