This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



# RFS: Hardware-Adaptive Risk Assessment for IoT-Enabled Sensitive Data Systems Using Heterogeneous Machine Learning Models

Darlan Noetzold, *(Member, IEEE),* Jorge L. V. Barbosa, Juan F. de Paz, and Valderi R. Q. Leithardt, *(Senior Member, IEEE)*

***Abstract*—IoT-enabled sensitive data systems require risk assessment that maintains accuracy and calibration stability from cloud servers to resource-constrained endpoints. Yet many IoT risk-scoring pipelines leave unresolved the gaps in leakage-safe preprocessing, calibration-consistent decision policies, and reproducible training-inference equivalence across deployment tiers. This paper presents Risk Factor Service (RFS), a Q-OPSEC module that binds categorical encoding, normalization, and feature selection into versioned artifacts, evaluates six model families with per-model calibration and threshold optimization, and enables hardware-adaptive deployment from GPU servers to ESP32-S3 microcontrollers. Evaluation on a synthetic imbalanced dataset and the public NSL-KDD benchmark yields MCC above 0.83 and F1 above 0.85 on synthetic data (AUC-ROC $>$ 0.95 and F1 $>$ 0.80 on NSL-KDD under distribution shift), with server latency below 13 ms and energy efficiency of 21.3 samples/J on IoT endpoints. Results constitute proof-of-concept evidence under controlled benchmarks rather than full operational deployment.**

***Index Terms*—Edge Computing, Feature Engineering, Hardware-Adaptive, Internet of Things, Machine Learning Models, Risk Assessment, Threshold Optimization.**

## I. Introduction

The proliferation of Internet of Things (IoT) devices across healthcare, access control, industrial monitoring, and financial services generates high-volume event streams that demand real-time risk assessment at multiple points in the network hierarchy. These IoT-enabled systems combine mixed feature types from heterogeneous sensors, evolving behavioral patterns, and strict latency and energy constraints that rule out centralized-only processing. Risk assessment is therefore a core cybersecurity function for IoT because it prioritizes response to high-impact threats such as unauthorized access escalation, anomalous device behavior, and adaptive multi-stage network attacks in resource-constrained environments [35], [36]. Deploying machine learning models across the IoT continuum, from cloud servers to edge gateways to microcontroller endpoints, demands stable preprocessing contracts, explicit artifact versioning, and consistent inference semantics across hardware tiers. The gap between offline training and production serving across heterogeneous IoT infrastructure remains a recognized barrier to reproducible risk scoring in security-critical environments [1]–[3].

Risk Factor Service (RFS) fills this gap through a versioned inference pipeline that binds preprocessing transforms, model artifacts, and calibration parameters into immutable bundles, ensuring training-inference semantic equivalence regardless of the target hardware. RFS is the risk assessment module of the Q-OPSEC research line [4], [5], targeting IoT-enabled sensitive data systems that require calibrated scoring and deployment spanning cloud, edge, and constrained IoT nodes.

The scientific contribution operates across three dimensions. First, a leakage-aware preprocessing contract integrates categorical encoding, z-score normalization, and mutual-information-based feature selection into versioned artifacts, preventing test-set leakage and enabling deterministic inference across deployment tiers. Second, the training orchestrator evaluates six heterogeneous model families under a unified protocol with per-model threshold optimization that decouples probability estimation from decision boundaries. Third, the serving architecture implements version-resolved artifact loading with model-specific adapters, enabling hardware-adaptive deployment from GPU servers to resource-constrained IoT endpoints such as ESP32-S3 microcontrollers, with calibration reported at two distinct levels: model-family calibration in non-quantized analysis and deployment-level calibration for the quantized endpoint profile.

On a 1:11 imbalanced risk dataset, gradient-boosted models reach Matthews Correlation Coefficient above 0.83 and F1 above 0.85, while hardware profiling yields 80.6 samples/s on servers, 11.6 samples/s on edge gateways, and energy efficiency of 21.3 samples/J on ESP32-S3 IoT endpoints. External validation on the public NSL-KDD intrusion-detection benchmark confirms pipeline portability, with the best models achieving AUC-ROC above 0.95 and F1 above 0.80 under distribution shift. These quantitative results provide proof-of-concept evidence across performance, calibration, and deployment dimensions, but do not constitute field operational validation under real temporal drift, adversarial traffic, or real-user behavior.

The main research objectives are: (1) define a leakage-safe preprocessing and artifact-governance contract that preserves training-inference semantic equivalence across cloud, edge, and IoT tiers; (2) evaluate heterogeneous model families under a unified calibration and threshold-selection protocol for imbalanced risk scoring; and (3) validate cross-tier deployment

D. Noetzold and J. F. de Paz are with the Expert Systems and Applications Laboratory (ESALAB), University of Salamanca, Salamanca, Spain (e-mail: darlan.noetzold@usal.es).

J. L. V. Barbosa is with the University of Vale do Rio dos Sinos (UNISINOS), PPGCA, São Leopoldo, Brazil (e-mail: jbarbosa@unisinos.br).

V. R. Q. Leithardt is with the University Institute of Lisbon (ISCTE-IUL), ISTAR, Lisbon, 1649-026, Portugal, with the Expert Systems and Applications Laboratory (ESALAB), University of Salamanca, Salamanca, Spain, and also with the Center of Technology and Systems (CTS-UNINOVA), Associated Laboratory of Intelligent Systems (LASI), 2829-516 Lisbon, Portugal (e-mail: valderi.leithardt@iscte-iul.pt).

This work was funded by the University of Vale do Rio dos Sinos (UNISINOS).

Corresponding author: Jorge L. V. Barbosa (e-mail: jbarbosa@unisinos.br).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



feasibility through quantitative discrimination, calibration, latency, and energy measurements, complemented by external benchmarking on NSL-KDD.

Section II reviews related work; Section III details the architecture, including data generation, feature engineering, model-family orchestration, and versioned artifact management; Section IV reports discriminative performance, calibration, external validation, ablations, and deployment efficiency; Section V concludes the study.

## II. Related Works

Prior work follows three main directions: self-adaptive optimization in smart environments and IoT ecosystems, security-oriented middleware for sensitive or post-quantum environments, and machine-learning-based security analysis targeting IoT and edge architectures. Table I compares selected prior works against RFS along primary focus, main mechanisms, and their relationship to the proposed module. Oraculum [4] established the earlier performance-centered line of research by targeting self-adaptive optimization in smart environments. Q-OPSEC extended this line toward adaptive AI middleware for quantum cryptography, and DN25 [5] introduced the protocol layer that fits into that middleware stack.

Recent studies also explore security and privacy functions through machine learning and adaptive control in IoT contexts. Latif et al. [6] proposed a blockchain-based authentication system for e-health with multi-factor authentication, access control, and post-quantum security. Mumtaz et al. [7] studied AI-based classification for post-quantum lightweight cryptography primitives. Ghosh et al. [8] presented a federated learning framework for adaptive security in edge-driven UAV networks. Holgado et al. [9] combined context awareness and post-quantum cryptography for IoT networks. Nallapareddy [10] examined machine learning for adaptive cybersecurity in IoT systems. Xu et al. [11] proposed an adaptive secure data sharing system with verifiable outsourced decryption. Alsalim [12] studied anomaly detection for enterprise storage security. Huang et al. [21] combined large language models and post-quantum-secure blockchain for IoT intrusion detection. In the same period, Haider et al. [37] surveyed anomaly detection techniques and 6G-integrated IoT security challenges, Aparcana-Tasayco et al. [38] systematically reviewed quantum machine learning for IoT anomaly detection, and Aly and Behiry [39] compared boosting and ensemble methods for anomaly detection in IoT-driven industrial environments. More recently, Haider et al. [22] reviewed resource-efficient cryptographic mechanisms for IoT security, and Wang et al. [29] proposed a privacy-preserving cloud auditing architecture for Medical IoT data governance.

These works share one or more aspects with RFS, yet they do not address the fundamental challenge of training-inference semantic equivalence under heterogeneous model deployment across the IoT hardware continuum. Existing approaches either assume static preprocessing without versioning guarantees, apply single-model architectures that bypass calibration-consistency requirements, or omit the artifact-governance layer necessary for audit trails when deploying from cloud servers to resource-constrained IoT endpoints. RFS contributes a formal preprocessing contract that binds categorical encoders, numeric scalers, and feature selectors into immutable versioned bundles, preventing test-set leakage and ensuring deterministic inference across model families and hardware configurations, from data-center GPUs to edge gateways and microcontroller-based IoT nodes. An explicit distinction separates engineering and machine-learning novelty: model registries, immutable preprocessing objects, experiment tracking, and deployment adapters are established MLOps practices, whereas the scientific novelty in RFS lies in their constrained coupling with leakage-safe fitting, per-model calibration/threshold governance, and cross-tier semantic invariance checks within a single risk-scoring protocol. Regarding computational efficiency, RFS currently emphasizes compact model families and quantized deployment profiles; complementary surrogate-assisted and active-learning strategies can further reduce expensive training/evaluation cycles and are an immediate methodological extension path [33], [34].

In summary, the research gap addressed by RFS concerns the absence of a unified framework that simultaneously enforces leakage-safe preprocessing, maintains calibration-consistent decision policies across heterogeneous model families, and guarantees reproducible training-inference semantic equivalence when deploying from cloud-tier GPU servers to resource-constrained IoT microcontroller endpoints. Prior work addresses subsets of these requirements in isolation, but no existing approach integrates them within a single versioned artifact-governance protocol that remains operational across the full cloud-edge-IoT continuum under imbalanced risk-scoring tasks.

