A Modular Risk Assessment Module for Adaptive Cryptographic Selection in Q-OPSEC

Darlan Noetzold*, Jorge L. V. Barbosa‡, Juan F. De Paz*, Valderi R. Q. Leithardt†*
*Expert Systems and Applications Laboratory, University of Salamanca, Salamanca, Spain
Emails: darlannoetzold@usal.es, fcofds@usal.es
‡PPGCA, University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
Email: jbarbosa@unisinos.br
†ISTAR-IUL, Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
Email: valderi.leithardt@iscte-iul.pt

***Abstract*—This paper presents RiskService, a modular risk assessment module integrated into the Q-OPSEC adaptive AI middleware for quantum cryptography. A synthetic dataset covering 58 features across nine groups, including behavioral, device, network, authentication, and LLM-derived signals, feeds a training pipeline evaluating six model families under class-imbalanced conditions. LightGBM achieves the best performance, with AUC-ROC of 0.9895, average precision of 0.9344, and Brier score of 0.0421 at threshold 0.60, with inference latency of 1.8 ms. Deployment benchmarks across three hardware tiers confirm feasibility under constrained resources: quantized XGBoost runs in 54.2 ms on the ESP32 with AUC-ROC of 0.9112, enabling a two-tier architecture where edge nodes perform preliminary screening and forward ambiguous events for full-precision regime determination. Calibrated risk scores govern the selection among classical TLS 1.3, post-quantum, and hybrid key derivation paths in the Q-OPSEC cryptographic layer.**

***Index Terms*—adaptive cryptography, edge inference, fraud detection, post-quantum security, risk assessment**

I. Introduction

The accelerating convergence of classical, post-quantum, and quantum-era cryptography is reshaping how security services should be engineered and operated. Modern systems must simultaneously preserve backward compatibility with widely deployed classical primitives, begin migration toward post-quantum cryptography (PQC) to mitigate harvest-now, decrypt-later threats, and integrate quantum cryptographic capabilities such as QKD and hybrid key establishment where available. This multi-regime setting creates a new operational problem: cryptographic agility under uncertainty. The optimal choice of protocol, ciphersuite, key-establishment mode, or authentication strength depends on context, including device constraints, latency budgets, trust level, channel conditions, and adversarial indicators. Manual policies and static configurations struggle to capture this high-dimensional and non-stationary environment, motivating adaptive decision engines supported by artificial intelligence.

Recent work highlights two complementary directions. PQC- and blockchain-enabled authentication architectures demonstrate how multi-factor authentication, access control, and post-quantum hardening can be integrated into real-world security workflows [1]. In parallel, AI-based methods are increasingly applied to analyze and classify cryptographic components, helping automate the selection or validation of lightweight post-quantum building blocks [2]. These trends indicate that future security systems will require both cryptographic diversity across classical, PQC, quantum, and hybrid regimes, and learning-driven adaptation mechanisms capable of selecting protections dynamically according to runtime context.

This paper presents RiskService, a module of the Q-OPSEC middleware architecture [3]. Q-OPSEC is an adaptive AI middleware for quantum cryptography designed to orchestrate cryptographic decisions across classical, post-quantum, and quantum regimes through reinforcement learning and context-aware policies in smart environments. Within this broader architecture, RiskService is responsible for the end-to-end risk assessment loop, encompassing a modular synthetic dataset pipeline, an ML training pipeline that learns risk predictors and calibrates decision thresholds, and a low-latency serving layer that exposes consistent prediction endpoints and versioned model artifacts. The module incorporates features and decision hooks for hybrid key establishment and context-dependent selection, enabling downstream integration with quantum gateways or PQC policy modules within the Q-OPSEC stack.

The remainder of this paper is organized as follows. Section II reviews related works on adaptive cryptographic selection, reinforcement learning for security, and quantum-aware architectures. Section III details the RiskService methodology, covering dataset generation, feature engineering, and the training and serving infrastructure. Section IV presents the experimental results and discusses the implications for adaptive cryptographic decision-making. Section V concludes with limitations and directions for future work.

II. Related Works

The design of RiskService, as a module of the Q-OPSEC middleware [3], draws on a growing body of work spanning reinforcement learning for security policy management, post-quantum cryptography, quantum key distribution, and AI-driven adaptive systems. Table I summarizes the most relevant recent contributions (2024–2026) and their relationship to the RL-based cryptographic selection engine targeted by this work.

