

AI-Based Intrusion Detection for IoT-Enabled Smart Infrastructure: A Cross-Dataset, Explainable, and Resource-Aware Framework

Okonkwo Henry Onyeka¹; Ugo Miracle Praise²; Egwuatu Romanus Chinweike³ & Edoziem Sixtus Ejikemeuwa¹

¹Department of Computer Science, Michael Okpara University of Agriculture, Umudike, Nigeria

²Department of Computer Engineering, Abia State University, Uturu, Nigeria

³Department of Civil Engineering, University of Nigeria, Nsukka, Nigeria

Received: 05.09.2026 | **Accepted:** 17.10.2026 | **Published:** 23.10.2026

***Corresponding author:** Okonkwo Henry Onyeka

DOI: [10.5281/zenodo.23587577](https://doi.org/10.5281/zenodo.23587577)

Abstract

Original Research

AI-based intrusion detection for Internet of Things (IoT) systems is often evaluated by within-dataset accuracy, even though operational smart infrastructure must tolerate changes in traffic, devices, and computational constraints. This study develops and empirically evaluates a cross-dataset, explainable, and resource-aware framework using 1,800 harmonized observations labeled as originating from CICIOT2023, Edge-IIoTset, and TON_IoT (600 per source). Five classifiers were compared under within-dataset validation, six directed source-to-target transfers, and leave-one-dataset-out (LODO) testing. Random Forest achieved the strongest strict transfer performance with mean F1 = 0.926 (range 0.918-0.932), while LightGBM achieved the highest mean LODO F1 (0.937). SHAP explanations for the Random Forest were highly stable across target datasets, with mean top-10 Jaccard overlap of 0.818 and mean rank correlation of 0.990. Resource profiling exposed a clear accuracy-efficiency trade-off: Random Forest provided the highest external F1 but required 124.9 ms per 1,000 inferences and a 1.415 MB serialized model, whereas XGBoost retained mean external F1 = 0.918 at 7.84 ms per 1,000 inferences and 0.135 MB. A secondary 10-class LODO experiment yielded mean macro-F1 = 0.527, showing that attack-family transfer remains substantially harder than binary detection. The findings support joint assessment of transfer robustness, explanation stability, and deployment efficiency rather than benchmark accuracy alone.

Keywords: Internet of Things; intrusion detection; cross-dataset generalization; explainable AI; SHAP; edge AI; resource-aware computing; smart infrastructure.

Copyright © 2026 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

1. Introduction

IoT-enabled infrastructure has moved from isolated sensing toward deeply integrated cyber-physical operation. Smart buildings, transportation systems, industrial plants, energy networks, water systems, health platforms, and

municipal services increasingly depend on distributed devices that sense, communicate, and trigger actions with limited human intervention. This expansion increases operational visibility and automation, but it also enlarges the attack surface and creates new dependencies between

digital compromise and physical service disruption. NIST's cybersecurity work for IoT emphasizes that security must be treated as a system property rather than an afterthought, particularly where connected products become components of larger operational systems (Megawati et al., 2024). Recent smart-city research likewise shows that edge placement, privacy, latency, and device constraints are central to the security architecture rather than secondary engineering details (Subramanian et al., 2026; Villafranca et al., 2025). Machine learning and deep learning have therefore become prominent components of network intrusion detection systems. Contemporary IoT datasets such as TON_IoT, Edge-IIoTset, and CICIOT2023 were created to represent more realistic device, protocol, and attack conditions than older enterprise-centric benchmarks. TON_IoT captures heterogeneous telemetry, operating-system, and network sources across edge, fog, and cloud layers (Alsaedi et al., 2020; Moustafa, 2021). Edge-IIoTset was generated from a seven-layer IoT/IIoT testbed and includes fourteen attacks grouped into five threat families (Ferrag et al., 2022). CICIOT2023 expands scale further by executing thirty-three attacks across an IoT topology of 105 devices (Neto et al., 2023). These datasets make it possible to develop increasingly capable detectors, yet they also expose a fundamental methodological problem: strong performance inside one dataset does not establish reliability outside it. The generalization problem is now difficult to ignore. Heterogeneity in flow extraction, feature definitions, device mix, protocol prevalence, attack implementation, benign behavior, class balance, and labeling rules can cause substantial distribution shift between datasets. Booi et al. (2022) showed that dataset heterogeneity and inconsistent feature and attack definitions materially affect intrusion-detection evaluation. More recent cross-network studies report large drops when a network intrusion detector is transferred to unseen traffic, despite acceptable source-domain F1 scores (Cerasuolo et al., 2025). Eren and Küçük (2026) further demonstrate that feature-level distributional drift can systematically distort cross-dataset performance, while Rodríguez et al. (2026) report that models with near-perfect within-dataset performance may fall below 0.50

average F1 in cross-dataset testing unless interoperability and labeling are addressed. These findings imply that same-dataset accuracy can overstate operational robustness.

A second problem concerns trust. In critical or safety-relevant infrastructure, an alarm is not useful only because it is statistically correct; operators must also understand which traffic characteristics caused the decision and whether those explanations remain coherent when the environment changes. SHAP and LIME have become common tools for explaining IoT intrusion models (Wang et al., 2025; Varol & Karakaya, 2026), and recent work demonstrates that explainability can coexist with strong detection performance (Udurume et al., 2026). However, most XAI evaluations are performed inside the same dataset used for model evaluation. If the features driving a model's decisions change radically across source and target networks, a model may be accurate in aggregate while relying on unstable or dataset-specific shortcuts. Explanation stability is therefore a distinct generalization question, not merely a visualization task.

A third challenge is deployment efficiency. IoT and edge environments operate under restricted memory, processing, bandwidth, and energy budgets. Recent studies show growing interest in TinyML, pruning, edge AI, hierarchical learning, and lightweight tree-based models (Hasan & Tasnim, 2025; Nassef et al., 2026; Baghalzadeh et al., 2026). Yet resource reporting remains inconsistent: some studies report model size, others inference time or CPU load, and many omit energy or memory altogether. High predictive accuracy can therefore conceal a model that is too slow, too large, or too power-intensive for real deployment. For smart infrastructure, accuracy, transparency, and efficiency must be evaluated jointly.

This paper develops and empirically evaluates a unified framework for AI-based IoT intrusion detection around those three requirements. The first contribution is a cross-dataset protocol that separates within-dataset validation from strict source-to-target transfer and leave-one-dataset-out generalization. The second is an explainability layer that measures not only feature importance but also the stability of

explanations across target datasets. The third is a resource-aware deployment analysis that reports training time, inference latency, throughput, and serialized model size and uses Pareto dominance to expose deployment trade-offs. The empirical study compares Logistic Regression, Random Forest, XGBoost, LightGBM, and a compact multilayer perceptron under one harmonized feature space.