## III. Methodology

RFS organizes the pipeline into two functional layers: Fig. 1 presents data generation, training orchestration, and serving, while Fig. 2 details feature engineering, model-family orchestration, and inference. Both share a common artifact registry that binds training-time transforms to inference-time adapters.[1]

### A. Data Generation, Training, and Versioning

The top layer of Fig. 1 covers controlled data generation.[2] Four generators operate under a shared Configs object: UserGenerator produces user profiles with demographic and behavioral attributes; EventGenerator produces timestamped interaction events; RecipientGenerator produces counterpart entities for each event; and LabelGenerator assigns binary risk labels according to configurable rules. These generators emit raw events that feed eleven feature blocks: device, temporal, transaction, location, network, fraud_history, security, text, user_account, behavior, and an optional LLM block for embedding-based signals.

[1]RFS implementation and source code are publicly available at https://github.com/DarlanNoetzold/Q-OPSEC

[2]The evaluation dataset generated by this pipeline is available at Mendeley Data: https://data.mendeley.com/datasets/crgh2ynmzv/2

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



TABLE I
COMPARISON OF SELECTED WORKS AND RFS

| Work | Focus | Main mechanism | Relation to Risk Factor Service |
|---|---|---|---|
| Noetzold et al. [4] | Self-adaptive optimization in smart environments | Runtime adaptation for performance tuning | Establishes the earlier performance-centered line of research. |
| Latif et al. [6] | E-health authentication | Blockchain, multi-factor authentication, access control, and post-quantum security | Addresses secure access control, but not module-level risk scoring or model serving. |
| Mumtaz et al. [7] | Post-quantum cryptography analysis | AI-based security classification of lightweight cryptographic structures | Focuses on primitive analysis, not on inference contracts or deployment. |
| Ghosh et al. [8] | Adaptive security in UAV networks | Federated learning for edge-driven systems | Supports distributed learning, but does not provide a versioned serving pipeline. |
| Holgado et al. [9] | Context-aware IoT security | Context-aware security with post-quantum cryptography | Addresses secure IoT design, but not heterogeneous model-family orchestration. |
| Nallapareddy [10] | Adaptive cybersecurity for IoT | Machine learning-based detection and adaptation | Focuses on detection, but not on calibrated multi-model inference. |
| Xu et al. [11] | Secure data sharing | Verifiable outsourced decryption and adaptive access control | Targets secure sharing, but not risk scoring or feature pipeline alignment. |
| Alsalim [12] | Enterprise storage protection | Efficient anomaly detection | Covers detection in storage systems, but not training-inference versioning. |
| Huang et al. [21] | IoT intrusion detection | Large language models and post-quantum-secure blockchain | Uses advanced detection, but not a module-level serving architecture. |
| RFS | Risk assessment in IoT-enabled sensitive data systems | Leakage-aware preprocessing, heterogeneous model families, per-model calibration, versioned artifacts, and hardware-adaptive IoT deployment | Integrates the module-level risk pipeline inside a Q-OPSEC deployment across cloud, edge, and IoT tiers. |

LabelGenerator operates through a composite scoring function rather than a single deterministic rule, computing weighted risk contributions from multiple feature blocks: transactional anomalies (e.g., `amount_increase_vs_30d_mean` exceeding configurable thresholds), behavioral deviations (e.g., `transactions_last_24h` outside historical norms), network and device risk indicators (e.g., `ip_on_blacklist`, `is_vpn_detected`, `is_jailbroken_or_rooted`), location inconsistencies (e.g., `country_change_since_last_session`, `distance_km_from_last_location`), authentication failures (e.g., `failed_logins_24h`, `tls_weak_cipher`), and historical fraud signals (e.g., `user_has_fraud_history`, `user_global_risk_score`).

These per-block scores are combined into a continuous aggregate risk score using configurable weights per block, and the binary label `label_fraud` is assigned by thresholding this aggregate. Controlled stochastic noise governed by a `label_confidence` parameter (range 0–1) introduces label flips proportional to $(1 - \text{label_confidence})$, preventing a deterministic feature-to-label mapping. The ground-truth schema also records `label_source` (one of `rule_engine`, `anomaly_model`, `confirmed_incident`, or `customer_dispute`) and `label_risk_level` (0=low through 3=critical), and the evaluation configuration uses a mixture of these provenance types with varying confidence levels.

While the evaluation dataset is synthetic, the machine-learning models do not receive the raw generator inputs directly. Instead, the feature engineering pipeline computes derived features (velocity scores, percentile ranks, temporal aggregations, cyclical time encodings) that introduce non-linear transformations not present in the label-generation rules. The LLM-derived feature block (`llm_risk_score`, `llm_phishing_score`, `llm_social_engineering_score`) further expands the input space with signals produced by an independent language model, whose internal representations are not encoded in the label-generation logic. Hence, reported performance reflects the models' capacity to approximate a noisy, multi-factor scoring function from engineered features rather than recovery of a deterministic rule set; the synthetic evaluation is a controlled proof-of-concept, not an operational deployment assessment, and claims are moderated accordingly.

Dataset partitioning operates at the *user level* rather than the event level. The generator produces 250,000 events from 20,000 synthetic user profiles, and the DataLoader assigns each `user_id` (together with all of its events, devices, and recipient interactions) to exactly one partition using a group-aware stratified split (60%/20%/20% for training, validation, and test), stratified on the event-level risk label so that the 1:11 class ratio is preserved in every partition. This yields a held-out test partition of 50,000 events (4,100 risk and 45,900 legitimate). Grouping by user guarantees that no user, device fingerprint, or behavioral profile is shared across partitions, eliminating entity-level leakage. Since events are timestamped, the validation and test partitions are additionally restricted to events whose timestamps are later than the last training event of the same simulation run, so that no temporal aggregations (e.g., `transactions_last_24h`) in the test set are computed from training-period activity. All splits use a fixed random seed (42), recorded in the artifact metadata alongside the dataset hash. FeatureEngineer applies the preprocessing contract (next subsection) to the splits, and ModelTrainer coordinates the six families instantiated by ModelFactory. Each model minimizes a task-specific loss $\mathcal{L}$; for binary risk classification, the cross-entropy loss over training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{p}_i + (1 - y_i)\log(1 - \hat{p}_i)\right] \qquad (1)$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



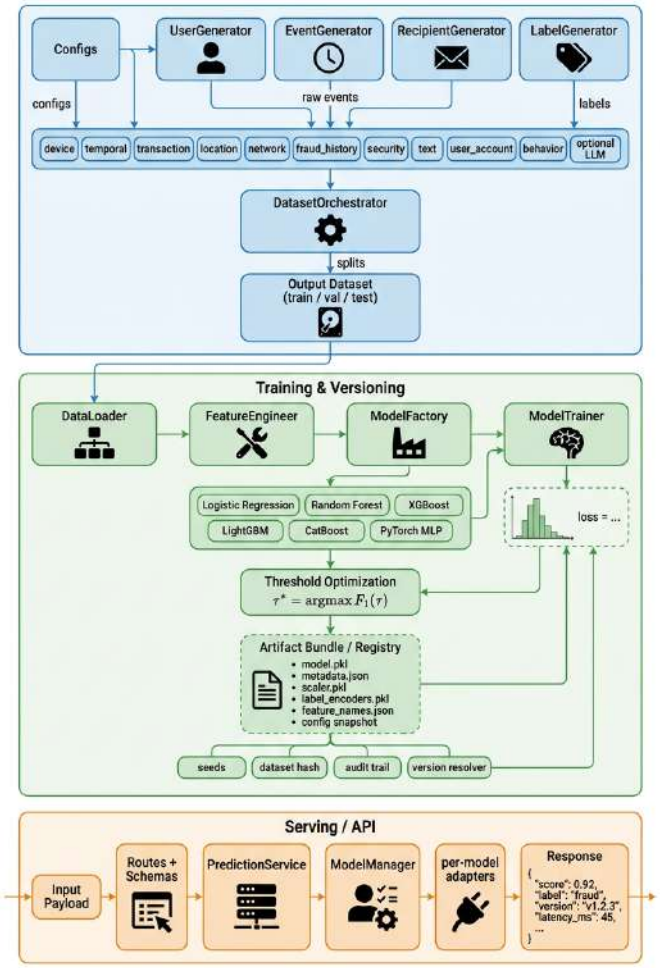



Fig. 1. Module overview of RFS.

where $\hat{p}_i \in [0,1]$ denotes the predicted risk probability for sample $i$ and $y_i \in \{0,1\}$ is the ground-truth label. After training, each model undergoes Threshold Optimization. The optimal decision threshold $\tau^*$ maximizes the $F_1$ score over the validation set:

$$\tau^* = \arg\max_{\tau \in [0,1]} F_1(\tau) = \arg\max_{\tau} \frac{2 \cdot P(\tau) \cdot R(\tau)}{P(\tau) + R(\tau)} \tag{2}$$

where $P(\tau)$ and $R(\tau)$ denote precision and recall at threshold $\tau$, respectively. This per-model optimization decouples the scoring function from the decision boundary. All outputs feed the Artifact Bundle/Registry (serialized estimators, hyperparameters, metrics, version tags, fitted transformers and encoders, ordered feature contracts, and experiment configurations), while the governance band records seeds, dataset hash, audit trail, and version resolver so that any artifact version maps to a unique training run.

The Serving/API layer exposes the artifacts through a FastAPI service in which ModelManager resolves the requested version from the registry and per-model adapters apply the stored preprocessing contract before inference, as detailed below.