TABLE I
SELECTED RECENT WORKS (2024–2026) RELEVANT TO RL-BASED ADAPTIVE CRYPTOGRAPHIC SELECTION

Ref.	Year	Topic	Relevance to RL engine	Characteristics
[4]	2024	Federated learning	Demonstrates resource-aware security adaptivity at the network edge.	Proposes lightweight aggregation protocols suitable for constrained environments.
[5]	2024	Context-aware PQC	Evaluates PQC algorithm trade-offs for IoT, providing cost data for reward modeling.	Empirical profiling of PQC candidates using heuristic selection strategies.
[6]	2024	ML-based adaptation	Uses anomaly detection to trigger policy changes, analogous to RL state transitions.	Implements automated control loops for adaptive security in IoT deployments.
[7]	2024	Secure data sharing	Models cost-versus-security trade-offs relevant to multi-objective reward design.	Proposes verifiable outsourced decryption with formal security guarantees.
[8]	2025	Anomaly detection	Provides telemetry-driven triggers applicable to RL environment observation.	Efficient anomaly detectors tailored for storage security workloads.
[9]	2025	DRL + VAE for QKD	Combines deep RL with variational autoencoders for quantum channel optimization.	Integrated QKD parameter tuning via reward engineering on quantum metrics.
[10]	2025	Lightweight crypto	Supplies latency and energy benchmarks usable as cost signals in RL rewards.	Empirical evaluation of lightweight cryptographic primitives on ESP32 hardware.
[11]	2025	Trust-based RL	Incorporates trust metrics into RL states for adaptive attack mitigation.	Trust-aware state representation with shaped rewards for resilient networking.
[12]	2025	Adaptive PQC	Addresses energy-aware cryptographic agility, informing long-term cost modeling.	Blockchain-oriented policies balancing operational cost and post-quantum security.
[13]	2025	Lattice signatures	Introduces PQC signature primitives that extend the cryptographic action space.	Adaptively secure lattice-based signatures with fine-grained puncturability.
[1]	2026	Blockchain-based auth	Multi-factor and PQC-aware authentication informs security state modeling.	Blockchain-backed e-health system with access control and post-quantum hardening.
[2]	2026	AI-powered S-box	ML-based classification of cryptographic primitives supports action selection.	Generator-matrix S-box design evaluated for post-quantum lightweight cryptography.
[20]	2026	Self-adaptive optimization	General self-adaptive framework applicable to middleware-level policy management.	MDP-inspired model for system optimization in smart and dynamic environments.
This work	2026	RL engine for crypto	MDP-based selection across classical, PQC, QKD, and hybrid cryptographic protocols.	Multi-objective reward function with context-aware middleware and utility model.

Rafat et al. [10] provide empirical benchmarks of lightweight cryptographic primitives on ESP32 hardware, supplying latency and energy measurements usable as cost signals in RL reward functions for edge and IoT deployments. Venkatesh et al. [11] extend this direction by incorporating trust metrics into RL states for adaptive attack mitigation, demonstrating that non-cryptographic signals such as node reputation and behavioral scores can be effectively encoded as environment observations, a principle that RiskService operationalizes through its feature set covering device posture, authentication history, and network signals. Shingne et al. [9] combine deep RL with variational autoencoders for quantum channel parameter optimization, targeting QKD throughput and error rate as reward signals, bridging the gap between classical policy optimization and quantum channel management.

Holgado et al. [5] provide empirical profiling of PQC algorithm candidates for IoT environments, evaluating trade-offs in latency, energy consumption, and security level, with heuristic selection strategies and cost data that serve as a baseline for reward modeling in constrained deployments. Kourtis et al. [12] address energy-aware cryptographic agility in blockchain contexts, proposing policies that balance long-term operational cost against post-quantum security guarantees, establishing part of the empirical cost landscape that informs the multi-objective reward function of the Q-OPSEC RL engine. Zhang et al. [13] introduce fine-grained privately puncturable signatures with adaptive security from lattices, extending the set of PQC primitives available as actions in a cryptographic selection system. Mumtaz et al. [2] apply AI-based classification to generator-matrix S-box designs, demonstrating that machine learning can automate the evaluation and selection of post-quantum lightweight building blocks. Latif et al. [1] integrate multi-factor authentication, access control, and post-quantum hardening into a blockchain-backed e-health architecture, illustrating how PQC-aware authentication can be modeled as a structured security state in adaptive systems.