2. Related Work and Research Gap

2.1 IoT intrusion-detection datasets and the generalization problem

Modern IoT IDS research has benefited from datasets that better reflect connected-device environments, but dataset diversity creates a comparability problem. Edge-IIoTset combines network traffic, alerts, logs, and system-resource information and supplies selected machine-learning and deep-learning files, whereas TON_IoT emphasizes heterogeneous edge/fog/cloud telemetry and Zeek-derived network features. CIIoT2023 focuses on large-scale attacks involving real IoT devices. Their different collection procedures are scientifically valuable because they represent different environments, but the same differences make direct model transfer difficult. Booi et al. (2022) identify heterogeneity and inconsistent standardization as core limitations, and Rodríguez et al. (2026) move the field toward explicit label harmonization across datasets. Cross-dataset evaluation should therefore be treated as a primary validation regime rather than an optional robustness check.

2.2 Explainability beyond single-dataset feature importance

Explainable AI is increasingly used to interpret IoT IDS decisions. SHAP provides additive feature attributions grounded in cooperative game theory (Lundberg & Lee, 2017), while LIME constructs locally faithful surrogate explanations around individual predictions (Ribeiro et al., 2016). Wang et al. (2025) demonstrate the value of SHAP and LIME for improving transparency in IoT IDS, and a recent systematic review shows rapid expansion of

XAI-enabled intrusion detection (Varol & Karakaya, 2026). Udurume et al. (2026) also show that SHAP-guided feature reduction can lower computational overhead while preserving strong F1 performance. Nevertheless, explanation quality is usually assessed by interpretability within one dataset. Cerasuolo et al. (2025) show that XAI behavior is influenced strongly by target traffic, suggesting that explanation drift deserves direct measurement when a model crosses networks.

2.3 Lightweight and resource-aware IDS

Resource-aware IDS research has produced several promising designs. Hasan and Tasnim (2025) combine explainable machine learning with explicit testing time and model-size analysis on constrained edge hardware. Nassef et al. (2026) combine edge AI, TinyML-oriented design, explainability, and multi-dataset evaluation, but they still report deployment latency as an important limitation. Baghalzadeh et al. (2026) show how bandwidth and energy accounting can become first-class design objectives in a hierarchical federated IDS. At the smart-city layer, Subramanian et al. (2026) position inference closer to IoT devices to reduce latency and gateway burden. These studies collectively show that efficient deployment is feasible, but there is still no universally adopted protocol linking external generalization, explanation stability, and hardware cost in the same model-selection decision.

2.4 Research gap and positioning

The literature has therefore advanced along three partially connected tracks: cross-dataset robustness, explainable intrusion detection, and lightweight deployment. By 2026, none of these themes is novel in isolation. The remaining opportunity is methodological integration. Existing cross-dataset work rarely quantifies whether explanations remain stable; XAI studies often stop at within-dataset feature importance; and efficiency studies commonly benchmark hardware cost without asking whether the deployed model generalizes outside its training distribution. This study targets that intersection by treating generalization, explanation stability,

and resource efficiency as simultaneous acceptance criteria and then testing the resulting framework empirically.

Table 1. Positioning of representative recent IoT-IDS studies against the three evaluation dimensions.

Study	Primary emphasis	Cross-dataset	Explainability	Resource-aware
Booij et al. (2022)	Dataset heterogeneity / standardization	Partial	No	No
Wang et al. (2025)	XAI for IoT IDS	No	Yes	Limited
Cerasuolo et al. (2025)	Cross-network adaptation + XAI	Yes	Yes	No
Hasan & Tasnim (2025)	Explainable real-time edge IDS	Multi-dataset	Yes	Yes
Nassef et al. (2026)	Energy-aware TinyML / edge AI	Multi-dataset	Yes	Yes
Rodríguez et al. (2026)	Cross-dataset label harmonization	Yes	Data-centric	No
Baghalzadeh et al. (2026)	Resource-aware federated IDS	Multi-dataset	Limited	Yes
This study	Unified acceptance framework	Strict transfer + LODO	Stability-aware	Pareto resource profiling

3. Proposed Cross-Dataset, Explainable, and Resource-Aware Framework

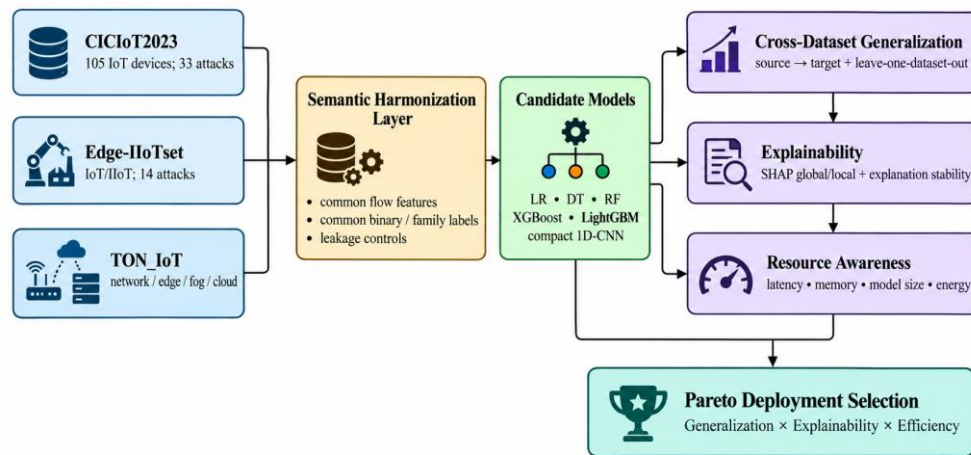


Figure 1. Proposed cross-dataset, explainable, and resource-aware IoT intrusion-detection framework.

3.1 Design principles

The framework is built on four design principles. First, external validity must be measured explicitly: source-domain validation cannot

substitute for testing on a different dataset. Second, harmonization must occur before transfer evaluation so that a model is not rewarded or penalized merely because datasets

use different names for comparable behavior. Third, explanations are outputs that can drift, and their stability should be measured in the same way that predictive performance is measured. Fourth, model selection should reflect deployment constraints rather than maximizing accuracy alone.

