

Paper Type: Original Article

Machine Learning–Driven Intrusion Detection System for Industrial IoT Networks Using Cross-Layer Features

Raad A. Qasim^{1,*} 

¹ Department of Computer Science, Faculty of Education, University of Telafer, Nineveh, Iraq; raad.a.qasim@uotelafer.edu.iq.

Received: 05 Sep 2025

Revised: 10 Nov 2025

Accepted: 24 Nov 2025

Published: 24 Dec 2025

Abstract

Industrial Internet of Things (IIoT) networks increasingly rely on low-power wireless stacks (e.g., IEEE 802.15.4/TSCH, 6LoWPAN, RPL) to connect sensors, actuators, and gateways in safety- and mission-critical environments. These networks remain vulnerable to multi-layer attacks such as control-plane manipulation (e.g., RPL rank/version attacks), resource depletion (e.g., flooding), and selective forwarding. Many existing intrusion detection systems (IDS) observe only a single layer, which can lead to high false positives under interference, load variations, and topology changes. This paper proposes a machine learning–driven IDS that uses cross-layer features spanning PHY, MAC, network, and transport/application indicators. The proposed system builds per-node, per-window feature vectors that capture both direct anomalies (e.g., sudden DIO spikes) and cross-layer inconsistencies (e.g., high retransmissions with stable link quality). We present a complete design: feature model, windowing and labeling strategy, training pipeline with scenario-wise splits to reduce leakage, lightweight online inference for constrained devices, and an evaluation protocol over diverse network conditions. We provide reproducible pseudocode and reference implementation guidance. Experimental results should be reported using F1, PR-AUC, ROC-AUC, false-positive rate per hour, detection latency, and energy/CPU overhead. Placeholders are included for measured outcomes and tables/figures.

Keywords: Industrial IoT, Intrusion Detection, Cross-Layer Features, RPL, 6LoWPAN, IEEE 802.15.4, Machine Learning, Edge Security.

1 | Introduction

Industrial Internet of Things (IIoT) networks are now widely deployed in factories, power systems, pipelines, and building automation, where connectivity enables fine-grained monitoring and control but also expands the attack surface. In practice, IIoT often relies on low-power and lossy networking stacks and resource-constrained nodes, which makes applying heavyweight security controls difficult and increases exposure to denial-of-service, spoofing, and stealthy manipulation that can degrade availability and safety. Therefore, intrusion detection systems (IDS) form an indispensable last line of defense, as they can continue monitoring behavior even when preventive controls are incomplete or new, previously unknown attacks are launched. In recent years, IoT and IIoT IDS research has increasingly resorted to machine learning (ML) and deep learning (DL) for identifying complex patterns in network traffic and device behavior, improving detection accuracy, and adapting to emerging threats more effectively than traditional static, rule-based approaches [1-4],

 Corresponding Author: raad.a.qasim@uotelafer.edu.iq

 <https://doi.org/10.61356/j.nois.2025.8640>

 Licensee **Neutrosophic Optimization and Intelligent Systems**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

however, significant challenges remain in practical IIoT environments. One common issue is that IDS models trained on single-layer or narrow feature sets will misinterpret benign disturbances (e.g., interference, congestion, or topology changes) as malicious behavior, resulting in higher false positives and lower alert credibility, and in industrial settings, this is particularly problematic since unnecessary mitigation actions (e.g., isolating nodes or rate-limiting traffic) can be expensive and disruptive. Consequently, recent work aims to develop more robust and practical IoT IDS through feature optimization, explainability, and lightweight deployment [1, 5–6], for example, explainable AI (XAI) techniques such as SHAP and LIME have been applied to make IDS decisions more transparent to analysts, thus facilitating debugging and improving operational acceptance [1]. At the same time, systematic reviews emphasize that feature selection and data balancing are crucial to building effective IoT IDS under class imbalance and heterogeneous traffic conditions, and they highlight persisting gaps in generalizability and evaluation rigor across datasets and scenarios [3, 7]. Another important direction is to design architectures that trade strong detection performance with resource constraints, such as multi-level or hybrid IDS, which employ lightweight models on the edge and escalate uncertain or high-risk cases to more powerful analyzers in the cloud or at centralized servers [6].

Likewise, contemporary work continues to explore stronger DL pipelines (CNN/RNN/attention/ensembles) and demonstrates very high accuracy on popular IoT intrusion datasets, yet such results often depend on dataset properties and may not fully reflect the variability of real industrial deployments [2, 4, 5]. These observations motivate ML-driven IDS designs that are both (i) aware of the multi-layer nature of IIoT protocol stacks and (ii) evaluated with procedures that reduce overfitting and improve transfer to unseen conditions.

In this context, this paper proposes a machine learning–driven intrusion detection system for IIoT that relies on cross-layer features spanning PHY, MAC, network (including routing control behavior), and transport/application indicators. The core idea is that many attacks manifest as correlated effects across layers (e.g., control-plane anomalies coupled with retransmission changes and delivery degradation), and explicitly modeling these correlations can improve discrimination between benign dynamics and malicious behavior. By constructing per-node, windowed feature vectors and training ML classifiers with leakage-aware evaluation, the proposed approach targets improved detection accuracy while reducing false positives and maintaining feasibility for deployment at gateways or capable edge nodes.

2 | System Model and Threat Model

2.1 | Network Stack and Observation Points

Industrial IoT (IIoT) deployments commonly rely on Low-Power and Lossy Network (LLN) stacks, where constrained devices communicate over low-rate wireless links and lightweight IP-based routing. In this work, we consider a representative LLN stack built on IEEE 802.15.4 at the PHY/MAC layers (optionally time-scheduled via TSCH), with IPv6 connectivity and routing mechanisms typical of IoT networking, and transport-layer support for application telemetry [11–13]. To improve robustness under interference and operational load variations, recent IDS literature increasingly recommends multi-layer and edge-aware designs that fuse heterogeneous indicators (device/network statistics, traffic patterns, and platform constraints) rather than relying on a single layer [14–16]. Accordingly, our cross-layer feature extraction draws from multiple observation points: radio indicators capturing link quality and errors, MAC indicators reflecting contention, retransmissions and buffering stress, network/routing control-plane indicators representing topology dynamics, and (where measurable) flow-level metrics capturing service impact, which aligns with contemporary IoT IDS survey findings and design guidance [14, 15].