TABLE II
MODEL FAMILIES IN THE MODEL PORTFOLIO, DECISION FUNCTIONS, AND INDUCTIVE BIASES

| Model | Decision function | Inductive bias |
|---|---|---|
| Logistic Regression | $\hat{p} = \sigma(\mathbf{w}^\top \mathbf{x} + b)$ | Linear separability, calibrated probabilities |
| Random Forest | $\hat{p} = \frac{1}{T}\sum_{t=1}^{T} h_t(\mathbf{x})$ | Non-linear interactions, variance reduction |
| XGBoost | $\hat{p} = \sum_{k=1}^{K} f_k(\mathbf{x})$ | Gradient boosting, regularized trees |
| LightGBM | $\hat{p} = \sum_{k=1}^{K} f_k(\mathbf{x})$ | Histogram-based boosting, leaf-wise growth |
| CatBoost | $\hat{p} = \sum_{k=1}^{K} f_k(\mathbf{x})$ | Ordered boosting, native categoricals |
| PyTorch MLP | $\hat{p} = \sigma(\mathbf{h}^{(L)})$ | Deep non-linear representations |

### *B. Feature Engineering, Model-Family Orchestration, and Inference*

Fig. 2 details feature engineering, model-family orchestration and training logic, and inference serving. Raw input passes through schema normalization to a canonical contract and then through three sequential transformations. Categorical encoding maps each nominal feature $x_j$ to either an integer label or a one-hot vector:

$$\text{label}(\mathbf{x}) \in \mathbb{Z}, \qquad \text{onehot}(\mathbf{x}) \in \{0,1\}^k \tag{3}$$

where $k$ is the cardinality of the categorical domain. The per-feature encoding strategy is stored in the categorical encoders so that inference applies identical mappings. Numeric scaling then standardizes each continuous feature using z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{4}$$

where $\mu$ and $\sigma$ denote the mean and standard deviation computed from the training partition exclusively. The fitted scaler is reloaded at inference time, preventing leakage of test or production statistics. Feature selection then applies mutual information between each feature $X_j$ and the target $Y$:

$$\text{MI}(\mathbf{X}; Y) = \sum_j I(X_j; Y) = \sum_j \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{5}$$

Features with $I(X_j; Y)$ below a configurable threshold are discarded, and the retained set and ordering are stored in the feature contract; encoded and scaled features are then concatenated into a dense vector $\mathbf{x} \in \mathbb{R}^n$ from the eleven domain blocks. A drift monitoring step applies Population Stability Index (PSI) or Kolmogorov-Smirnov (KS) tests to detect distributional shift between training and production data.

The assembled vector $\mathbf{x}$ enters the model family matrix, which instantiates six model families in parallel. Table II summarizes each family, its decision function, and primary inductive bias, spanning a linear baseline (Logistic Regression), bagged trees (Random Forest), gradient-boosted additive trees with distinct growing strategies (XGBoost, LightGBM) and ordered boosting with native categorical handling (CatBoost), and a feedforward neural network (PyTorch MLP) for higher-order feature interactions.

RFS does not implement prediction fusion (no voting, stacking, blending, or weighted averaging) in the current study. Instead, it follows a model-portfolio protocol: all model families are trained and evaluated under the same preprocessing/calibration contract, and deployment resolves one model

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



variant per request according to artifact version and hardware tier. The deployed decision is therefore single-model, given by Eq. 6.

After training, each model passes through a calibration step that aligns raw scores to well-calibrated probabilities using Platt scaling or isotonic regression. Calibrated scores then enter the threshold optimizer, which applies Eq. 2 over the validation partition to determine $\tau^*$ per model. The four data roles are therefore explicitly separated: the training partition fits the estimators and all preprocessing statistics; the validation partition fits the calibrator and selects $\tau^*$; the test partition serves exclusively for the final metrics and no fitting step touches it. Because the calibrator and the threshold are both fitted on the same validation partition, an experimental verification confirmed that this sharing does not inflate test performance by splitting the validation set into two disjoint stratified halves (one for calibration, one for threshold selection) and comparing held-out test metrics (Section IV-D). The evaluation protocol computes AUC-ROC, $F_1$, precision, recall, and Brier score and writes them to `metadata.json`.

At inference time, a request enters through `prepare_input(df)`, which normalizes and validates the incoming dataframe. Model-specific adapters load the artifact bundle for the requested version via `load(version)`, applying the stored scaler, encoders, and feature contract to the input. The adapted features pass to *PredictionService*, which calls the loaded estimator and obtains a raw prediction $\hat{p}$. Response mapping applies the stored threshold and returns the final binary label:

$$\hat{y} = \mathbf{1}[\hat{p} \geq \tau^*] \tag{6}$$

The response payload includes $\hat{p}$, $\hat{y}$, the model identifier, and the resolved version string. The infrastructure layer (persistent logs, model registry, feature store, experiment tracking) supports auditability and operational monitoring.

## IV. Results

This section reports the evaluation of RFS across three dimensions: discriminative performance across six model families, threshold calibration behavior, and deployment efficiency across heterogeneous hardware targets representative of the cloud-edge-IoT continuum. All models were assessed on a held-out test partition with a class imbalance of approximately 1:11 (no-skill AP $= 0.082$), with each metric computed per model at its optimized threshold $\tau^*$.

### A. Discriminative Performance and Decision Analysis

Fig. 3 presents three complementary views of model discrimination: the ROC curve, the precision-recall (PR) curve, and the detection error tradeoff (DET) curve.[3] The ROC curve measures the tradeoff between true positive rate (TPR) and false positive rate (FPR) across all thresholds, with area under the curve given by:

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t))\, dt \tag{7}$$

[3]More results can be viewed at: https://github.com/DarlanNoetzold/Q-OPSEC/tree/main/risk_service_V2

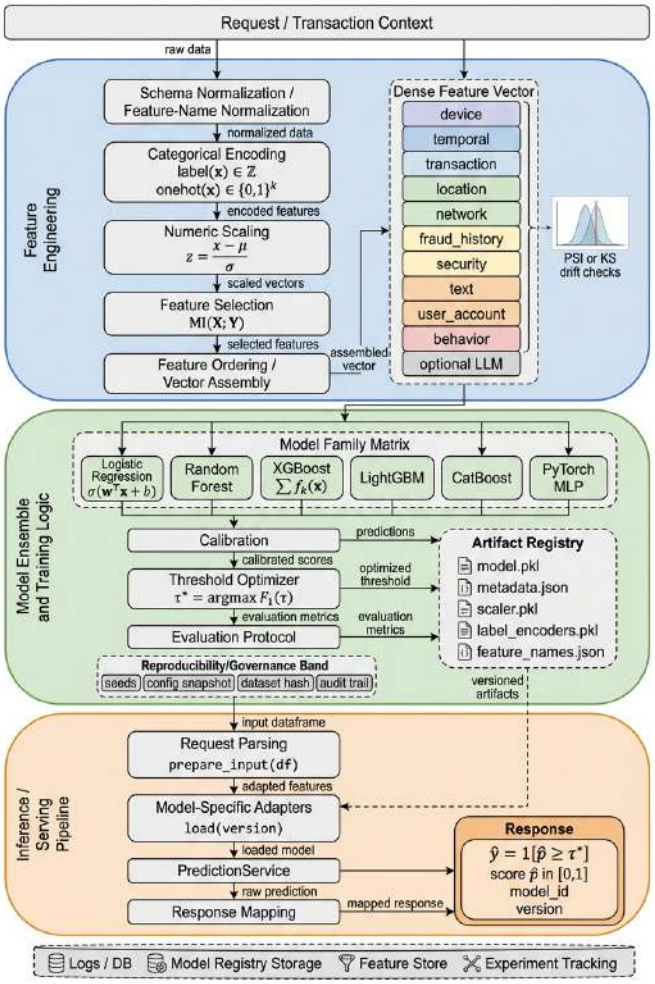



Fig. 2. Architecture of RFS.

LightGBM achieves the highest AUC-ROC of 0.9874, followed by CatBoost (0.9845), XGBoost (0.9754), PyTorch MLP (0.9562), Random Forest (0.9197), and Logistic Regression (0.7701). In the operationally critical low-FPR region (FPR $\leq 0.10$, ROC inset), gradient-boosted models maintain TPR above 0.75 while Logistic Regression drops below 0.60 TPR at FPR = 0.05, a divergence that the global AUC does not reveal.