Ghosh et al. [4] address resource-aware security adaptivity through federated learning, proposing lightweight aggregation protocols for constrained environments and highlighting the importance of accounting for computational heterogeneity when designing adaptive security policies. Nallapareddy et al. [6] implement automated control loops for adaptive security in IoT deployments, using anomaly detection to trigger policy changes in a manner analogous to RL state transitions. Alsalim et al. [8] propose efficient anomaly detectors for storage security workloads, providing telemetry-driven triggers applicable to the observation layer of an RL environment. Xu et al. [7] model cost-versus-security trade-offs in verifiable outsourced decryption, providing formal security guarantees alongside a utility analysis relevant to multi-objective reward design, aligned with the utility model pursued in Q-OPSEC. Noetzold et al. [20] propose a general self-adaptive optimization model for smart environments drawing on MDP-inspired formulations for system-level policy management, providing architectural grounding for the middleware layer of Q-OPSEC where RiskService operates as the risk-aware observation and inference module feeding the adaptive decision engine.

The surveyed works confirm that no existing system addresses the full pipeline from labeled security telemetry generation through risk-aware ML inference to RL-based cryptographic selection across classical, PQC, QKD, and hybrid

regimes. RiskService is positioned to fill this gap as a concrete, deployable module within the Q-OPSEC architecture.

III. Methodology

RiskService is structured as four interdependent modules, illustrated in Figure 1: Dataset Generation (A), Training Pipeline (B), Inference API (C), and LLM Integration (D). Each module addresses a distinct concern in the adaptive risk assessment loop, and together they form the observation and inference layer of the Q-OPSEC middleware [3].

The dataset generation module, shown in Figure 1A, produces a labeled security telemetry corpus $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector encoding a security event and $y_i \in \{0, 1\}$ is the associated risk label. The generation pipeline follows three sequential stages. A User Generator instantiates synthetic user profiles with associated behavioral parameters, device characteristics, and communication patterns, producing a population $\mathcal{U} = \{u_1, \ldots, u_M\}$, where each user u_j carries a profile tuple $\langle \pi_j, \delta_j, \beta_j \rangle$ encoding profile type π_j, device fingerprint δ_j, and behavioral distribution parameters β_j.

A Transaction Simulator then generates event streams for each user, sampling transaction amounts from a log-normal distribution $a \sim \mathcal{LN}(\mu_\pi, \sigma_\pi)$ parameterized per profile type, alongside channels, timestamps, and authentication outcomes. Risk scenarios covering account takeover, card testing, phishing-induced transfers, money mule activity, and synthetic identity fraud are injected at configurable rates drawn from a multinomial distribution over scenario types. An Anomaly Injection stage introduces controlled deviations from baseline behavior, including impossible travel events, sudden amount spikes, and device switching patterns, after which a Labeling and Export stage assigns ground-truth labels and partitions the corpus into training, validation, and test splits according to a strict time-based cut to prevent temporal leakage.

Each event e_i is mapped to a feature vector through a modular extraction pipeline. Let $\phi_k : \mathcal{E} \to \mathbb{R}^{d_k}$ denote the k-th feature extractor, covering temporal, behavioral, device, network, authentication, location, and content dimensions. The full feature vector is the concatenation:

$$\mathbf{x}_i = \bigoplus_{k=1}^{K} \phi_k(e_i), \quad \mathbf{x}_i \in \mathbb{R}^d, \quad d = \sum_{k=1}^{K} d_k \tag{1}$$

Table II details the nine feature groups, their dimensionality d_k, mathematical formulation, parameters, and role in the risk model, yielding a total feature space of $d = \sum_{k=1}^{9} d_k = 58$ dimensions per event. The behavioral rolling group deserves particular attention: the aggregates $\rho_w^{(v)}$ capture velocity patterns across four time windows, enabling detection of card-testing bursts that manifest as high transaction counts with low individual amounts within short intervals. The authentication group ϕ_6 is the primary bridge between RiskService and the Q-OPSEC cryptographic selection layer, as the composite authentication score s_{auth} and the PQC flag $\nVdash_{\text{pqc}}$ directly inform the regime escalation logic described later in this section.