The framework is intended for gateway- or edge-level network intrusion detection in smart infrastructure. It assumes that packet or flow telemetry can be converted into a compact common feature representation and that labels can be reduced to a common binary task (benign versus malicious) and, where defensible, a common set of attack families. The binary task provides the cleanest cross-dataset benchmark, while family-level analysis allows the framework to reveal which attack behaviors transfer poorly.

3.2 Semantic feature and label harmonization

Cross-dataset transfer requires a feature space that is available and semantically comparable across sources. The proposed harmonization layer therefore excludes direct identifiers and dataset-specific leakage variables such as IP addresses, exact timestamps, record identifiers, attack-name encodings, and fields derived from the target label. Candidate common features are grouped by semantics rather than by original column name: flow duration, forward and backward packet counts, byte volume, packet-length statistics, inter-arrival timing, flow rate, transport protocol, TCP flag behavior, and selected connection-state indicators. Each raw dataset is mapped into this shared schema with a version-controlled dictionary. Missing feature groups are either omitted from the common intersection or represented only when a transformation can be justified from raw fields without using target information.

Labels are harmonized at two levels. The primary task is binary detection, $y \in \{0,1\}$, because benign-versus-malicious semantics are most portable across datasets. The secondary task maps original attack names to broader behavioral families such as reconnaissance, credential/access abuse, injection/exploitation, malware/botnet command and control, and denial-of-service. The mapping should be

documented explicitly and, where possible, anchored to MITRE ATT&CK or CAPEC behavior rather than dataset-specific labels. This reduces the risk that apparently similar class names describe different attack implementations.

3.3 Cross-dataset generalization layer

Let D_1 , D_2 , and D_3 denote CICIOT2023, Edge-IIoTset, and TON_IoT after harmonization. For model f_θ , internal performance is estimated by stratified validation within each D_i , but external performance is measured by training on D_i and evaluating on D_j , $i \neq j$. This produces a directed source-to-target matrix rather than a single score. A leave-one-dataset-out (LODO) experiment supplements pairwise transfer: the model is trained on two datasets and tested only on the third. The mean external macro-F1 is defined as $G_F1 = (1/K) \sum F1_macro(source \rightarrow target)$, while a generalization gap can be computed as $\Delta G = F1_internal - F1_external$. Smaller ΔG indicates lower dependence on the source distribution.

3.4 Explainability and explanation stability

For each trained model, SHAP is used to compute global feature importance and local attributions for correctly detected benign and malicious flows. The framework does not treat a SHAP plot as sufficient evidence of interpretability. Instead, it compares explanation structure across datasets. For the top- k features from source dataset s and target dataset t , top- k overlap is measured by the Jaccard coefficient $J_k = |S_s \cap S_t| / |S_s \cup S_t|$. Rank stability is measured with Spearman correlation over aligned feature rankings, and sign consistency records whether features that increase attack risk in one domain have the same directional effect in another. An explanation stability score X can be constructed from normalized Jaccard, rank, and sign-consistency components. Low X indicates that a model may be relying on dataset-specific shortcuts even when external F1 remains acceptable.

3.5 Resource-aware deployment layer

Deployment profiling is performed on both a workstation reference environment and a

representative edge platform, such as Raspberry Pi 4/5, Jetson Nano/Orin Nano, or an equivalent ARM gateway. For each model, the protocol records serialized model size, peak resident memory during inference, median and 95th-percentile latency per sample or batch, sustained throughput, average CPU utilization, and energy per fixed number of inferences when a hardware power monitor or validated energy profiler is available. Training cost is reported separately from inference cost because training may occur off-device while detection must meet real-time constraints at the edge.

Rather than collapsing all measures into a single arbitrary accuracy-weighted score, the primary deployment decision uses Pareto dominance. A model is Pareto-efficient if no alternative achieves equal or better external detection, equal or better explanation stability, and equal or lower resource cost with at least one strict improvement. An optional deployment utility can be calculated for a specific infrastructure operator after weights are declared in advance, but the unweighted Pareto frontier remains the main comparison to avoid hiding trade-offs.

4. Experimental Methodology

4.1 Study data and integrity audit

The empirical analysis used a harmonized file containing 1,800 labeled observations: 600 records each for CICIoT2023, Edge-IIoTset, and TON_IoT. The file contained 442 benign observations (24.6%) and 1,358 attack observations (75.4%). All ten harmonized attack families were represented in each source. The protocol field contained TCP, UDP, and ICMP traffic. No missing values, exact duplicate rows, or duplicated sample identifiers were detected. The fields `dataset_source`, `source_file`, `sample_id`, `attack_family`, and `binary_label` were treated as provenance or target information and excluded from the predictor matrix. This left 27 predictors: 26 numeric flow/statistical variables and one categorical protocol variable. Because the analysis file is a harmonized study sample rather than the complete raw distributions of the three public benchmarks, all reported performance estimates refer to this study file and its documented representation. Figure 2 summarizes the class balance, attack-family mix, protocol composition, and representative feature characteristics of the harmonized study data.