The PR curve conditions precision on the positive class and is therefore more informative under severe imbalance. Average precision (AP) summarizes the area under the PR curve as:

$$\text{AP} = \sum_k (R_k - R_{k-1}) \cdot P_k \tag{8}$$

where $P_k$ and $R_k$ are precision and recall at the $k$-th threshold. LightGBM leads with AP $= 0.9041$, followed by CatBoost (0.8752), XGBoost (0.8432), PyTorch MLP (0.7528), Random Forest (0.6165), and Logistic Regression (0.2763), against a no-skill baseline of 0.082. At $R = 0.20$, LightGBM and CatBoost maintain precision above 0.95, while Random Forest drops below 0.85 and Logistic Regression below 0.50, indicating that boosted models assign high scores almost exclusively to true positives in the conservative regime relevant to alert quality.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



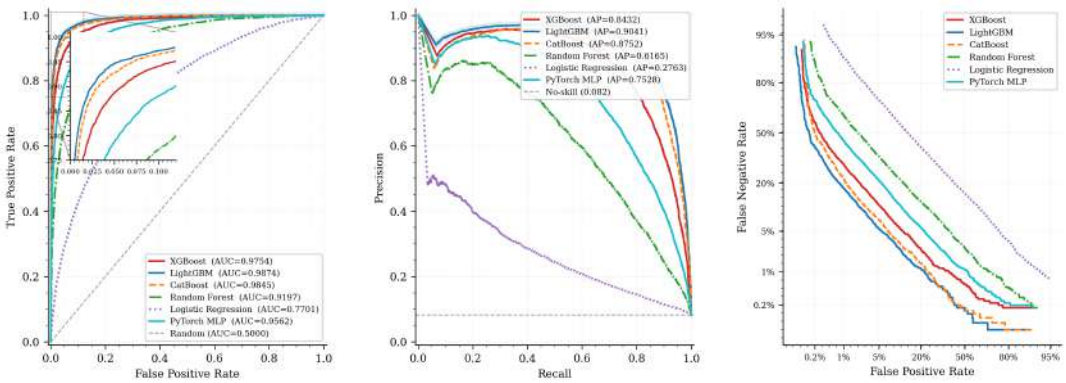



Fig. 3. ROC curve (left), precision-recall curve (center), and detection error tradeoff curve (right) for all six model families on the held-out test set. The ROC inset zooms into the low-FPR region ($\leq 0.10$).[3]

The DET curve (log-log FNR vs. FPR) shows XGBoost, LightGBM, and CatBoost achieving FNR below 5% at FPR below 1%, with PyTorch MLP at a moderate distance and Logistic Regression requiring FPR above 10% to reach FNR below 60%; only the boosted families satisfy strict simultaneous constraints on both error types.

At the optimized thresholds $\tau^*$, LightGBM achieves the highest MCC of 0.8376 with $F_1 = 0.8506$, identifying 3,429 of 4,100 positive cases (recall 83.63%) with 534 false positives out of 45,900 legitimate events (specificity 98.84%). CatBoost follows (MCC = 0.8191, $F_1 = 0.8338$, recall 82.98%, 658 false positives), then XGBoost (MCC = 0.7699, $F_1 = 0.7872$, recall 75.46%, 667 false positives), PyTorch MLP (MCC = 0.6766, $F_1 = 0.7003$, recall 66.10%, 930 false positives), and Random Forest (MCC = 0.5629, $F_1 = 0.5991$, recall 60.66%, 1,715 false positives). Logistic Regression is weakest (MCC = 0.2695, $F_1 = 0.3353$, recall 41.22%, 4,290 false positives): its linear boundary cannot separate the mixed-type feature space, and the false-positive volume would be operationally unacceptable. Cohen's kappa corroborates these rankings (LightGBM $\kappa = 0.8374$; Logistic Regression $\kappa = 0.2637$).

Fig. 4 presents the per-model threshold analysis, showing how $F_1$, precision, recall, specificity, and normalized cost vary as a function of the decision threshold $\tau \in [0, 1]$.[3] As $\tau$ increases, recall decreases while precision increases, and the $F_1$ peak defines $\tau^*$. The normalized cost curve follows:

$$C_{\text{norm}}(\tau) = \frac{c_{\text{FN}} \cdot \text{FN}(\tau) + c_{\text{FP}} \cdot \text{FP}(\tau)}{C_{\text{max}}} \tag{9}$$

where $c_{\text{FN}}$ and $c_{\text{FP}}$ are asymmetric misclassification costs and $C_{\text{max}}$ normalizes the curve to $[0, 1]$. In Fig. 4, the illustrative cost curve uses $c_{\text{FN}} : c_{\text{FP}} = 5 : 1$ to reflect a higher penalty for missed risk events. $F_1$ serves as the primary threshold criterion because it gave stable operating points across all six families under class imbalance and matched the study objective of balancing precision/recall for screening. To address operational alternatives, a direct comparison of threshold objectives on NSL-KDD (LightGBM, threshold selected on validation and evaluated on held-out test): $F_1$ and MCC selected the same threshold ($\tau$=0.251, test $F_1$=0.8057, MCC= 0.6651); expected-cost minimization with 5:1 and an application-specific utility selected $\tau$=0.154 (test $F_1$=0.8121, MCC= 0.6728, lower normalized cost 0.2649 vs 0.2730 at $F_1$-optimal $\tau$); and an FPR-constrained objective (FPR$\leq 1\%$ on validation) selected $\tau$=0.003, which increased test recall (0.803) but did not preserve the 1% bound under distribution shift (test FPR 3.76%). A sensitivity sweep over $c_{\text{FN}}$:$c_{\text{FP}} \in \{1{:}1, 2{:}1, 5{:}1, 10{:}1, 20{:}1\}$ on NSL-KDD showed that the excess normalized cost of the $F_1$-optimal threshold relative to the cost-optimal one grows monotonically with the ratio (LightGBM: from $1.8\times$ at 1:1 to $7.3\times$ at 20:1), with cost-optimal thresholds collapsing toward very low values ($\tau < 0.01$) as missed detections become dominant. These results show that $F_1$ is a practical default for discrimination-focused operation, but threshold policy should be re-optimized when explicit cost/utility or strict error constraints are required. Gradient-boosted models converge to $\tau^* \in [0.559, 0.598]$ (LightGBM 0.559, CatBoost 0.569, XGBoost 0.598) with broad, flat $F_1$ peaks, a desirable property for deployment stability across IoT tiers. Random Forest ($\tau^* = 0.614$) and PyTorch MLP ($\tau^* = 0.620$) exhibit narrower peaks, indicating greater threshold sensitivity, while Logistic Regression ($\tau^* = 0.633$, peak $F_1 = 0.335$) keeps an elevated cost curve across the entire range. Specificity exceeds 0.90 at $\tau^*$ for all families.

### B. Score Calibration and Feature Interaction Structure

Fig. 5 presents the score distribution and calibration curves for all six models. Each panel overlays kernel density estimates of predicted scores per class, the empirical calibration curve, and $\tau^*$. The Kolmogorov-Smirnov (KS) statistic quantifies class separation:

$$\text{KS} = \sup_{\hat{p} \in [0,1]} |F_{\text{neg}}(\hat{p}) - F_{\text{pos}}(\hat{p})| \tag{10}$$

where $F_{\text{neg}}$ and $F_{\text{pos}}$ are the empirical CDFs of predicted scores for the negative and positive classes. The Brier score measures calibration quality as the mean squared error between predicted probabilities and labels:

$$\text{BS} = \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - y_i)^2 \tag{11}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



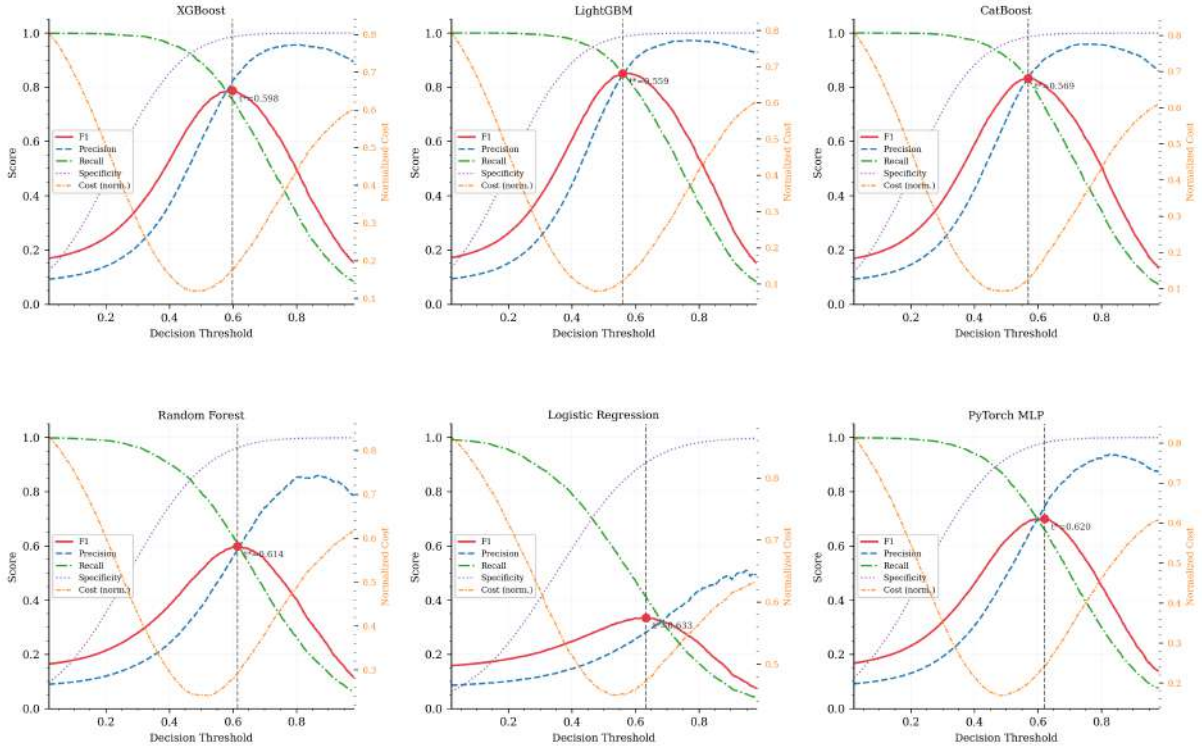



Fig. 4. Per-model threshold analysis showing $F_1$, precision, recall, specificity, and normalized cost as functions of $\tau \in [0, 1]$. Red dots mark the optimal $\tau^*$ for each model family.[3]

These Brier values refer to model-level (non-quantized) calibration curves in Fig. 5; the hardware subsection reports deployment-level calibration for the quantized ESP32-S3 profile separately through FP32-vs-quantized endpoint metrics. LightGBM achieves the highest KS of 0.8971 with Brier = 0.0686 (clearly bimodal scores, calibration curve close to the diagonal), followed by CatBoost (KS = 0.8804, Brier = 0.0727) and XGBoost (KS = 0.8469, Brier = 0.0804), whose broader risk-class distribution explains its lower recall. PyTorch MLP (KS = 0.7815, Brier = 0.0939) shows mild overconfidence at intermediate scores, Random Forest (KS = 0.6889, Brier = 0.1141) presents visible overlap in $[0.30, 0.70]$, and Logistic Regression (KS = 0.4051, Brier = 0.1733) shows the weakest separation. In all cases $\tau^*$ falls where the class densities cross.