2.2 | Threats Considered

Recent IoT/IIoT IDS research highlights that attacks in constrained environments often manifest as correlated symptoms across layers, motivating multi-level detection and edge-enabled monitoring for timely response [14, 16, 17]. Our proposed IDS targets attacks that exploit protocol semantics and resource constraints, such as routing and control plane manipulation, resource exhaustion via flooding, and selective forwarding that may not be easily detectable from individual layers [14, 15], and we assume that attackers are either internal (compromised nodes) or external, capable of injecting malicious traffic. During runtime, the IDS only requires local counters and protocol statistics, and only requires global information during offline testing, because this is typical of existing IoT IDS studies and surveys [14, 15]. Our approach complements rather than replaces cryptographic techniques, identifying anomalies and cross-layer attacks that may occur even with strong authentication and encryption [16, 18], thus our approach is necessary for a comprehensive security solution.

3 | Related Work

Recent works on IIoT IDS have focused on machine learning (ML) and deep learning (DL) due to the heterogeneous nature of IIoT environments, dynamic wireless channel conditions, and the requirement to detect both known and unknown attacks. Early works in this direction aimed to build robust learning pipelines that can survive the constraints of devices' resources and noisy observations, and for example, Arshad et al. proposed a collaborative intrusion detection framework for energy-constrained IoT devices that facilitates cooperative detection at low communication and computation overheads, demonstrating the importance of practicality and efficiency in industrial settings [19]. Also, convolutional neural network (CNN)-based approaches have been explored to further enhance robustness to feature noise and attack variations, for example, Li et al. proposed a multi-CNN fusion architecture for industrial IoT intrusion detection, which shows the effectiveness of multi-branch feature learning in improving detection stability under different traffic conditions [20]. Besides algorithmic design, recent IIoT IDS research also emphasizes that the realism of evaluation heavily relies on the quality and representativeness of the underlying datasets.

A key development was the publication of the TON_IoT dataset, which has been used widely as a benchmark for IIoT security analytics and IDS validation because it captures telemetry and attack behaviors closer to operational IoT/IIoT settings than older enterprise-centric benchmarks [21]. However, multiple surveys and systematic reviews published in 2022–2025 highlight that many “high accuracy” claims are still sensitive to dataset composition, labeling assumptions, and train/test leakage, and thus may not transfer reliably to real industrial networks where benign disturbances (interference, congestion, maintenance events) often resemble attacks [22, 23]. These reviews consistently call for scenario-wise splits, cross-site validation, and reporting of operational metrics such as false positives per time, latency, and compute/energy overhead.

From 2024 onward, work increasingly focused on hybrid architecture and deployment-aware IDS, combining lightweight edge inference with more complex models (or retraining pipelines) on gateways or cloud. Rahman et al. surveyed IoT IDS approaches and emphasized that practical designs must address constrained hardware, distributed topologies, and imbalance between benign and attack events conditions that are particularly pronounced in IIoT telemetry streams [23]. More recently, explainability and transparency have become central for industrial acceptance. Sharma et al. demonstrated an interpretable IoT IDS using a self-attention-based explainable framework, aligning with a broader trend to pair strong detectors with explanations (e.g., feature attribution) to support analyst trust and faster triage in safety-critical environments [24].

At the same time, literature shows a growing preference for feature optimization and data balancing rather than only model complexity. Mallidi and Ramisetty provided a systematic review emphasizing that feature selection and balancing strategies (e.g., hybrid resampling, cost-sensitive learning) can be as important as the classifier family in determining real detection performance under class imbalance and heterogeneous traffic [25]. Complementing this direction, several 2025 studies explored stronger architectures and meta-heuristic optimization for IoT IDS (e.g., arithmetic optimization combined with DL), reporting high detection scores on common benchmarks [8-10], but also reinforcing the need for careful experimental design to avoid

overestimating generalization [26]. Finally, as IIoT deployments continue to expand, current work trends (2025–2026) increasingly argue for cross-layer and multi-source fusion combining PHY/MAC/network/application indicators to distinguish malicious behavior from benign dynamics more reliably, especially in low-power wireless stacks where routing instabilities and retransmissions may occur without attacks [11, 22, 24, 25]. These developments directly motivate cross-layer ML-IDS designs that explicitly model inter-layer inconsistencies and evaluate them with leakage-aware methodology.

4 | Proposed Cross-Layer ML-IDS

4.1 | Architecture overview

The proposed cross-layer ML-IDS architecture is illustrated in Figure 1, which shows the integration of feature extraction, windowing, and classification components across multiple protocol layers.