The training pipeline, shown in Figure 1B, consumes $\mathcal{D}$ and produces a set of calibrated risk models $\{\mathcal{M}_k\}$ stored in a versioned model registry. Data preprocessing applies z-score normalization to continuous features and ordinal encoding to categorical ones, followed by missing value imputation using median statistics computed exclusively on the training split. Model training minimizes a weighted binary cross-entropy loss to account for class imbalance arising from the low base rate of fraud events:

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} w_{y_i} \Big[-y_i \log f(\mathbf{x}_i; \theta) - (1 - y_i) \log(1 - f(\mathbf{x}_i; \theta)) \Big] \tag{2}$$

where $w_1 = N/(2N_+)$ and $w_0 = N/(2N_-)$ are inverse-frequency class weights, N_+ and N_- denote the number of positive and negative samples respectively, and $f(\mathbf{x}; \theta)$ is the model output. The pipeline supports Logistic Regression, Random Forest, XGBoost, and LightGBM, all governed by a unified ModelFactory interface that enforces a consistent fit-predict-calibrate contract. Validation and tuning apply stratified k-fold cross-validation with $k = 5$ and Bayesian hyperparameter search over a predefined configuration space. Probability calibration via isotonic regression is applied post-training to ensure that $f(\mathbf{x}; \theta^*)$ yields reliable posterior estimates suitable for threshold-based regime escalation.

Algorithm 1 details the full training procedure, including the time-based split, Bayesian search loop, calibration, and registry promotion logic. The time-based split is a deliberate design choice: because fraud patterns evolve over time, random splits would allow future behavioral signals to leak into training, producing optimistic estimates that do not reflect deployment conditions. The final registry promotion step selects the single best model across all families for the active inference slot, while retaining all trained artifacts to support rollback and A/B evaluation. The Inference API, shown in Figure 1, exposes trained models as versioned REST endpoints. At inference time, an incoming API request carrying a raw event payload is preprocessed by the same Φ fitted during training, and the active model produces a calibrated posterior risk estimate:

$$\tilde{y} = P(y = 1 \mid \mathbf{x}, \theta^*) = \mathcal{C}_{\mathcal{M}^*}(f(\Phi(\mathbf{x}); \theta^*)) \tag{3}$$

where $\mathcal{C}_{\mathcal{M}^*}$ is the isotonic calibrator fitted for the active model. The response includes the risk score $\tilde{y} \in [0, 1]$, a discretized risk tier τ derived from configurable thresholds $\lambda_1 < \lambda_2 < \lambda_3$:

$$\tau(\tilde{y}) = \begin{cases} \text{low} & \text{if } \tilde{y} < \lambda_1 \\ \text{medium} & \text{if } \lambda_1 \le \tilde{y} < \lambda_2 \\ \text{high} & \text{if } \lambda_2 \le \tilde{y} < \lambda_3 \\ \text{critical} & \text{if } \tilde{y} \ge \lambda_3 \end{cases} \tag{4}$$

alongside the model version identifier and per-feature attribution scores computed via SHAP values $\phi_j^{\text{SHAP}}(\mathbf{x}) = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{j\}) - f(S)]$, where F is the full feature set. The serving layer maintains a model

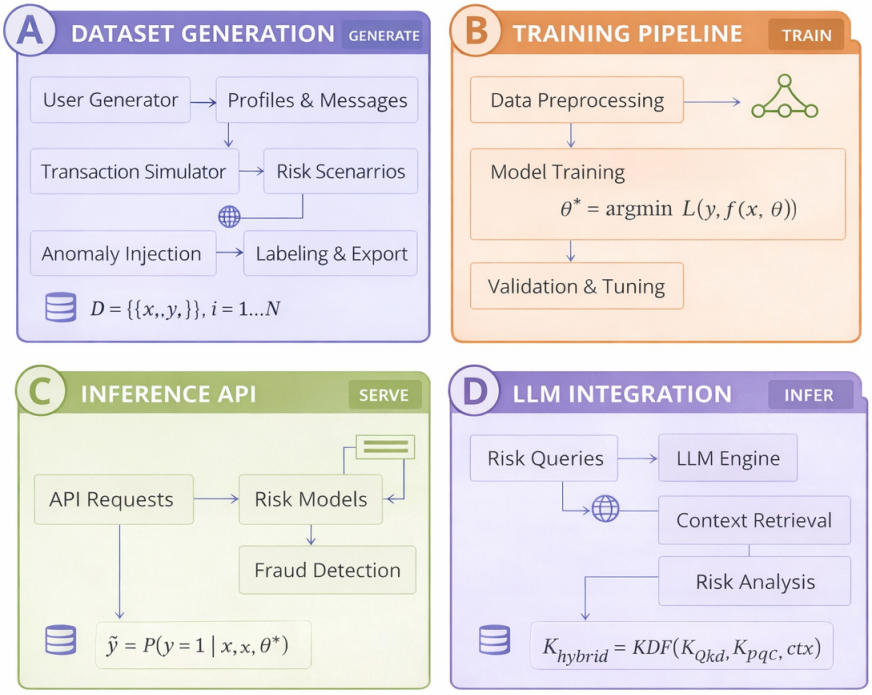

Fig. 1. RiskService architecture: (A) Dataset Generation, (B) Training Pipeline, (C) Inference API, and (D) LLM Integration.