Fig. 6 presents the feature interaction network derived from the multi-model analysis. Nodes represent features retained after mutual information selection, node size encodes mean normalized permutation importance aggregated across all six model families, and edges encode pairwise Spearman rank correlations $\rho$ with $|\rho| > 0.15$. Red/blue edges indicate positive/negative correlations, with thickness proportional to $|\rho|$:

$$G = (V, E), \quad E = \{(i, j) : |\rho(X_i, X_j)| > 0.15\} \qquad (12)$$

where $V$ is the set of selected features and $\rho(X_i, X_j)$ is the Spearman rank correlation between features $i$ and $j$ computed on the training partition. To ensure cross-family comparability, node size is not based on native model-specific importance scores. Instead, for each model family $m$, the validation-set permutation importance per feature $j$ equals $\Delta_{m,j} = F1_m(\mathbf{X}_{\text{val}}) - F1_m(\pi_j(\mathbf{X}_{\text{val}}))$, where $\pi_j$ denotes random permutation of feature $j$. Normalization within each model yields $\tilde{\Delta}_{m,j} = \Delta_{m,j} / \sum_k \Delta_{m,k}$, and the node size becomes the cross-model mean $s_j = \frac{1}{6} \sum_m \tilde{\Delta}_{m,j}$. The *llm_risk_score* node is the largest in the graph, indicating the highest mean normalized permutation importance in this synthetic benchmark rather than a universal causal ranking, with strong positive correlations to *velocity_score*, *country_risk*, and *failed_logins_24h*. The *llm_phishing* node is second largest, negatively correlated with *tls_version* and *mfa_used*; *country_risk* acts as a structural hub, whereas *device_age_days* and *hour_sin* show lower marginal importance and sparser connectivity. Because labels are generated by configurable synthetic rules with controlled noise, highly predictive features may partially reflect generator design choices; therefore, this analysis represents an internal consistency check of the simulation setting, not independent confirmation of real-world security causality.

### C. Deployment Efficiency Across IoT Hardware Tiers

Table III reports inference performance, resource utilization, and energy efficiency for RFS on three deployment tiers of the Q-OPSEC stack: a cloud server (Intel Xeon with NVIDIA T4 GPU), an edge gateway (Raspberry Pi 4), and a constrained IoT endpoint (ESP32-S3). Each tier runs a different model variant selected from the artifact registry based on available memory and compute budget: the cloud server runs the full six-model portfolio, the edge gateway runs the two highest-performing models, and the IoT endpoint runs a quantized

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



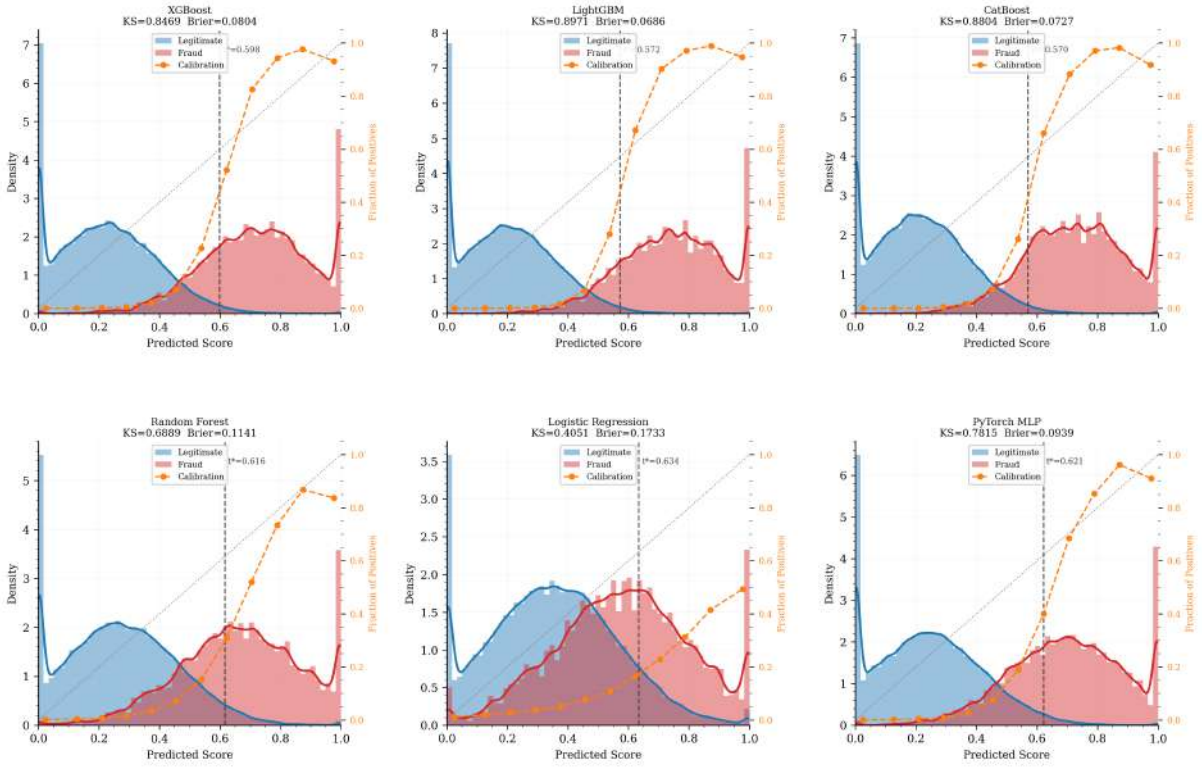



Fig. 5. Score distributions and calibration curves for all six models.[3]

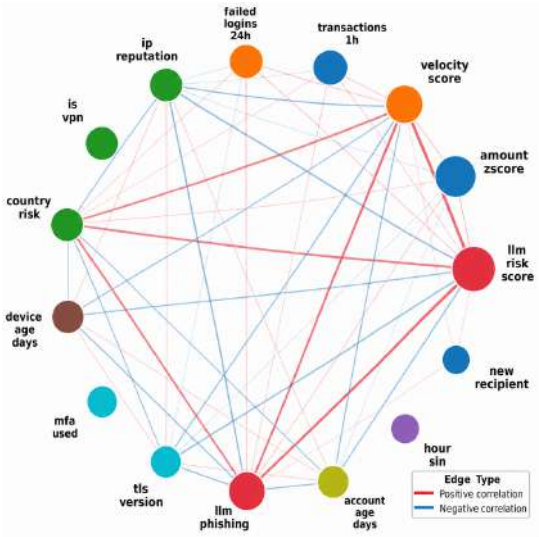



Fig. 6. Feature interaction network.

LightGBM variant with reduced precision. Accordingly, cross-tier differences in latency, memory, and energy reflect the joint effect of hardware and model complexity; readers should not interpret them as hardware-only effects. As a controlled reference, a common single-model anchor (LightGBM family) appears across all tiers through the deployed-variant AUC/$F_1$ row, while the adaptive-profile measurements remain the primary deployment result. For the ESP32-S3 column, memory values denote on-device runtime footprint (internal SRAM), while the 8 MB flash entry denotes storage capacity; host-side profiling memory is not mixed into these values. The following expressions define throughput $\Phi$ and energy efficiency $\eta$:

$$\Phi = \frac{1}{\bar{L}} = \frac{b}{\bar{T}_{\text{batch}}}, \bar{L} = \frac{\bar{T}_{\text{batch}}}{b}, \eta = \frac{\Phi}{P_{\text{active}}}, E = \frac{1}{\eta} \tag{13}$$

where $\bar{L}$ is the mean per-sample latency in seconds and $P_{\text{active}}$ is the active power consumption in watts. Here, $b$ denotes batch size and $\bar{T}_{\text{batch}}$ denotes mean measured batch execution time (s/batch). Benchmarking protocol clarification: latency corresponds to warm in-process end-to-end execution (input validation, stored preprocessing transforms, model inference, and threshold mapping), excluding external network/API transport and cold-start artifact loading. Power values correspond to steady-state active power during inference load; idle power appears separately and does not enter $\eta$ or $E$. Therefore, $\eta$ has units of samples/J, and $E$ is its inverse in J/sample. Clock-governor pinning, instrument sampling-rate harmonization, and per-run uncertainty traces were not retained in the archived logs for all tiers; consequently, Table III represents reproducible point estimates rather than full metrological confidence intervals. Measured mean batch execution times are 793.6 ms (cloud, $b = 64$), 1,371.2 ms (edge, $b = 16$), and 1,242.0 ms (ESP32-S3, $b = 4$); the reported throughput values are computed as $\Phi = b/\bar{T}_{\text{batch}}$ from these direct batch measurements. The cloud server achieves $\bar{L} = 12.4$ ms and $\Phi = 80.6$ samples/s (GPU utilization 64%, CPU 72% due to the parallel six-model portfolio); its 2,850 MB peak memory holds all serialized models and preprocessing contracts for zero-latency version switching.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