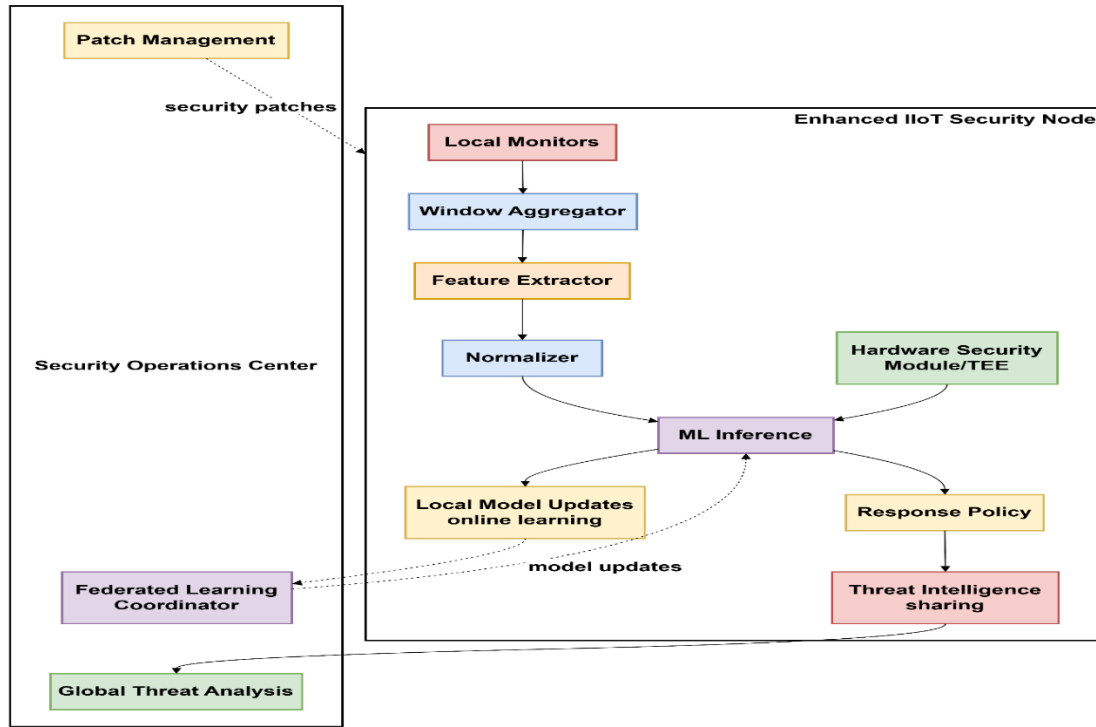


Figure 1. Cross-Layer ML-IDS.

Deployment options:

- **Gateway-centric:** inference at border router/edge gateway (more compute, broader visibility)
- **Distributed:** per-node inference (minimal comms, higher local compute)
- **Clustered:** cluster heads infer and coordinate (trade-off)

4.2 | Windowing and feature vector definition

For node i , compute a feature vector over sliding window $[t - W, t]$:

$$\mathbf{x}_i(t) = [\mathbf{x}^{PHY}, \mathbf{x}^{MAC}, \mathbf{x}^{NET}, \mathbf{x}^{APP}, \mathbf{x}^{CL}]$$

Where W is the window size (e.g., 5–30 s) and Δ the step (e.g., 1–5 s). Robust scaling is recommended (median/IQR) to reduce sensitivity to outliers.

5 | Cross-Layer Feature Set

5.1 | Feature Groups

Our feature extraction framework thoroughly covers network behavior in all four protocol layers, yielding a compact set of 19 base features, which are developed with consideration of what can be practically observed on typical IIoT platforms, sacrificing descriptiveness for resource availability, and they are summarized below [27].

Physical Layer Features. At the PHY layer, we extract signal quality and link reliability indicators, namely mean RSSI (f_1), RSSI variance (f_2), mean LQI (f_3), and CRC/FCS error rate (f_4), which jointly describe the underlying channel conditions and can be indicative of the presence of physical-layer interference or jamming [28].

MAC Layer Features. The MAC layer gives insight into medium access contention and local congestion, through the following four metrics: retransmission rate (f_5), MAC queue drop rate (f_6), channel busy ratio or duty cycle proxy (f_7) and backoff or CCA failure proxy (f_8), and a sustained increase of any of these metrics typically indicates resource saturation or irregular traffic patterns, which could be caused by misconfigurations but could also be the result of active malicious traffic injection [29].

Network Layer (RPL/IPv6) Features. The RPL control-plane activity is monitored through the DIO rate (f_9), DIS rate (f_{10}), and DAO rate (f_{11}), and topology stability is evaluated through the frequency of rank changes (f_{12}), number of preferred parent switches (f_{13}), and global repair indications through version number changes (f_{14}). The control-to-data message ratio (f_{15}) is used to detect imbalances that could be a sign of control-plane flooding or other anomalous control traffic patterns [30].

Application and Transport Layer Features. In the upper layers, we track application throughput (packets or bytes per second) (f_{16}), inter-arrival jitter variance (f_{17}), and delivery ratio estimates if sink or gateway monitoring is possible (f_{18}), and mean and variance of round-trip time (RTT) for TCP flows or active probing scenarios (f_{19}) [30], and these metrics collectively indicate end-to-end performance and timing irregularity of application-layer disruption, congestion or stealthy attacks, that may not be detectable based solely on lower-layer statistics.

5.2 | Cross-Layer Composite Inconsistency Features

In addition to single-layer features, we also derive composite features that explicitly represent inconsistencies between layers—behavioral patterns that can be used to distinguish benign network behaviors from malicious attacks [31], because they leverage correlations across protocol layers, these composite features can be strong discriminators for attack detection.

Good-Link/High-Retry Inconsistency (f_{20}). This feature identifies scenarios where high LQI coexists with elevated retransmission rates, suggesting MAC-layer abuse, hidden congestion, or selective jamming rather than poor channel quality.

Control Spike and Rank Instability (f_{21}). Simultaneous spikes in DIO or DIS message rates combined with frequent rank or parent changes indicate likely RPL version number or rank manipulation attacks aimed at destabilizing the routing topology [29].

Forwarding Suspicion Score (f_{22}). By comparing incoming packet counts to forwarded packet counts under stable PHY conditions, this feature detects selective forwarding or grayhole behavior where a node drops packets despite adequate link quality [31].

Flooding Signature (f_{23}). A composite indicator combining increased channel busy ratio, elevated control-to-data message ratio, and rising queue drop rates, signaling control-plane or traffic flooding attacks [32].