TABLE II
FEATURE GROUPS IN THE RISKSERVICE DATASET GENERATION PIPELINE, WITH FORMULATIONS, DIMENSIONALITY, AND ROLE IN THE RISK MODEL

Group	d_k	Formulation	Parameters	Role in risk model
Temporal	6	$\phi_1(e_i) = (h_i,\ w_i,\ \delta t_i,\ \mathbb{1}_{\text{wknd}},\ \mathbb{1}_{\text{night}},\ \mathbb{1}_{\text{hol}})$	$h_i \in [0, 23]$, $w_i \in [0, 6]$, $\delta t_i = t_i - t_{i-1}$	Captures off-hours and holiday anomalies correlated with fraud onset.
Behavioral rolling	8	$\rho_w^{(v)}(i) = \sum_{j:\, t_j \in [t_i - w,\, t_i)} v_j$	$w \in \{1\text{h}, 6\text{h}, 24\text{h}, 7\text{d}\}$, $v_j \in \{1, a_j\}$	Rolling count and amount aggregates detect velocity bursts and card-testing patterns.
Transaction	6	$\phi_3(e_i) = (a_i,\ \log(1 + a_i),\ \hat{a}_i,\ c_i,\ r_i,\ \mathbb{1}_{\text{new}})$	$\hat{a}_i = (a_i - \mu_u)/\sigma_u$, $c_i \in \mathbb{Z}$, $r_i \in [0, 1]$	Amount deviation $\hat{a}_i$ from user baseline flags outlier transactions.
Device	8	$\phi_4(e_i) = (\text{os}_i,\ \text{br}_i,\ v_{\text{os}},\ v_{\text{br}},\ \mathbb{1}_{\text{emu}},\ \mathbb{1}_{\text{vpn}},\ \mathbb{1}_{\text{out}},\ n_{\text{dev}})$	$v_{\text{os}}, v_{\text{br}} \in \mathbb{Z}^+$, n_{dev} = distinct devices in 30d	Emulator and VPN flags, outdated software, and device proliferation signal account takeover.
Network	7	$\phi_5(e_i) = (\text{ip}_{\text{rep}},\ \text{asn}_i,\ \text{isp}_i,\ \mathbb{1}_{\text{tor}},\ \mathbb{1}_{\text{prx}},\ \mathbb{1}_{\text{blk}},\ \Delta_{\text{geo}})$	$\text{ip}_{\text{rep}} \in [0, 1]$, Δ_{geo} in km	Blacklisted IPs, anonymization proxies, and geodesic displacement expose network-layer evasion.
Authentication	9	$\phi_6(e_i) = (\text{mfa}_i,\ \text{tls}_i,\ \mathbb{1}_{\text{pqc}},\ \mathbb{1}_{\text{fail}},\ n_{\text{fail}},\ \Delta t_{\text{mfa}},\ s_{\text{auth}},\ \text{alg}_i,\ \mathbb{1}_{\text{cap}})$	n_{fail} cumulative, Δt_{mfa} in seconds, $s_{\text{auth}} \in [0, 1]$	Authentication posture feeds directly into Q-OPSEC cryptographic regime selection.
Location	5	$\phi_7(e_i) = (\text{cty}_i,\ \text{reg}_i,\ \mathbb{1}_{\text{trav}},\ \Delta_{\text{geo}},\ \Delta t_{\text{loc}})$	$\mathbb{1}_{\text{trav}} = \mathbb{1}[\Delta_{\text{geo}}/\Delta t_{\text{loc}} > v_{\max}]$	Impossible travel detection via velocity threshold $v_{\max} = 900$ km/h.
Security tech	6	$\phi_8(e_i) = (\text{waf}_i,\ \text{ids}_i,\ \text{av}_i,\ v_{\text{tls}},\ \mathbb{1}_{\text{hsts}},\ \mathbb{1}_{\text{csp}})$	$v_{\text{tls}} \in \{1.0, 1.1, 1.2, 1.3\}$, binary flags	Security stack posture informs reward shaping in the Q-OPSEC RL engine.
LLM-derived	3	$\phi_9(e_i) = (s_{\text{risk}},\ s_{\text{phish}},\ s_{\text{intent}}) \in [0, 1]^3$	Produced by Module D for message-bearing events	Semantic risk signals from unstructured content complement structured features.

a_i: transaction amount; μ_u, σ_u: user-level mean and std; c_i: channel encoding; r_i: recipient risk score; n_{dev}: device count in 30-day window.

registry that supports atomic version promotion and rollback, enabling safe deployment of retrained artifacts without service interruption.