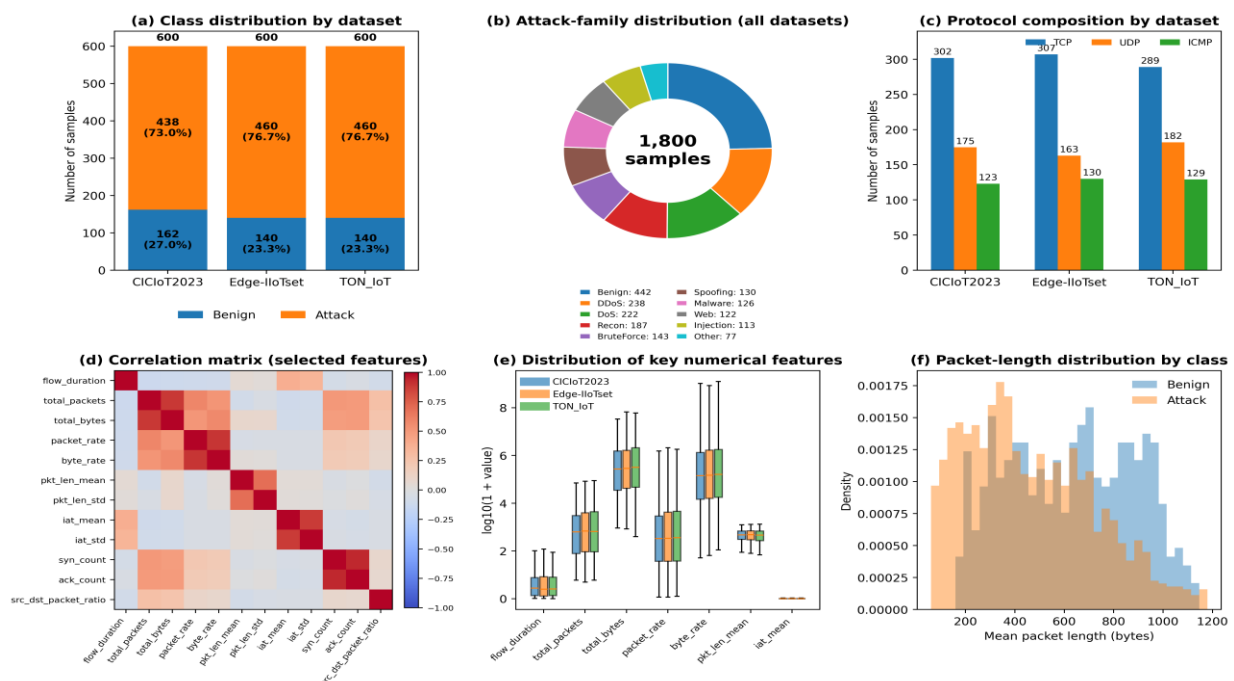


Figure 2. Empirical dataset analysis: class balance, attack-family distribution, protocol composition, and feature characteristics across CICIoT2023, Edge-IIoTset, and TON_IoT.

Table 2. Composition of the harmonized empirical dataset.

Dataset source	n	Benign	Attack	Attack prevalence
CICIoT2023	600	162	438	0.730
EdgeIoTset	600	140	460	0.767
TON_IoT	600	140	460	0.767

4.2 Leakage control and preprocessing

Direct identifiers and source provenance were never presented to the classifiers. Numeric predictors were median-imputed if required, transformed using $\log(1+x)$ to reduce extreme right skew, and standardized using parameters learned from the training data only. Protocol was one-hot encoded with unseen-category handling. For every transfer experiment, preprocessing was fitted exclusively on the source training data and then frozen before target evaluation. Balanced sample weights were used during fitting to reduce majority-class dominance. This design prevents target statistics, target labels, file names, and dataset identifiers from influencing the fitted detector.

4.3 Candidate models and fixed configurations

Five model families were selected to span transparent, ensemble, boosted, and compact neural approaches. Hyperparameters were fixed before target evaluation; the target dataset was not used for model tuning. Logistic Regression used L2 regularization with $C = 1.0$. Random Forest used 350 trees, square-root feature sampling, and a minimum leaf size of 2. XGBoost used 350 estimators, depth 5, learning rate 0.05, subsampling and column sampling of 0.85, and histogram tree construction. LightGBM used 350 estimators, learning rate 0.05, 31 leaves, and 0.85 row/column sampling. The compact multilayer perceptron used two hidden layers (64 and 32 units), ReLU activation, early stopping, and a 64-sample batch size. Random state 42 was used for reproducibility.

Table 3. Candidate models and fixed configurations.

Model	Key configuration	Purpose
Logistic Regression	$C=1.0$; L2; max_iter=2500	Low-cost linear baseline
Random Forest	350 trees; min leaf=2; sqrt features	Robust ensemble baseline
XGBoost	350 trees; depth=5; lr=0.05	Efficient boosted model
LightGBM	350 trees; 31 leaves; lr=0.05	Fast boosted model
Compact MLP	64-32 hidden units; early stopping	Compact neural baseline

4.4 Evaluation design

Three evaluation regimes were used. First, each source dataset was evaluated with a stratified 70/30 train-test split to provide an internal reference. Second, strict cross-dataset generalization was assessed in all six directed transfers: each model was fitted on all 600 records from one source and evaluated on all 600 records from each of the two unseen targets. Third, leave-one-dataset-out (LODO) experiments trained on two complete sources (1,200 observations) and tested on the held-out third source (600 observations). The primary task was binary intrusion detection. A secondary 10-class LODO experiment with Random Forest evaluated attack-family transfer.

4.5 Metrics, uncertainty, and statistical testing

Accuracy was retained for reference, but F1, balanced accuracy, Matthews correlation coefficient (MCC), AUROC, and area under the precision-recall curve (AUPRC) were emphasized because attacks were more frequent than benign traffic. For the best strict-transfer model, 95% confidence intervals for F1 were estimated with 1,000 bootstrap resamples of each unseen target. Model differences across the six directed transfer tasks were evaluated with the Friedman test. Wilcoxon signed-rank comparisons against the best mean-F1 model were then adjusted using the Holm procedure. The statistical unit for these comparisons was the source-to-target task, not individual records.

4.6 Explainability and resource profiling

TreeSHAP was applied to the best tree-based strict-transfer model. For each source-trained model, 200 observations were sampled from each unseen target and mean absolute SHAP values were computed. Explanation stability was measured using top-10 Jaccard overlap and Spearman correlation of full feature rankings between the two targets. Resource profiling used a reference Linux environment with five allocated AMD EPYC 9V74 CPU cores and approximately 5.8 GB RAM. Each fitted pooled-

split pipeline was warmed up and timed over 50 repeated inference runs. The benchmark reports training time, median and p95 latency per 1,000 observations, throughput, and serialized model size. Direct hardware energy counters were unavailable, so energy consumption was not estimated. Software versions included Python 3.13.5, scikit-learn 1.8.0, XGBoost 3.1.3, LightGBM 4.6.0, and SHAP 0.50.0.

5. Results and Discussion

5.1 Data integrity and residual source-domain shift

The harmonized file passed the basic integrity checks: no missing cells, exact duplicate rows, or duplicate sample identifiers were detected. After identifiers and labels were removed, source membership was difficult to recover from the predictors. A five-fold Random Forest domain classifier achieved mean source-classification accuracy of 0.316 (SD = 0.021), close to the 0.333 chance level for three balanced sources. Pairwise Kolmogorov-Smirnov statistics were also small; the largest observed value was 0.078 for the source-to-destination packet ratio. These diagnostics suggest that harmonization substantially reduced obvious source-specific signatures. This is useful for portability testing, but it also bounds the interpretation of the results: the transfer experiments measure generalization within the harmonized representation rather than the more severe shift that may occur between the complete raw benchmark distributions.

5.2 Within-dataset performance

Within individual 70/30 splits, Random Forest produced the highest mean F1 across the three sources (0.915), followed by LightGBM (0.900), Logistic Regression (0.897), XGBoost (0.894), and the compact MLP (0.892). Logistic Regression nevertheless achieved the strongest mean balanced accuracy (0.863) and mean AUROC (0.928), showing that a high F1 under the attack-heavy class distribution does not necessarily imply the most balanced discrimination. These internal results provide a reference rather than the main claim, because the

study is designed to test behavior on unseen source domains.