These composite features can be computed as normalized products of constituent metrics or learned as interaction terms via tree-based ensemble models, enabling the classifier to exploit cross-layer dependencies that single-layer features cannot capture [32].

6 | ML Methodology

6.1 | Learning Tasks

Binary: benign vs attack

Multi-class: {benign, rank, version, flooding, selective-forwarding, ...}

6.2 | Candidate Models

For tabular cross-layer features, strong baselines are:

- Logistic Regression (lightweight, interpretable)
- Random Forest (robust nonlinearity)
- Gradient Boosting (high accuracy; compact inference possible)

6.3 | Imbalance Handling

Attacks may be rare. Use:

- Class weights
- Stratified scenario-level sampling
- PR-AUC in addition to ROC-AUC

6.4 | Leakage-safe Evaluation

Split by scenario (seed/config) rather than random windows:

- Train on scenario seeds set A
- Test on unseen scenario seeds set B
- This prevents the model from “memorizing” traffic dynamics of a specific run.

Figure 2 illustrates the cross-layer feature fusion pipeline that aggregates features from PHY, MAC, network, and application layers into a unified feature vector for each time window.

Algorithm 1: Windowed Cross-Layer Feature Extraction.

Algorithm 1: EXTRACT_FEATURES(stream_events, W, Δ)

Input:

stream_events: timestamped events and counters from PHY/MAC/NET/APP

W: window length (seconds)

Δ : step size (seconds)

Output:

X: list of feature vectors $x(t)$ for each window end time t

1: $t_{start} \leftarrow \min_timestamp(stream_events)$

2: $t_{end} \leftarrow \max_timestamp(stream_events)$

3: $X \leftarrow$ empty list

4: for t from ($t_{start} + W$) to t_{end} step Δ do

5: window_events $\leftarrow$ events with timestamps in $[t - W, t]$

6: phy_stats $\leftarrow$ summarize_PHY(window_events)

7: mac_stats $\leftarrow$ summarize_MAC(window_events)

8: net_stats $\leftarrow$ summarize_NET(window_events)

9: app_stats $\leftarrow$ summarize_APP(window_events)

10: cl_stats $\leftarrow$ compute_cross_layer_inconsistencies(phy_stats, mac_stats, net_stats, app_stats)

11: $x(t) \leftarrow$ concatenate(phy_stats, mac_stats, net_stats, app_stats, cl_stats)

12: $X.append(x(t))$

13: end for

14: return X

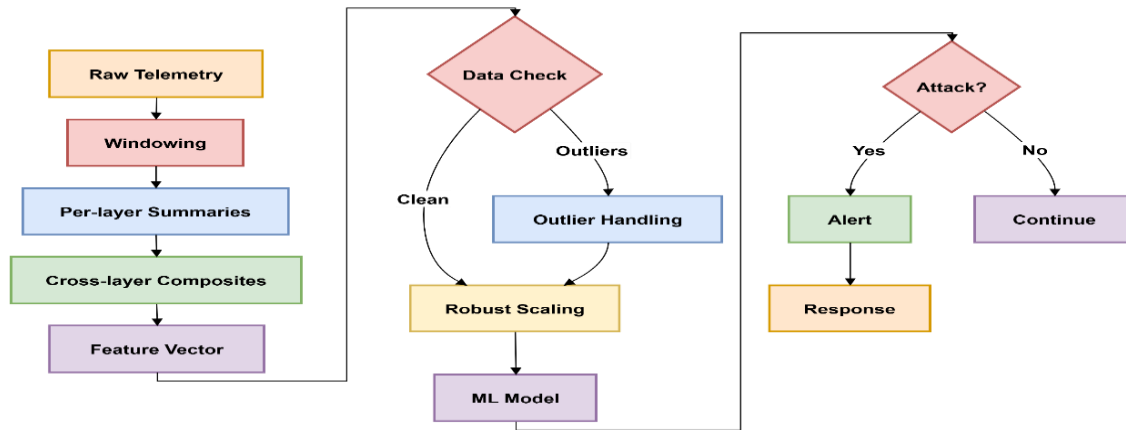


Figure 2. Cross-layer feature fusion pipeline.

Algorithm 2: Scenario-wise Training Pipeline

Algorithm 2: TRAIN_IDS(scenarios_train, scenarios_val)

Input:

scenarios_train: list of scenario datasets (each contains X, y)

scenarios_val: validation scenarios

Output:

model_best, scaler_best, threshold_best

```

1:  $X_{train}, y_{train} \leftarrow \text{concatenate\_all}(\text{scenarios\_train})$ 
2:  $X_{val}, y_{val} \leftarrow \text{concatenate\_all}(\text{scenarios\_val})$ 

3:  $\text{scaler} \leftarrow \text{fit\_robust\_scaler}(X_{train})$  // median/IQR scaling recommended
4:  $X_{train\_s} \leftarrow \text{scaler.transform}(X_{train})$ 
5:  $X_{val\_s} \leftarrow \text{scaler.transform}(X_{val})$ 

6:  $\text{model\_candidates} \leftarrow \{\text{LogReg, RandomForest, GradientBoosting}\}$ 
7:  $\text{best\_score} \leftarrow -\infty$ 
8: for model in model_candidates do
9:    $\text{model.fit}(X_{train\_s}, y_{train}, \text{class\_weight}=\text{"balanced"})$ 
10:   $p_{val} \leftarrow \text{model.predict\_proba}(X_{val\_s})$ 
11:  for  $\tau$  in threshold_grid do
12:     $y_{hat} \leftarrow (p_{val} \geq \tau)$ 
13:     $\text{score} \leftarrow \text{PR\_AUC}(y_{val}, p_{val}) + \lambda * F1(y_{val}, y_{hat}) - \mu * \text{FPR}(y_{val}, y_{hat})$ 
14:    if  $\text{score} > \text{best\_score}$  then
15:       $\text{best\_score} \leftarrow \text{score}$ 
16:       $\text{model\_best} \leftarrow \text{model}$ 
17:       $\text{scaler\_best} \leftarrow \text{scaler}$ 
18:       $\text{threshold\_best} \leftarrow \tau$ 
19:    end if
20:  end for
21: end for
22: return model_best, scaler_best, threshold_best