TABLE III
HARDWARE SPECIFICATIONS AND MEASURED DEPLOYMENT PERFORMANCE OF RFS ACROSS THREE IOT DEPLOYMENT TIERS. THESE MEASUREMENTS CORRESPOND TO HARDWARE-ADAPTIVE DEPLOYMENT PROFILES (DIFFERENT MODEL VARIANTS), NOT A CONTROLLED SAME-MODEL BENCHMARK. THROUGHPUT $\Phi$ AND ENERGY EFFICIENCY $\eta$ FOLLOW EQ. 13.

| Metric | Cloud Server (Xeon + T4 GPU) | Edge Gateway (RPi 4) | IoT Endpoint (ESP32-S3) |
|---|---|---|---|
| CPU Architecture | Intel Xeon (x86_64) @ 2.2 GHz | ARM Cortex-A72 @ 1.5 GHz | Xtensa LX7 @ 240 MHz |
| RAM / SRAM | 12.6 GB DDR4 | 4 GB LPDDR4 | 512 KB SRAM |
| Storage / Flash | 100 GB SSD | 32 GB microSD | 8 MB Flash |
| Operating System | Ubuntu 22.04 LTS | Raspberry Pi OS (64-bit) | FreeRTOS / ESP-IDF |
| Model variant | Full model portfolio (6 models) | LightGBM + CatBoost | Quantized LightGBM |
| **Inference Performance** | | | |
| Mean Latency $\bar{L}$ (ms/sample) | 12.4 | 85.7 | 310.5 |
| Throughput $\Phi$ (samples/s) | 80.6 | 11.6 | 3.2 |
| Batch size | 64 | 16 | 4 |
| Mean batch time $\bar{T}_{\text{batch}}$ (ms/batch) | 793.6 | 1,371.2 | 1,242.0 |
| **Resource Utilization** | | | |
| Peak Runtime Memory (MB) | 2,850 | 920 | 0.18 ($\approx$184 KB) |
| CPU Utilization (%) | 72 | 81 | 88 |
| GPU Utilization (%) | 64 | – | – |
| **Energy and Efficiency** | | | |
| Active Power $P_{\text{active}}$ (W) | 95.0 | 7.5 | 0.15 |
| Idle Power (W) | 45.0 | 2.8 | 0.04 |
| Energy per Inference $E$ (J) | 1.18 | 0.64 | 0.05 |
| Energy Efficiency $\eta$ (samples/J) | 0.85 | 1.55 | 21.3 |
| **Model Performance at Target** | | | |
| AUC-ROC (common LightGBM-family anchor) | 0.9874 (LightGBM) | 0.9874 (LightGBM) | 0.961 (Quantized LightGBM) |
| $F_1$ (at $\tau^*$, common LightGBM-family anchor) | 0.8506 | 0.8506 | 0.821 |
| Quantization loss $\Delta F_1$ | – | – | $-0.030$ |

The edge gateway operates at $\bar{L} = 85.7$ ms and $\Phi = 11.6$ samples/s, consuming 7.5 W and achieving energy efficiency of 1.55 samples/J, nearly double that of the server (0.85 samples/J) due to the lower power envelope of the ARM Cortex-A72 architecture. CPU utilization reaches 81% at batch size 16 (near the compute ceiling for the two-model configuration), and peak memory of 920 MB holds both LightGBM and CatBoost artifacts. Energy per inference of 0.64 J is 46% lower than the server (1.18 J), making the gateway suitable for power-constrained edge deployments requiring sub-100 ms latency.

The IoT endpoint, running a quantized model variant under FreeRTOS with the ESP-IDF framework, achieves $\bar{L} = 310.5$ ms with energy efficiency of 21.3 samples/J at 0.15 W active power. The corresponding energy per inference is 0.05 J (about $23\times$ lower than the cloud profile), but operational battery life depends on full-node duty cycle and non-inference loads (sensing, communication, memory, idle/sleep transitions). As a reference envelope, a 3.7 V/2000 mAh battery ($\approx$7.4 Wh) would sustain about 49 h at a hypothetical 100% inference-active 0.15 W regime, whereas event-driven low-duty operation extends this bound; therefore, this result constitutes proof of energy feasibility, not a complete lifetime claim. CPU utilization of 88% at batch size 4 reflects the limited parallelism of the single-core Xtensa LX7 architecture, and the reported peak runtime memory is 0.18 MB ($\approx$184 KB), not 180 MB. This value corresponds to on-device inference memory and is consistent with the 512 KB internal SRAM budget after quantization and feature-contract pruning. For reproducibility, the endpoint profile uses mixed precision: offline preprocessing remains in FP32, the deployed LightGBM tree traversal keeps FP32 thresholds/leaf values, and the calibrated posterior is serialized as an 8-bit lookup table for decision mapping. The conversion path exports the trained LightGBM booster to a static model artifact and packages it into the ESP32-S3 firmware image together with the calibration lookup and threshold metadata. The serialized endpoint artifact occupies about 2.86 MB in flash (model file 2,858,440 bytes + calibration/threshold metadata $< 1$ KB), which fits the 8 MB flash budget of ESP32-S3. The firmware was built with ESP-IDF (FreeRTOS target) using size-oriented optimization ($-Os$). This latency suits asynchronous scoring of low-frequency IoT events (e.g., access-control decisions or periodic telemetry) but not real-time stream processing. To quantify calibration drift under endpoint quantization, an emulation of the deployed 8-bit posterior table on NSL-KDD yielded: Brier changed from 0.1835 (FP32) to 0.1833 (quantized), ECE (10 bins) from 0.1911 to 0.1910, AUC-ROC by $-5.6 \times 10^{-6}$, and $F_1$ remained 0.8029. Artifact versioning enables this hardware-adaptive deployment by resolving a separate bundle per model variant at load time without retraining.

### D. External Validation on NSL-KDD

To assess pipeline portability beyond the synthetic dataset, this study applied the complete RFS preprocessing and training protocol to the public NSL-KDD intrusion-detection benchmark [32]. KDDTrain+ was split 80/20 stratified into train and validation partitions; KDDTest+ served as the held-out test set, which contains attack types absent from training, inducing a distribution shift that tests generalization. The leakage-aware contract was applied identically: label encoding, z-score normalization (statistics from training only), and mutual-information feature selection (top 30 of 41 features, computed on training only). Calibration (isotonic regression) and threshold optimization ($F_1$-maximizing) used the validation partition. Table IV reports held-out test results.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



TABLE IV
RFS PIPELINE RESULTS ON NSL-KDD (HELD-OUT KDDTEST+, $n$=22,544, ATTACK SHARE 56.9%).

| Model | AUC | $F_1$ | Prec. | Rec. | MCC | Brier |
|---|---|---|---|---|---|---|
| Log. Reg. | 0.905 | 0.744 | 0.925 | 0.623 | 0.566 | 0.199 |
| Rand. Forest | 0.952 | 0.797 | 0.967 | 0.677 | 0.654 | 0.180 |
| XGBoost | 0.940 | 0.771 | 0.968 | 0.641 | 0.626 | 0.193 |
| LightGBM | 0.942 | 0.803 | 0.967 | 0.686 | 0.662 | 0.184 |
| CatBoost | 0.921 | 0.797 | 0.968 | 0.678 | 0.656 | 0.184 |
| MLP | 0.948 | 0.761 | 0.974 | 0.625 | 0.619 | 0.200 |

Gradient-boosted models and Random Forest achieve AUC-ROC between 0.92 and 0.95 with precision consistently above 0.96, confirming that the preprocessing contract and calibration protocol transfer effectively to an independent dataset with different feature semantics and class distribution. The recall reduction relative to the synthetic evaluation (from $\sim$0.84 to $\sim$0.68 for LightGBM) is attributable to the 17 novel attack types in KDDTest+ that are absent from training, representing a realistic distribution shift. Despite this shift, Brier scores remain below 0.20 for all boosted/tree models, indicating that calibration quality is preserved. These results demonstrate that the RFS pipeline is not coupled to the synthetic data generator and generalizes to an established public benchmark under distribution shift.

To verify that fitting the calibrator and selecting $\tau^*$ on the same validation partition does not produce optimistic estimates, a controlled repetition split the validation partition into two disjoint stratified halves ($n$=12,597 for calibration and $n$=12,598 for threshold selection). Held-out test $F_1$ changed by $-0.013$ (Random Forest), 0.000 (XGBoost, MLP), $+0.007$ (Logistic Regression, CatBoost), and $+0.008$ (LightGBM), while Brier scores moved by at most $\pm 0.011$. The differences are small, non-systematic in sign, and in three of six models favor the disjoint protocol, indicating that the shared-validation protocol does not inflate reported test performance. The main results retain the shared protocol because it leaves more samples available for the isotonic fit.

### E. Ablation and Leakage-Sensitivity Analysis

Using the synthetic benchmark split adopted in the main study, a LightGBM ablation removed one component at a time from the full pipeline (feature selection, calibration, threshold optimization, LLM feature block, and artifact-aware preprocessing). The full configuration reached $F_1$=0.292, MCC= 0.162, and Brier= 0.1069. Removing the LLM block reduced $F_1$ to 0.273 ($\Delta F_1 = -0.0189$) and MCC to 0.135 ($\Delta$MCC= $-0.0269$), while removing threshold optimization (fixed $\tau$=0.5) collapsed $F_1$ to 0.000, confirming that validation-set threshold selection is operationally necessary under class imbalance. Removing calibration preserved discrimination ($\Delta F_1 \approx 0$) but degraded probability quality (Brier 0.1069$\rightarrow$0.1210), indicating that calibration mainly improves probability reliability rather than ranking. Feature-selection removal caused only marginal changes ($\Delta F_1 = -0.0005$), indicating limited sensitivity around the selected feature budget in this dataset. Removing artifact-aware preprocessing also produced a small but negative change in decision metrics ($\Delta F_1 = -0.0007$), which is consistent with the contract acting primarily as a robustness and reproducibility mechanism rather than a standalone discriminative booster.