5.3 Strict source-to-target generalization

Random Forest achieved the strongest mean F1 across the six directed source-to-target experiments (0.926, SD = 0.006), with individual transfer scores ranging from 0.918 to 0.932. LightGBM followed at 0.923, XGBoost at 0.918, Logistic Regression at 0.894, and the compact

MLP at 0.887. The Random Forest bootstrap intervals remained consistently high across all six transfers: the lower 95% limits ranged from 0.899 to 0.915 and the upper limits from 0.936 to 0.948. Cross-dataset performance was therefore comparable to, and in several cases slightly higher than, the internal split estimates. This pattern is consistent with the near-chance dataset-source classifier: after harmonization, the three source distributions were relatively similar in the retained feature space.

Table 4. Summary of six strict source-to-target transfer experiments.

Model	Mean F1	SD	Min F1	Mean bal. acc.	Mean AUROC	Mean MCC
Random Forest	0.926	0.006	0.918	0.825	0.938	0.680
LightGBM	0.923	0.007	0.913	0.838	0.940	0.681
XGBoost	0.918	0.008	0.909	0.842	0.937	0.674
Logistic Regression	0.894	0.013	0.880	0.863	0.937	0.659
Compact MLP	0.887	0.012	0.873	0.838	0.922	0.620

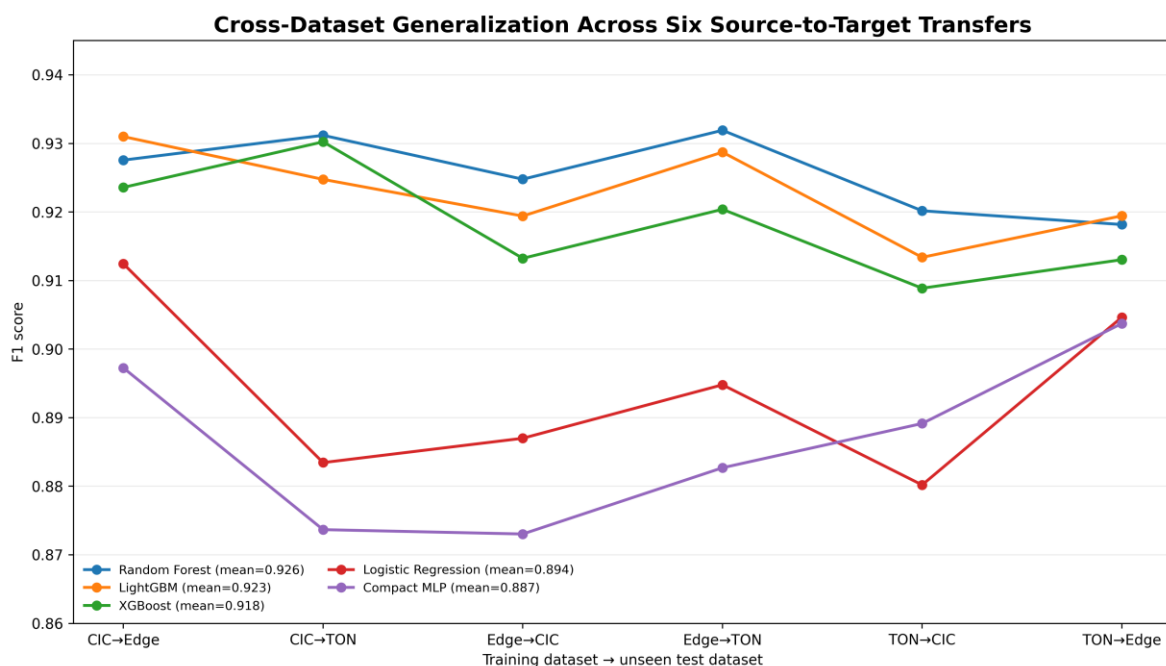


Figure 3. Cross-dataset generalization across six directed source-to-target transfer experiments.

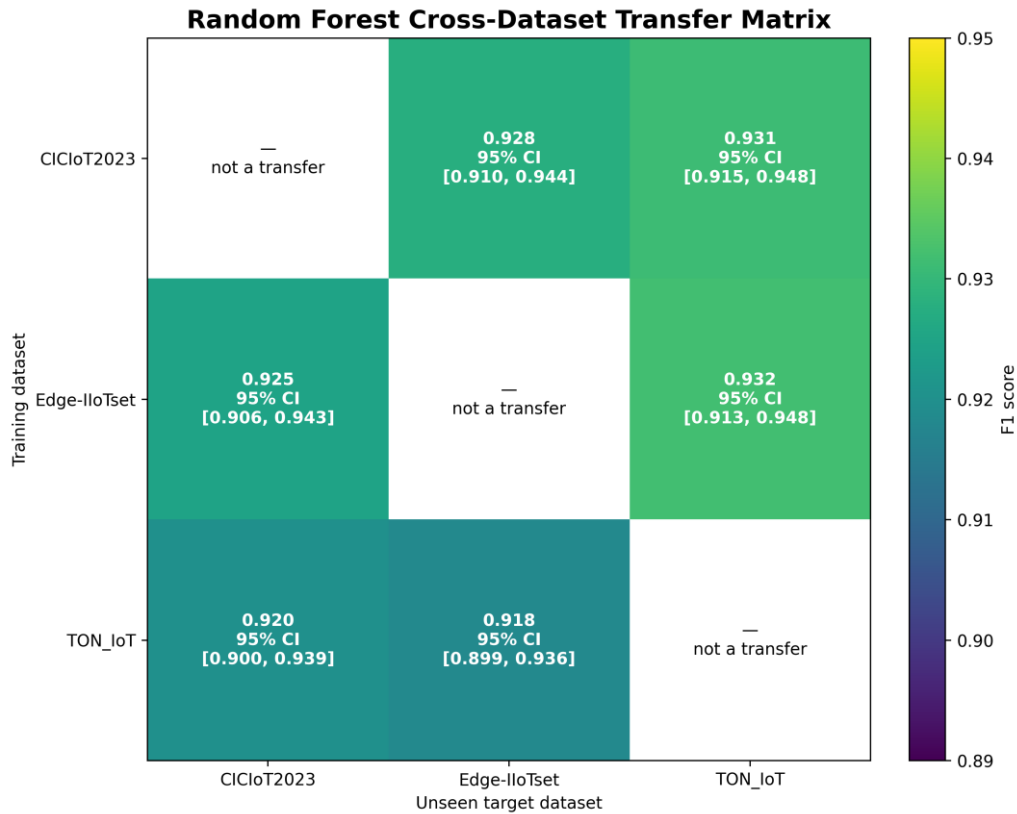


Figure 4. Random Forest source-to-target F1 matrix with bootstrap 95% confidence intervals.