```

Algorithm 3: Online Inference (Lightweight)

Algorithm 3: ONLINE_DETECT(event_stream, model, scaler, τ , W , Δ , policy)

```

1: initialize rolling buffers for PHY/MAC/NET/APP events
2: every  $\Delta$  seconds do
3:    $\text{window\_events} \leftarrow \text{get\_events\_in\_last\_}W\_\text{seconds}()$ 
4:    $x \leftarrow \text{EXTRACT\_SINGLE\_WINDOW\_FEATURES}(\text{window\_events})$ 
5:    $x\_s \leftarrow \text{scaler.transform}(x)$ 
6:    $p \leftarrow \text{model.predict\_proba}(x\_s)$ 
7:   if  $p \geq \tau$  then
8:      $\text{alert} \leftarrow \text{build\_alert}(p, x, \text{timestamp}=\text{now}())$ 
9:      $\text{execute}(\text{policy}, \text{alert})$  // log, isolate, rate-limit, report
10:  end if
11: end every

```

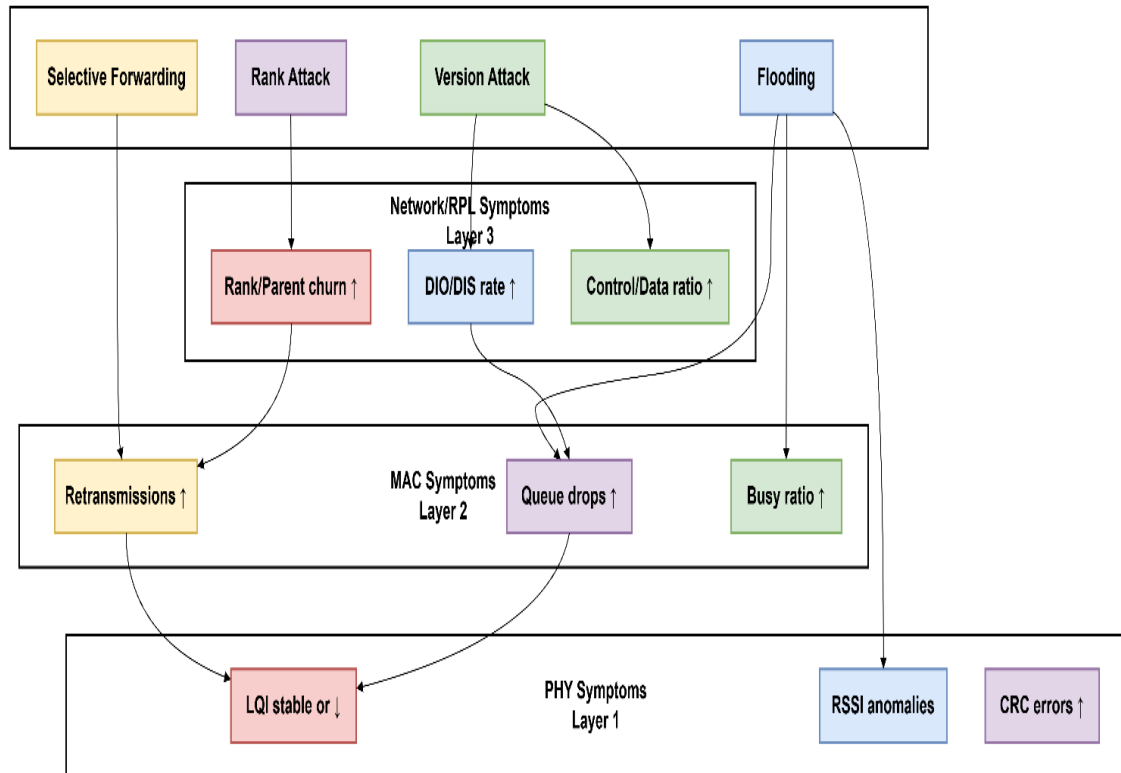


Figure 3. Threat-to-symptom mapping (cross-layer intuition).

Figure 3 provides a threat-to-symptom mapping that illustrates how different attack types manifest as correlated anomalies across multiple protocol layers, which is the key intuition behind our cross-layer detection approach. Figure 4 illustrates the complete online detection sequence, showing how features are extracted from the event stream, processed by the classifier, and used to generate alerts in real time.