For direct evidence on the leakage-aware contract, a direct comparison tested the leakage-safe protocol against an intentionally naive variant that fits encoding/scaling/feature-selection transforms before separation. On the synthetic benchmark, the naive protocol degraded test performance from $F_1$=0.311 to 0.280 ($\Delta = -0.031$) and AUC from 0.669 to 0.653 ($\Delta = -0.016$). On NSL-KDD, the same naive strategy reduced AUC from 0.942 to 0.928 ($\Delta = -0.015$) and $F_1$ from 0.803 to 0.797 ($\Delta = -0.006$). These results show that pre-split transform fitting perturbs model selection and can distort reported performance; therefore, all reported primary results retain the leakage-safe, artifact-aware contract.

### F. Reproducibility Specification

To support independent replication of the principal results reported in this revision, the following execution profile matches the released scripts and artifacts: random seed 42 (NumPy/Torch and data splitting), Python 3.11.6, NumPy 1.26.4, pandas 2.2.3, scikit-learn 1.9.1, XGBoost 3.2.0, LightGBM 4.7.0, CatBoost 1.2.10, and PyTorch 2.14.0. NSL-KDD preprocessing uses train-only label encoding, train-only z-score statistics, and MI top-$k$ selection with $k$=30. Model hyperparameters are explicitly fixed as follows: LogisticRegression ($C$=1.0, max_iter= 2000), RandomForest ($n_estimators$=300), XGBoost ($n_estimators$=400, max_depth= 6, lr= 0.1, subsample/colsample= 0.9), LightGBM ($n_estimators$=400, num_leaves= 63, lr= 0.05, subsample/colsample= 0.9), CatBoost (iterations= 500, depth= 6, lr= 0.1), and MLP (hidden layers 128/64, dropout 0.1, Adam lr= $10^{-3}$, batch 512, max 30 epochs, early-stopping patience 5). Calibration uses isotonic regression on validation and threshold selection uses validation-set $F_1$ maximization. Hardware/firmware profile: cloud Xeon+NVIDIA T4 tier, edge Raspberry Pi 4 tier, and ESP32-S3 endpoint compiled with ESP-IDF/FreeRTOS under $-Os$.

### G. Limitations

The current evidence has four major limitations. First, the main dataset is synthetic and rule-labeled, so learned decision structure may partially reflect generator design. Second, external validation is currently limited to NSL-KDD; although useful for portability checks, this is not equivalent to independent real-world operational validation. Third, hardware-tier comparisons use different model variants per tier, so cross-tier deltas reflect joint hardware+model-profile effects rather than isolated hardware effects. Fourth, the study does not yet include realistic temporal drift, adaptive adversarial traffic, or live user-behavior evaluation. For these reasons, deployment-readiness claims are intentionally bounded to proof-of-concept scope pending field validation campaigns.

## V. CONCLUSION

RFS defines a risk assessment module inside Q-OPSEC, aligning model training, artifact governance, and inference

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



semantics under a single serving contract for cloud-edge-IoT deployment of IoT-enabled sensitive data systems. Results demonstrate that heterogeneous model families under shared preprocessing and calibration contracts produce consistent probability scores and support threshold-based decision policies; the manuscript reports calibration evidence separately for non-quantized model-family analysis and quantized endpoint deployment.

On 50,000 samples with 1:11 class imbalance, LightGBM achieves MCC of 0.8376 and F1 of 0.8506 (recall 83.63%, specificity 98.84%), while Logistic Regression reaches only MCC=0.2695 and F1=0.3353. Hardware profiling yields latency of 12.4 ms on cloud servers (80.6 samples/s), 85.7 ms on edge gateways (11.6 samples/s), and 310.5 ms on ESP32-S3 endpoints (3.2 samples/s), with energy efficiency spanning 0.85 to 21.3 samples/J. Because tiers run different model variants, these deltas represent joint hardware+model-profile effects rather than hardware-only gains. Model-level calibration on non-quantized families is summarized by Brier 0.0686–0.0804 (best boosted models), whereas the FP32 vs quantized endpoint comparison captures deployment-level calibration for the quantized ESP32-S3 profile (Brier 0.1835 vs 0.1833; ECE 0.1911 vs 0.1910).

RFS extends the Q-OPSEC research line [1], [3]–[5] through: (1) a leakage-aware preprocessing contract binding encoding, normalization, and feature selection into versioned artifacts with deterministic training-inference equivalence; (2) a unified orchestrator evaluating six heterogeneous families with per-model calibration and threshold optimization; and (3) a version-resolved serving architecture enabling hardware-adaptive deployment with audit traceability. Accordingly, the novelty claim is not that individual MLOps components are new, but that their joint formulation is constrained by leakage-safe learning and calibration-aware decision governance for cross-tier IoT risk scoring.

External validation on the NSL-KDD intrusion-detection benchmark indicates portability beyond the synthetic dataset, with LightGBM achieving AUC-ROC of 0.942 and $F_1$ of 0.803 under distribution shift. These results remain proof-of-concept and do not yet establish operational suitability under real temporal drift, adversarial traffic, or live user behavior. Further validation on real operational data and additional public benchmarks remains a direction for future work. Future work includes continuous monitoring with drift detection, calibration tracking, and version-aware threshold adaptation integrated with alerting, retraining, and rollback policies; fairness and adversarial-robustness assessment; hardware-adaptive deployment through quantization, distillation, and feature pruning for sub-watt sensor nodes; streaming and temporal learning with sequence models and incremental updates; and privacy-preserving federated training within Q-OPSEC while maintaining traceability and reproducibility.

## ACKNOWLEDGMENTS

The authors acknowledge financial support from CNPq (National Council for Scientific and Technological Development, grant 307137/2022-8) and CAPES (Coordination for the Improvement of Higher Education Personnel – Brazil, Finance Code 001). We are also grateful to Unisinos (University of Vale do Rio dos Sinos), ISCTE-IUL (University Institute of Lisbon) and the University of Salamanca for undertaking this research.

## REFERENCES

[1] D. Noetzold, J. L. V. Barbosa, J. F. De Paz, and V. R. Q. Leithardt, "Reinforcement learning-based adaptive quantum-safe cryptography for DN25-compliant smart environments," *IEEE Access*, vol. 14, pp. 62087–62104, 2026, doi: 10.1109/ACCESS.2026.3685890.

[2] D. Noetzold, V. R. Q. Leithardt, and J. L. V. Barbosa, "Performance monitoring and self-adaptation in smart environments: a systematic literature review," *Telecommunication Systems*, vol. 89, p. 13, 2026, doi: 10.1007/s11235-025-01387-8.

[3] D. Noetzold, V. R. Q. Leithardt, J. F. de Paz, and J. L. V. Barbosa, "End-to-end IoT sensor data simulation and predictive analysis: framework implementation and experimental evaluation," *Scientific Reports*, 2026, doi: 10.1038/s41598-026-52981-y.

[4] D. Noetzold, V. R. Q. Leithardt, J. F. de Paz Santana, and J. L. V. Barbosa, "Oraculum: A model for self-adaptive system optimization in smart environments," *Expert Systems with Applications*, p. 131705, 2026, doi: 10.1016/j.eswa.2026.131705.

[5] D. Noetzold, J. L. V. Barbosa, J. F. P. Santana, and V. R. Q. Leithardt, "Q-OPSEC: An Architecture Proposal for Adaptive AI Middleware for Quantum Cryptography," in *2025 International Conference on Electrical and Computer Engineering Researches (ICECER)*, 2025, pp. 1–6, doi: 10.1109/ICECER65523.2025.11401311.

[6] R. Latif, B. M. Yakubu, N. S. M. Jamail, et al., "BBAS: A blockchain-based authentication system for e-health with multi-factor authentication, access control, and post-quantum security," *Scientific Reports*, 2026, doi: 10.1038/s41598-026-39415-5.

[7] S. Mumtaz, N. Sanam, and F. Mumtaz, "AI-Powered security classification of generator-matrix s-box for post-quantum lightweight cryptography," *Telecommunication Systems*, vol. 89, p. 20, 2026, doi: 10.1007/s11235-025-01395-8.

[8] U. Ghosh, L. Njilla, D. Das, and E. Levin, "FLAS: A Federated Learning Framework for Adaptive Security in Edge-Driven UAV Networks," in *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 1298–1303, doi: 10.1109/ICC51166.2024.10622352.

[9] A. Holgado, J. Portilla, D. López-Fernández, and L. Redondo, "Context-Aware Security and Post Quantum Cryptography Applied to IoT Networks," in *2024 4th Intelligent Cybersecurity Conference (ICSC)*, 2024, pp. 175–179, doi: 10.1109/ICSC63108.2024.10894874.

[10] V. S. S. Reddy Nallapareddy, "Machine Learning-Based Adaptive Cybersecurity Framework for the Internet of Things," in *2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC)*, 2024, pp. 1–6, doi: 10.1109/ICEC59683.2024.10837268.

[11] S. Xu, X. Han, G. Xu, J. Ning, X. Huang, and R. H. Deng, "An Adaptive Secure and Practical Data Sharing System With Verifiable Outsourced Decryption," *IEEE Transactions on Services Computing*, vol. 17, no. 3, pp. 776–788, 2024, doi: 10.1109/TSC.2023.3321314.