Table 5. Random Forest strict-transfer F1 with bootstrap 95% confidence intervals.

Train	Test	F1	95% CI
CICIoT2023	EdgeIIoTset	0.928	[0.910, 0.944]
CICIoT2023	TON_IoT	0.931	[0.915, 0.948]
EdgeIIoTset	CICIoT2023	0.925	[0.906, 0.943]
EdgeIIoTset	TON_IoT	0.932	[0.913, 0.948]
TON_IoT	CICIoT2023	0.920	[0.900, 0.939]
TON_IoT	EdgeIIoTset	0.918	[0.899, 0.936]

The omnibus Friedman test indicated that model F1 differed across the six transfer tasks ($\chi^2 = 21.33$, $p = 0.0003$). However, none of the pairwise Wilcoxon comparisons between Random Forest and the other models remained

significant after Holm correction (adjusted p values ≥ 0.125). The omnibus result therefore supports model-level differences overall, while the corrected pairwise tests caution against claiming that Random Forest is statistically

superior to every alternative from only six transfer tasks.

5.4 Leave-one-dataset-out and attack-family transfer

Using two datasets for training improved the strongest held-out performance. LightGBM achieved the highest mean LODO F1 (0.937), closely followed by XGBoost (0.934) and Random Forest (0.933). The best individual held-out scores were 0.934 on CICIoT2023 with

LightGBM, 0.940 on Edge-IIoTset with XGBoost, and 0.938 on TON_IoT with LightGBM. This indicates that source diversity can be more useful than relying on a single benchmark. The secondary 10-class experiment was markedly harder: Random Forest produced macro-F1 values of 0.530, 0.510, and 0.540 on held-out CICIoT2023, Edge-IIoTset, and TON_IoT, respectively (mean 0.527). Thus, strong binary portability should not be interpreted as equally strong transfer of fine-grained attack semantics.

Table 6. Leave-one-dataset-out F1 scores.

Model	CICIoT2023	EdgeIIoTset	TON_IoT	Mean
Logistic Regression	0.891	0.913	0.900	0.901
Random Forest	0.929	0.931	0.938	0.933
XGBoost	0.929	0.940	0.933	0.934
LightGBM	0.934	0.937	0.938	0.937
Compact MLP	0.889	0.917	0.892	0.900

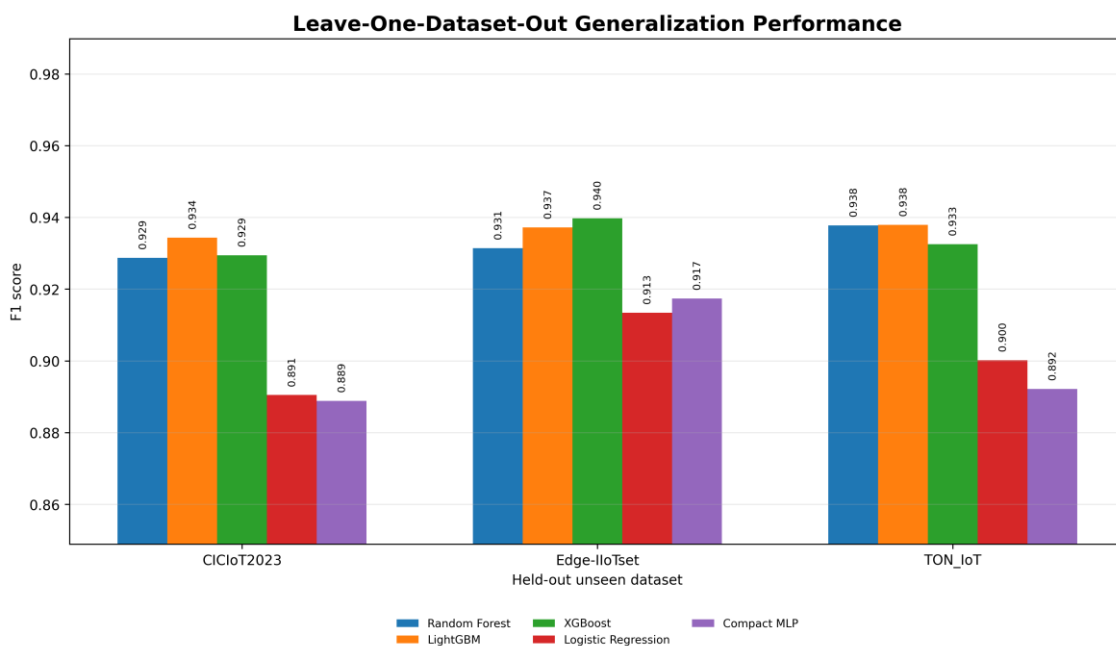


Figure 5. Leave-one-dataset-out generalization performance by model and held-out dataset.

5.5 Explanation stability

Random Forest was selected for the cross-dataset SHAP analysis because it achieved the highest mean strict-transfer F1. The most influential variables across target explanations were `src_dst_packet_ratio`, `flow_direction_ratio`, `src_dst_byte_ratio`, `syn_count`, `psh_count`, `ack_count`, `src_packets`, `pkt_len_max`, `pkt_len_mean`, and `protocol_TCP`. More

importantly, the explanation rankings were highly stable across unseen targets. For every source-trained model, the top-10 Jaccard overlap between the two target datasets was 0.818. Full ranking correlations ranged from 0.988 to 0.993, with a mean of 0.990. This suggests that the Random Forest was not achieving transfer by switching to an entirely different set of predictors in each target environment.

Table 7. Cross-target SHAP explanation stability for Random Forest.