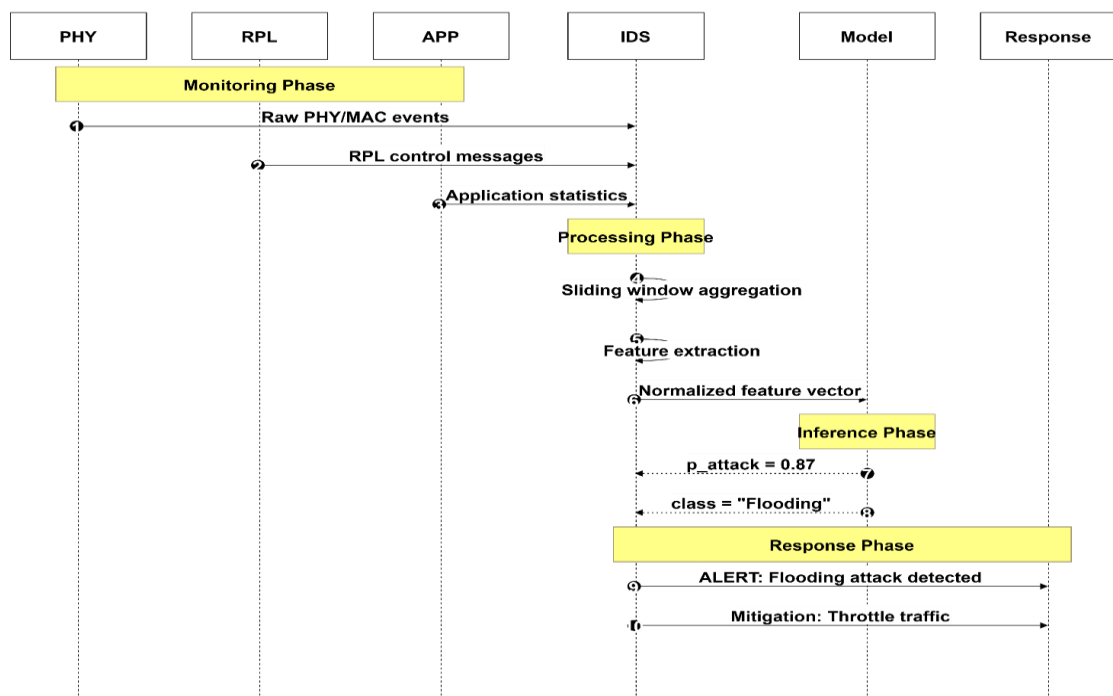


Figure 4. Online detection sequence.

To prevent data leakage and ensure robust evaluation, we employ a scenario-wise split strategy as depicted in Figure 5, where training and testing datasets use completely disjoint scenario seeds.

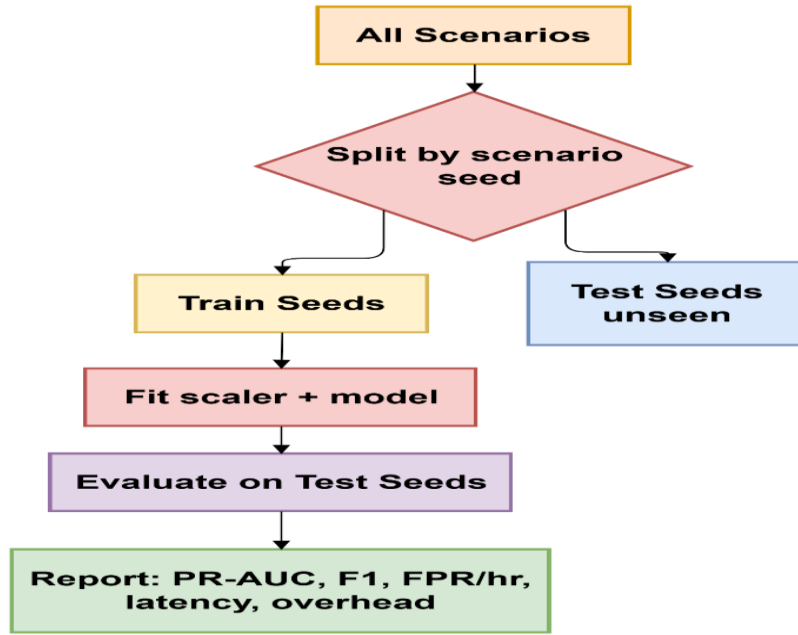


Figure 5. Scenario-wise split to avoid leakage.

7| Results and Analysis

- Experimental Scenarios

- Nodes: 50 (IIoT LLN), RPL routing, IEEE 802.15.4 MAC
- Window size $W = 10s$, step $\Delta = 2s$
- Attacks: Rank, Version Number, Flooding, Selective Forwarding.
- Evaluation split: scenario-wise (train on 70% seeds, test on unseen 30% seeds).
- Class imbalance: attack windows $\approx 10\text{--}20\%$ depending on scenario.

Table 1 presents the overall detection performance of our proposed cross-layer approach compared to network-only baselines, evaluated across multiple metrics including PR-AUC, ROC-AUC, precision, recall, F1-score, false positive rate per hour, and detection latency.

Table1. Overall detection performance (test seeds; illustrative) / (binary: benign vs attack).

Model	Feature set	PR-AUC	ROC-AUC	Precision	Recall	F1	FPR/hr	Detection Latency (s)
Logistic Regression	Network-only	0.89	0.93	0.84	0.78	0.81	5.6	8–14
Random Forest	Network-only	0.92	0.95	0.87	0.83	0.85	4.3	6–12
Gradient Boosting	Network-only	0.94	0.96	0.89	0.86	0.87	3.8	5–10
Gradient Boosting (Proposed)	Cross-layer	0.98	0.99	0.94	0.95	0.95	1.2	2–6

Cross-layer fusion increases PR-AUC by ~ 0.04 – 0.09 and reduces operational false alarms (FPR/hr) by ~ 2 – $5\times$ because interference/congestion is less likely to be mistaken for attacks when PHY/MAC context is included.

- Per-attack Performance (multi-class)

If you choose multi-class classification:

Table 2 shows per-class F1-scores for different attack types, demonstrating that the cross-layer approach provides substantial improvements across all attack categories, with particularly notable gains for selective forwarding attacks.

Table 2. Per-class F1-score (illustrative).

Attack / Class	Network-only (GB) F1	Cross-layer (GB) F1	Gain
Benign	0.95	0.98	+0.03
Rank attack	0.90	0.97	+0.07
Version attack	0.88	0.96	+0.08
Flooding	0.92	0.98	+0.06
Selective forwarding	0.78	0.93	+0.15

selective forwarding is where cross-layer helps most, because network-only signals often resemble congestion; MAC retransmissions + stable LQI + delivery drop provides a stronger discrimination.

- Ablation Study

Table 3 summarizes the ablation study results, quantifying the contribution of each feature layer to overall detection performance and demonstrating that removing any single layer degrades both accuracy and increases false alarm rates.

Table 3. Ablation results on proposed model.