[12] A. S. Alsalim, "Improving the Security of Enterprise Storage Systems With Efficient Anomaly Detection," *IEEE Reliability Magazine*, vol. 2, no. 2, pp. 35–39, 2025, doi: 10.1109/MRL.2025.3556806.

[13] H. Shingne, D. Chikmurge, P. Parkhi, and P. Agrawal, "Design of an integrated model using deep reinforcement learning and Variational Autoencoders for enhanced quantum security," *MethodsX*, vol. 15, p. 103445, 2025, doi: 10.1016/j.mex.2025.103445.

[14] S. H. Rafat et al., "Lightweight Cryptographic Algorithm Analysis for Secure IoT Communication on ESP-32 Platforms," in *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*, 2025, pp. 1–6, doi: 10.1109/QPAIN66474.2025.11171742.

[15] H. Venkatesh et al., "Reinforcement Learning for Trust-Based Black Hole Attack Mitigation in IIoT Networks," in *2025 2nd International Conference On Multidisciplinary Research and Innovations in Engineering (MRIE)*, 2025, pp. 364–369, doi: 10.1109/MRIE66930.2025.11156663.

[16] M.-A. Kourtis et al., "Adaptive Post-Quantum Cryptography for Blockchain: Enhancing Security and Energy Efficiency with Cryptographic Agility," in *2025 6th International Conference in Electronic Engineering & Information Technology (EEITE)*, 2025, pp. 1–5, doi: 10.1109/EEITE65381.2025.11166257.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3748674



[17] J. Zhang, Z. Liu, and D. Yao, "Fine-grained Privately Puncturable Signatures with Adaptive Security from Lattices," in *Proceedings of the 10th International Conference on Cyber Security and Information Engineering (ICCSIE '25)*, 2025, pp. 251–255, doi: 10.1145/3759179.3760358.

[18] B. K. Kumar et al., "Automated cryptographic weakness discovery: A reinforcement learning approach for adaptive cryptanalysis," in *2025 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, 2025, pp. 1–5, doi: 10.1109/ASSIC64892.2025.11158372.

[19] J. Du, B. Ai, J. Liang, S. Liu, Y. Lu, and B. Shao, "An adaptive secure identity-based signcryption with equality test for high-speed railway train–ground communication," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 1872–1886, 2026, doi: 10.1109/TNSE.2025.3605157.

[20] D. Noetzold and V. R. Q. Leithardt, *Fundamentos, Avanços Recentes e Desafios com Algoritmos Criptográficos: Criptografia Clássica, Quântica e Pós Quântica*, KS OmniScriptum Publishing, 2026, ISBN: 9786209263446.

[21] Y. Huang, M. Ma, W. J. K. Raymond, and C.-O. Chow, "An adaptive intrusion detection system for the internet of things using large language models and post-quantum-secure blockchain," *Computer Networks*, vol. 274, p. 111819, 2026, doi: 10.1016/j.comnet.2025.111819.

[22] Z. A. Haider, A. Zeb, A. K. M. M. Islam, T. Rahman, A. Arishi, and I. Ullah, "Enhancing IoT security with resource-efficient cryptography: A comprehensive review of lightweight and hybrid algorithms," *Computer Science Review*, vol. 59, p. 100861, 2026, doi: 10.1016/j.cosrev.2025.100861.

[23] T.-C. Ho, Y.-M. Tseng, and S.-S. Huang, "LHSC-SGC: A lightweight hybrid signcryption scheme for smart grid communications in heterogeneous cryptographic public-key systems," *Computer Standards & Interfaces*, vol. 96, p. 104078, 2026, doi: 10.1016/j.csi.2025.104078.

[24] H. Sohail, D. Noetzold, and V. R. Q. Leithardt, "PRISEC III: Dynamic Cryptographic Adaptation for Balancing Performance and Security," *Journal of Internet Services and Applications*, vol. 17, no. 1, pp. 174–191, 2026, doi: 10.5753/jisa.2026.6502.

[25] P. S. Sasongko, A. Wibowo, and D. Hartanto, "Extended TAM analysis for e-learning acceptance in higher education," *Array*, vol. 25, p. 100192, 2025, doi: 10.1016/j.array.2025.100192.

[26] Z. Tari, A. Mahmood, and P. Bertok, "Human-centric cybersecurity for resilient digital ecosystems," *Future Generation Computer Systems*, vol. 162, pp. 107–121, 2025, doi: 10.1016/j.future.2024.09.043.

[27] R. Ibrahim, N. Zainuddin, and S. Azizan, "Effect of VR on decision-making, presence, and task load in operational scenarios," *Array*, vol. 26, p. 100343, 2025, doi: 10.1016/j.array.2025.100343.

[28] M. A. Khan, M. Alazab, and Q. Xu, "Advancing Security in Industrial Environments: Deep Learning Solutions for Cyberattack Detection," *Processes*, vol. 12, no. 9, p. 1843, 2024, doi: 10.3390/pr12091843.

[29] X. A. Wang, Z. Liu, W. Liu, W. Jiang, W. Jiang, and Y. Li, "Privacy-Preserving Revocable Certificateless Cloud Data Auditing Scheme Based on IoT Data Collection for Secure Medical Data Sharing," *IEEE Internet of Things Journal*, vol. 13, no. 13, pp. 28428–28444, 2026, doi: 10.1109/JIOT.2026.3681221.

[30] J. Lin, J. Li, X. Di, and G. Yan, "A Lattice-Based Traceable and Direct Revocable ABPRE With Fair Verification for Data Sharing in Medical Internet of Things," *IEEE Internet of Things Journal*, vol. 13, no. 13, pp. 28476–28489, 2026, doi: 10.1109/JIOT.2026.3684480.

[31] W. Sun et al., "An Advanced GNSS Interference Classification Detector Based on Multidomain Feature Fusion," *IEEE Internet of Things Journal*, vol. 13, no. 13, pp. 28181–28196, 2026, doi: 10.1109/JIOT.2026.3678055.

[32] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[33] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998, doi: 10.1023/A:1008306431147.

[34] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

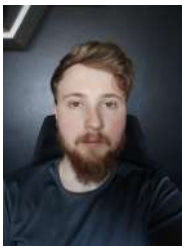
[35] A. Ibrahim, A. F. Kadhim, A. E. Hamzah, and M. A. Al-Shareeda, "A secure and scalable IoT home automation architecture with web and biometric control," *International Journal of Cybersecurity Engineering and Innovation*, vol. 2026, no. 1, 2026.

[36] V. Abdullayev, A. Khang, and N. Ragimova, "Quantum-Inspired Cyber Threat Detection and Adaptive Security Management for IoT Networks," *Journal of Information and Cybersecurity*, 2026. [Online]. Available: https://journals.theitap.org/index.php/jics/article/view/25

[37] Z. A. Haider, A. Zeb, T. Rahman, S. K. Singh, R. Akram, A. Arishi, and I. Ullah, "A Survey on anomaly detection in IoT: Techniques, challenges, and opportunities with the integration of 6G," *Computer Networks*, vol. 270, p. 111484, 2025, doi: 10.1016/j.comnet.2025.111484.

[38] A. J. Aparcana-Tasayco, X. Deng, and J. H. Park, "A systematic review of anomaly detection in IoT security: towards quantum machine learning approach," *EPJ Quantum Technology*, vol. 12, p. 112, 2025, doi: 10.1140/epjqt/s40507-025-00414-6.

[39] M. Aly and M. H. Behiry, "Enhancing anomaly detection in IoT-driven factories using Logistic Boosting, Random Forest, and SVM: A comparative machine learning approach," *Scientific Reports*, vol. 15, p. 23694, 2025, doi: 10.1038/s41598-025-08436-x.



**Darlan Noetzold** (Member, IEEE) is currently Ph.D. Student in Computer Engineering at the University of Salamanca, Spain. He received his Master's degree from the University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil. He also works as a Senior Software Developer at CWI Software. His main research interests are High Performance Computing, Smart Environments, Cybersecurity, Quantum Computing, Quantum Cryptography and Machine Learning.



**Jorge L. V. Barbosa** received M.Sc. and Ph.D. in computer science from the Federal University of Rio Grande do Sul, Brazil. He conducted post-doctoral studies at Sungkyunkwan University (SKKU, Suwon, South Korea) and University of California Irvine (UCI, Irvine, USA). Jorge is a full professor at the Applied Computing Graduate Program (PPGCA) of the University of Vale do Rio dos Sinos (UNISINOS), head of the university's Mobile Computing Lab (MOBILAB), and a researcher at the Brazilian Council for Scientific and Technological Development (CNPq). His main research interests are Ubiquitous Computing, Ambient Intelligence, Big Data, Internet of Things (IoT), and Machine Learning.



**Juan F. P. Santana** received a degree in Technical Engineering in Computer Systems in 2003, a degree in Computer Science Engineering in 2005, a degree in Statistics in 2007, and a Ph.D. in Computer Science in 2010, all from the University of Salamanca, Spain. He is currently a Full Professor at the University of Salamanca and a researcher at the Expert Systems and Applications Laboratory (ESALab). He is the coauthor of several papers published in journals, workshops, and symposiums.



**Valderi R. Q. Leithardt** (Senior Member, IEEE) received the Ph.D. degree in computer science from INF-UFRGS, Brazil, in 2015. He is currently a Professor with the ISCTE - Instituto Universitario de Lisboa and a Researcher integrated with the Istar-Information Sciences, Technologies and Architecture Research Centre (ISTA). His research interests include distributed systems with a focus on data privacy, communication, and programming protocols, involving scenarios and applications for intelligent systems, internet of things, smart cities, big data and cloud computing.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/