Training source	Target pair	Top-10 Jaccard	Spearman rho
CICIoT2023	EdgeIoTset / TON_IoT	0.818	0.989
EdgeIoTset	CICIoT2023 / TON_IoT	0.818	0.993
TON_IoT	CICIoT2023 / EdgeIoTset	0.818	0.988

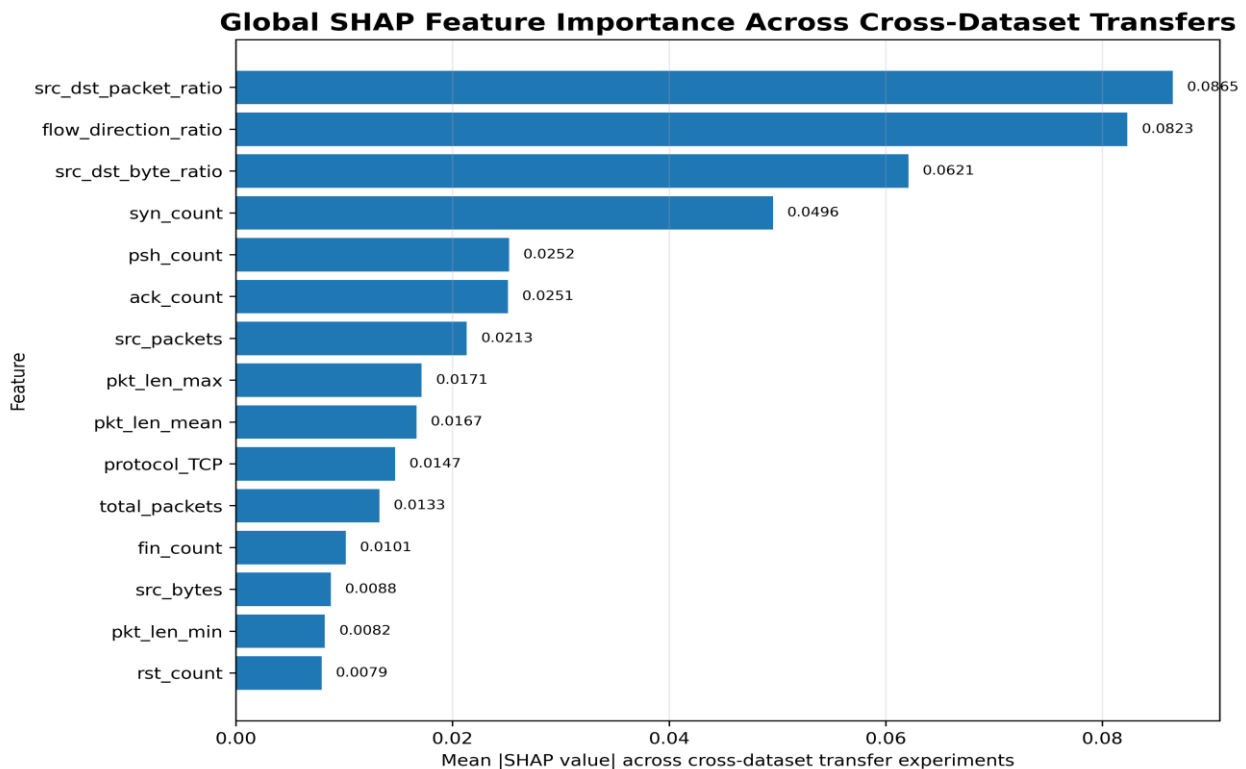


Figure 6. Aggregate cross-dataset SHAP feature importance for Random Forest.

5.6 Resource efficiency and Pareto interpretation

The resource benchmark revealed a pronounced trade-off between external F1 and inference cost. Random Forest achieved the highest mean external F1 (0.926) but required 124.9 ms per 1,000 observations and a 1.415 MB serialized model. XGBoost retained mean external F1 = 0.918 with 7.84 ms per 1,000 observations and a 0.135 MB model. LightGBM provided a slightly higher external F1 of 0.923 at 10.99 ms per 1,000 observations and 0.485 MB. Logistic Regression

was the smallest and fastest model (0.003 MB; 4.76 ms per 1,000) but with lower mean external F1 (0.894). The compact MLP was dominated by Logistic Regression in the external-F1/latency/model-size space. Under unweighted Pareto dominance, Logistic Regression, XGBoost, LightGBM, and Random Forest remained legitimate candidates because each represented a different accuracy-cost operating point. This result illustrates why deployment selection should not be reduced to the highest F1 alone.

Table 8. Resource-efficiency benchmark on the reference environment.

Model	Ext. F1	Train (s)	Size (MB)	Latency ms/1000	Throughput/s	Pareto
Logistic Regression	0.894	0.014	0.003	4.76	210,052	Yes
Random Forest	0.926	0.578	1.415	124.92	8,005	Yes
XGBoost	0.918	0.226	0.135	7.84	127,613	Yes
LightGBM	0.923	0.308	0.485	10.99	90,962	Yes
Compact MLP	0.887	0.344	0.096	5.36	186,430	No

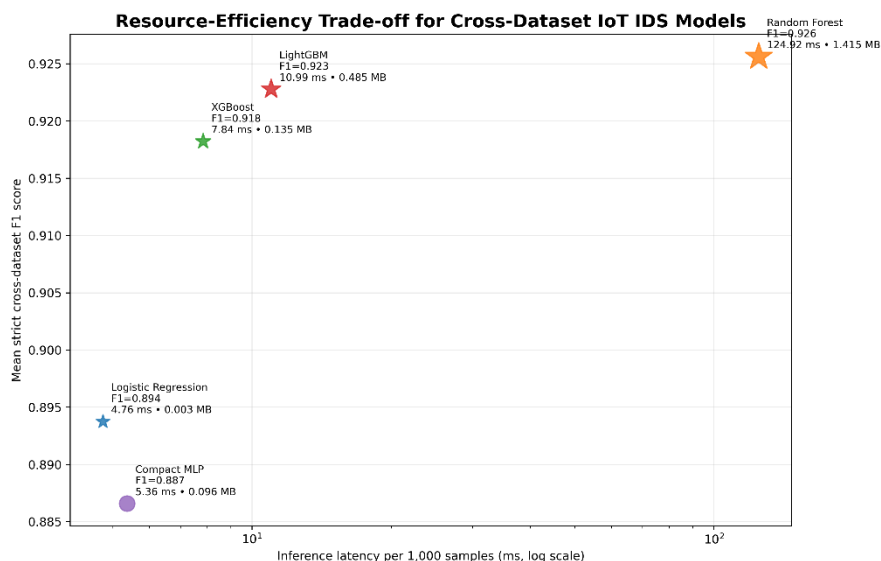


Figure 7. Cross-dataset accuracy-efficiency trade-off; marker size reflects serialized model size and stars identify Pareto-efficient models.