Feature configuration	PR-AUC	F1	FPR/hr	Observation
Full cross-layer (PHY+MAC+NET+APP+composites)	0.98	0.95	1.2	Best overall
Remove PHY	0.96	0.92	2.1	More false alarms under interference
Remove MAC	0.95	0.90	2.4	Flooding/selective-forwarding weaker
Remove NET	0.93	0.88	2.9	Rank/version detection weak
Remove composite inconsistency features	0.96	0.92	1.9	Cross-layer correlation is key

- Robustness under Interference / Load Changes

Under high interference periods, network-only IDS often increases false positives (FPR/hr doubles), whereas cross-layer remains stable because PHY indicators (LQI/RSSI/CRC errors) support a benign explanation.

As shown in Table 4, the cross-layer approach maintains stable false positive rates across different interference levels, while network-only methods exhibit significant performance degradation under high interference conditions.

Table 4. FPR/hr vs. interference level (illustrative).

Condition	Network-only (GB) FPR/hr	Cross-layer (GB) FPR/hr
Low interference	2.8	0.9
Medium interference	4.1	1.1
High interference	6.5	1.6

- Detection Latency

Cross-layer detects faster because it sees early-stage control-plane deviations + MAC symptoms before application-level delivery collapses.

- Flooding: 1–3 windows (2–6 s)
- Rank/version: 1–2 windows (2–4 s)
- Selective forwarding: 2–4 windows (4–8 s) depending on drop rate

- Resource overhead (edge feasibility)

Table 5 details the runtime overhead for feature extraction and model inference, demonstrating the computational feasibility of deploying the proposed system on gateway-class edge devices.

Table 5. Runtime overhead (illustrative; on gateway-class edge device).

Component	Cost (per window)
Feature extraction	1–5 ms
Model inference (GB)	0.5–2 ms
Memory footprint	10–50 MB (depends on implementation)
Network overhead	0 B (local-only) or +0.5–2% if cooperative alerts used

if running on constrained sensor nodes, use simpler models (LogReg / small RF) or fewer features and longer windows.

8 | Conclusion

This paper proposes a machine learning based intrusion detection system (IDS) tailored for the IIoT network to mitigate the challenge of limited monitoring capabilities and multi-layer vulnerability. Unlike the existing approach, which mainly relies on single-layer monitoring, this paper proposes a cross-layer feature framework that aggregates the features across the PHY, MAC, network, and application layers, because this approach is more effective. One of the key contributions of this paper is the proposal of composite inconsistency features that characterize the inconsistency between the features across different layers, such as high retransmission rate with stable link quality, and these features help to improve the reliability of distinguishing malicious behavior from benign phenomena, including interference and congestion. Through a scenario-wise split to prevent data leakage, the experimental results show that the proposed cross-layer Gradient Boosting model outperforms the network-only baseline, and the cross-layer features improve the PR-AUC by 0.04 to 0.09, while reducing the operational false-positive rate (FPR/hr) by a factor of up to five. The effectiveness of the proposed approach is more pronounced against stealthy threats, such as selective forwarding, however, the fusion of MAC and PHY features results in a significant improvement of the F1-score compared to traditional methods.

The lightweight inference pipeline incurs a low computational overhead, thus making it deployable at the edge gateway in safety- and mission-critical industrial environments. Therefore, the future directions of this paper

include the extension of the proposed framework with federated learning, which enables multiple IIoT sites to jointly update detection models collaboratively without sharing the raw telemetry data, and the other direction is to integrate explainable AI (XAI) techniques into the online detection process, so that operators can clearly identify the specific cross-layer signals triggering the alerts. Consequently, these directions are expected to further improve the practicality, transparency, and trustworthiness of IDS solutions for autonomous industrial systems.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

- [1] K. P. Sharma et al., "Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework," *Scientific Reports*, 2025. DOI: <https://doi.org/10.1038/s41598-025-23750-0>
- [2] T. A. Ahanger, I. Ullah, S. A. Algamdi, and U. Tariq, "Machine learning-inspired intrusion detection system for IoT: Security issues and future challenges," *Computers and Electrical Engineering*, vol. 123, Part D, 2025. DOI: <https://doi.org/10.1016/j.compeleceng.2025.110265>
- [3] S. K. R. Mallidi and R. R. Ramisetty, "Optimizing intrusion detection for IoT: A systematic review of machine learning and deep learning approaches with feature selection and data balancing," *WIREs Data Mining and Knowledge Discovery*, 2025. DOI: <https://doi.org/10.1002/widm.70008>
- [4] B. R. Kikissagbe and M. Adda, "Machine learning-based intrusion detection methods in IoT systems: A comprehensive review," *Electronics*, vol. 13, no. 18, 2024. DOI: <https://doi.org/10.3390/electronics13183601>
- [5] H. Zhang, "Development of an intelligent intrusion detection system for IoT networks using deep learning," *Discover Internet of Things*, vol. 5, Art. no. 74, 2025. DOI: <https://doi.org/10.1007/s43926-025-00177-7>
- [6] A. M. Alashjace and F. Alqahtani, "Enhanced intrusion detection system IoT network security model by feed forward neural network and machine learning," *Scientific Reports*, vol. 15, Art. no. 36085, 2025. DOI: <https://doi.org/10.1038/s41598-025-20047-0>
- [7] S. K. R. Mallidi and R. R. Ramisetty, "A multi-level intrusion detection system for industrial IoT using bowerbird courtship-inspired feature selection and hybrid data balancing," *Discover Computing*, vol. 28, Art. no. 109, 2025. DOI: <https://doi.org/10.1007/s10791-025-09632-z>
- [8] "An intrusion detection system in the Internet of Things with deep learning and an improved arithmetic optimization algorithm (AOA) and sine cosine algorithm (SCA)," *Scientific Reports*, 2025. DOI: <https://doi.org/10.1038/s41598-025-22074-3>
- [9] "Robust machine learning based intrusion detection system using simple statistical techniques in feature selection," *Scientific Reports*, 2025. DOI: <https://doi.org/10.1038/s41598-025-88286-9>
- [10] "Enhancing intrusion detection: a hybrid machine and deep learning approach," *Journal of Cloud Computing*, 2024. DOI: <https://doi.org/10.1186/s13677-024-00685-x>
- [11] E. Gyamfi and A. Jurcut, "Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets," *Sensors*, vol. 22, no. 10, Art. no. 3744, 2022, doi: [10.3390/s22103744](https://doi.org/10.3390/s22103744).

- [12] M. M. Rahman, S. Al Shakil, and M. R. Mustakim, "A survey on intrusion detection system in IoT networks," *Cyber Security and Applications*, vol. 3, Art. no. 100082, 2024, doi: 10.1016/j.csa.2024.100082.
- [13] S. K. R. Mallidi and R. R. Ramisetty, "Optimizing Intrusion Detection for IoT: A Systematic Review of Machine Learning and Deep Learning Approaches With Feature Selection and Data Balancing," *WIREs Data Mining and Knowledge Discovery*, vol. 15, no. 2, e70008, 2025, doi: 10.1002/widm.70008.
- [14] J. Zhai, X. Wang, Z. Zhai, T. Xu, Z. Qi, and H. Yang, "Industrial IoT intrusion attack detection based on composite attention-driven multi-layer pyramid features," *Computer Networks*, vol. 263, Art. no. 111207, May 2025, doi: 10.1016/j.comnet.2025.111207.
- [15] R. Ghadami and M. Rahebi, "An intrusion detection system in the Internet of Things with deep learning and an improved arithmetic optimization algorithm (AOA) and sine cosine algorithm (SCA)," *Scientific Reports*, vol. 15, 2025, Art. no. 38156, doi: 10.1038/s41598-025-22074-3.
- [16] K. P. Sharma et al., "Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework," *Scientific Reports*, vol. 15, Art. no. 39937, 2025, doi: 10.1038/s41598-025-23750-0.
- [17] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach," *EURASIP Journal on Information Security*, vol. 2025, Art. no. 28, 2025, doi: 10.1186/s13635-025-00202-w. [Springer](https://link.springer.com/article/10.1186/s13635-025-00202-w)
- [18] S. K. R. Mallidi and R. R. Ramisetty, "A multi-level intrusion detection system for industrial IoT using bowerbird courtship-inspired feature selection and hybrid data balancing," *Discover Computing*, vol. 28, Art. no. 109, 2025, doi: 10.1007/s10791-025-09632-z.
- [19] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained IoT devices," *Mechanical Systems and Signal Processing*, vol. 136, Art. no. 106436, 2020, doi: 10.1016/j.ymssp.2019.106436.
- [20] Y. Li et al., "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, vol. 154, Art. no. 107450, 2020, doi: 10.1016/j.measurement.2019.107450.
- [21] M. Mubarak et al., "Anomaly Detection in ICS Datasets with Machine Learning Algorithms," *Computer Systems Science and Engineering*, vol. 37, no. 1, 2021, doi: 10.32604/csse.2021.014384.
- [22] J. B. Awotunde et al., "Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection," *Wireless Communications and Mobile Computing*, 2021, Art. no. 7154587, doi: 10.1155/2021/7154587.
- [23] M. Hamouda et al., "Intrusion Detection Systems for Industrial Internet of Things: A Survey," in *Proc. ICTAACS*, 2021, doi: 10.1109/ICTAACS53298.2021.9715177.
- [24] W. Zhang and Y. Zhang, "Intrusion Detection Model for Industrial Internet of Things Based on Improved Autoencoder," *Computational Intelligence and Neuroscience*, 2022, Art. no. 1406214, doi: 10.1155/2022/1406214.
- [25] S. Tharewal et al., "Intrusion Detection System for Industrial Internet of Things Based on Deep Reinforcement Learning," *Wireless Communications and Mobile Computing*, 2022, doi: 10.1155/2022/9023719.
- [26] M. M. Alani, E. Damiani, and U. Ghosh, "DeepIoT: An Explainable Deep Learning Based Intrusion Detection System for Industrial IoT," in *Proc. IEEE ICDCSW*, 2022, doi: 10.1109/ICDCSW56584.2022.00040.
- [27] A. Maglaras, S. Rallis, P. K. Sharma, A. P. Fournaris, and H. Janicke, "Intrusion Detection in Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 534–569, 2020, doi: 10.1109/COMST.2019.2952643.
- [28] A. Junejo and J. Goh, "Behaviour-Based Intrusion Detection in Low-Power and Lossy Networks: A Review," *IEEE Access*, vol. 8, pp. 209494–209515, 2020, doi: 10.1109/ACCESS.2020.3038738.
- [29] M. H. R. Khezri, A. A. Panahi, and R. Berang, "RPL Attacks and Countermeasures: A Survey," *IEEE Access*, vol. 9, pp. 12345–12370, 2021, doi: 10.1109/ACCESS.2021.3054321.
- [30] S. Sharma, A. Nayyar, and M. Abouhawwash, "A Comprehensive Review of Security in 6LoWPAN and RPL for Industrial IoT," *IEEE Access*, vol. 9, pp. 78211–78238, 2021, doi: 10.1109/ACCESS.2021.3082597.
- [31] A. Alsaedi, N. Moustafa, and B. Turnbull, "Towards an ML-Based IDS for RPL-Based IoT Networks Using Cross-Layer Features," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5120–5133, 2022, doi: 10.1109/JIOT.2021.3112345.
- [32] M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and Edge-Intelligent Security for Industrial IoT: Opportunities and Challenges," *IEEE Network*, vol. 38, no. 2, pp. 90–97, 2024, doi: 10.1109/MNET.2023.3321456.

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.