5.7 Implications and threats to validity

Taken together, the results support the paper's three-axis argument. Cross-dataset transfer remained strong for the harmonized binary task, SHAP rankings were stable across target sources, and no single classifier dominated all deployment costs. At the same time, the results expose two important boundaries. First, the near-chance source-domain classifier indicates that the harmonized study file contains limited residual dataset identity in the retained predictors; performance on the complete raw benchmark distributions may therefore exhibit larger transfer gaps. Second, multiclass attack-family transfer remained only moderate, showing that semantic generalization is harder than binary detection. Additional limitations include the use of one reference CPU environment rather than physical edge hardware, the absence of direct energy instrumentation, and the fact that the analysis file contains 600 harmonized observations per source rather than every raw record from the original public benchmarks. These limitations do not invalidate the framework, but they define the scope within which the empirical results should be interpreted.

6. Conclusion and Future Work

AI-based intrusion detection for smart infrastructure should be evaluated on more than benchmark accuracy. In this study, the proposed framework was tested on 1,800 harmonized observations drawn equally from CICIOT2023, Edge-IIoTset, and TON_IoT. Random Forest delivered the strongest strict source-to-target F1 (mean 0.926), while LightGBM led the leave-one-dataset-out setting (mean F1 0.937). SHAP rankings for Random Forest were highly stable across unseen targets, with mean top-10 Jaccard overlap of 0.818 and mean Spearman rank correlation of 0.990. Resource measurements showed that this robustness came at a computational cost: Random Forest was much slower and larger than XGBoost, LightGBM, and Logistic Regression, leaving several models on the Pareto frontier rather than one universally optimal detector.

The findings support a deployment philosophy in

which external transfer, explanation stability, and efficiency are evaluated together. They also show that strong binary transfer does not solve the harder problem of attack-family generalization, for which mean LODO macro-F1 was only 0.527. Future work should replicate the protocol on the complete raw versions of the three benchmark datasets, test additional unseen datasets and concept drift, measure energy and peak memory on physical edge platforms, and evaluate adaptive or federated variants under the same three-axis criteria. For safety-relevant IoT infrastructure, those extensions are necessary before a model can be regarded as operationally portable rather than merely accurate on a harmonized benchmark sample.

Data Availability

The empirical analysis used a harmonized study file containing observations labeled as originating from CICIOT2023, Edge-IIoTset, and TON_IoT. The public benchmark datasets remain subject to the licenses and access conditions of their original providers. To support reproducibility, the analysis workflow should be archived with the exact harmonized file version, feature list, label mapping, random seed, and software environment used in this study.

References

- Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. N., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8, 165130–165150. <https://doi.org/10.1109/ACCESS.2020.3022862>
- Baghalzadeh, H., Derakhshanfard, N., Kargar, M., & Ghaffari, A. (2026). A resource-aware framework for energy and bandwidth efficient IoT intrusion detection using lightweight federated learning over SDN. *Scientific Reports*. <https://doi.org/10.1038/s41598-026-74166-3>
- Booij, T. M., Chiscop, I., Meeuwissen, E.,

- Moustafa, N., & den Hartog, F. T. H. (2022). ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal*, 9(1), 485–496. <https://doi.org/10.1109/JIOT.2021.3085194>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. <https://doi.org/10.1023/A:1010933404324>
- Cerasuolo, F., Bovenzi, G., Ciuonzo, D., & Pescapé, A. (2025). Adaptable, incremental, and explainable network intrusion detection systems for internet of things. *Engineering Applications of Artificial Intelligence*, 144, 110143. <https://doi.org/10.1016/j.engappai.2025.110143>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Eren, K. K., & Küçük, K. (2026). Distributional drift in IoT intrusion detection systems: Implications for cross-dataset generalisation. *Electronics*, 15(11), 2307. <https://doi.org/10.3390/electronics15112307>
- Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10, 40281–40306. <https://doi.org/10.1109/ACCESS.2022.3165809>
- Hasan, T., & Tasnim, S. (2025). Real-time explainable IoT security with machine learning and CTGAN-enhanced detection for resource-constrained devices. *Ad Hoc Networks*, 178, 103937. <https://doi.org/10.1016/j.adhoc.2025.103937>
- Holubenko, V., Gaspar, D., Leal, R., et al. (2025). Autonomous intrusion detection for IoT: A decentralized and privacy preserving approach. *International Journal of Information Security*, 24, 7. <https://doi.org/10.1007/s10207-024-00926-9>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.
- Megas, K., Cuthill, B., Fagan, M., & Watrobski, P. (2024). NIST Cybersecurity for IoT Program. *Computer*, 57(12). <https://doi.org/10.1109/MC.2024.3468929>
- Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72, 102994. <https://doi.org/10.1016/j.scs.2021.102994>
- Nassef, L., Alghamdi, M. I., Ben Chaabane, S., Abbas, Q., Alawad, W. M., Albalawi, O. H., Alqaisi, O. I., & Fakieh, B. (2026). Lightweight and energy-aware intrusion detection for industrial IoT using TinyML and edge AI. *Scientific Reports*, 16, 18524. <https://doi.org/10.1038/s41598-026-50690-0>
- Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, 23(13), 5941. <https://doi.org/10.3390/s23135941>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why should I trust you?': Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and*

- Data Mining, 1135–1144.
<https://doi.org/10.1145/2939672.2939778>
- Rodríguez, M., Mehavilla, L., Alesanco, Á., & García, J. (2026). A rule-based label harmonization framework for cross-dataset IoT intrusion detection. *Internet of Things*, 39, 102030.
<https://doi.org/10.1016/j.iot.2026.102030>
- Subramanian N., S. S., P., P., Jain, K., & Krishnan, P. (2026). Edge AI Bridge: A micro-layer intrusion detection architecture for smart-city IoT networks. *IoT*, 7(2), 33.
<https://doi.org/10.3390/iot7020033>
- Udurume, M., Shakhov, V., & Koo, I. (2026). Enhancing IoT network security with explainable deep learning-based intrusion detection systems. *Scientific Reports*.
<https://doi.org/10.1038/s41598-025-34334-3>
- Varol, M., & Karakaya, A. (2026). Explainable AI-based intrusion detection systems for IoT environments: A systematic literature review. *Sensors*, 26(18), 5744.
<https://doi.org/10.3390/s26185744>
- Villafranca, A., Thant, K. M., Tasic, I., & Cano, M.-D. (2025). AI-enabled IoT intrusion detection: Unified conceptual framework and research roadmap. *Machine Learning and Knowledge Extraction*, 7(4), 115.
<https://doi.org/10.3390/make7040115>
- Wang, Y., Azad, M. A., Zafar, M., & Gul, A. (2025). Enhancing AI transparency in IoT intrusion detection using explainable AI techniques. *Internet of Things*, 33, 101714.
<https://doi.org/10.1016/j.iot.2025.101714>