The LLM Integration module, shown in Figure 1D, augments structured feature vectors with semantic risk signals derived from unstructured event content such as transaction messages, authentication challenge responses, and communication metadata. Risk queries are dispatched to a locally hosted LLM engine via the Ollama interface, accompanied by retrieved contextual information including user history summaries and known fraud pattern templates assembled through a lightweight retrieval step. The engine returns three scalar scores: a general risk score s_{risk}, a phishing probability s_{phish}, and an intent score s_{intent}, all normalized to $[0, 1]$ through a structured JSON parsing layer with fallback values for inference failures. These scores are incorporated into the feature

Algorithm 1: RiskService Training Pipeline

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, model families $\mathcal{F}$, folds k, search budget B
Output: Calibrated model registry $\mathcal{R}$, evaluation report $\mathcal{E}$

Split $\mathcal{D}$ by time cut into $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, $\mathcal{D}_{\text{test}}$ with ratios $0.70/0.15/0.15$;
Fit preprocessor Φ on $\mathcal{D}_{\text{train}}$; apply Φ to all splits without refitting;
Compute class weights w_0, w_1 from $\mathcal{D}_{\text{train}}$ label distribution;
for *each model family* $\mathcal{M} \in \mathcal{F}$ **do**
    Initialize Bayesian search over $\Lambda_{\mathcal{M}}$ with budget B;
    for *each candidate* $\theta \in \Lambda_{\mathcal{M}}$ **do**
        Evaluate AUC-ROC($\mathcal{D}_{\text{val}}$, $\mathcal{M}(\theta)$) via k-fold CV on $\mathcal{D}_{\text{train}}$;
        Update surrogate model of Bayesian optimizer;
    Select $\theta^*_{\mathcal{M}} \leftarrow \arg\max_{\theta \in \Lambda_{\mathcal{M}}}$ AUC-ROC;
    Retrain $\mathcal{M}(\theta^*_{\mathcal{M}})$ on full $\mathcal{D}_{\text{train}}$;
    Fit isotonic calibrator $\mathcal{C}_{\mathcal{M}}$ on $\mathcal{D}_{\text{val}}$ outputs;
    Compute {AUC-ROC, F1, precision, recall, Brier score} on $\mathcal{D}_{\text{test}}$;
    Store ($\mathcal{M}$, $\theta^*_{\mathcal{M}}$, $\mathcal{C}_{\mathcal{M}}$, metrics) in $\mathcal{R}$;
Promote $\mathcal{M}^* = \arg\max_{\mathcal{M} \in \mathcal{F}}$ AUC-ROC$_{\mathcal{D}_{\text{test}}}$ to active slot in $\mathcal{R}$;
return $\mathcal{R}$, $\mathcal{E}$;

vector as the LLM-derived group ϕ_9 described in Table II, and are available both at dataset generation time, enriching training labels, and at inference time, augmenting live predictions for message-bearing events.

The cryptographic output of the Q-OPSEC layer, informed by the risk tier τ produced by RiskService, governs the selection of key derivation path through the hybrid key derivation function:

$$K_{\text{hybrid}} = \text{KDF}(K_{\text{QKD}},\, K_{\text{PQC}},\, \text{ctx}) \tag{5}$$

where K_{QKD} is the quantum-distributed key material, K_{PQC} is the post-quantum key encapsulation output, and ctx is a context binding string encoding the event identifier, risk tier τ, timestamp, and session metadata. When $\tilde{y} \geq \lambda_3$, the Q-OPSEC engine escalates to the full hybrid derivation path activating both K_{QKD} and K_{PQC}; for $\lambda_2 \leq \tilde{y} < \lambda_3$, a PQC-only path is selected; and for $\tilde{y} < \lambda_2$, a classical TLS 1.3 path suffices, balancing security strength against computational overhead in resource-constrained deployments.

IV. Results and Discussion

The experimental evaluation assesses three complementary aspects: separability of the generated feature space, predictive performance across model families, and deployment feasibility under constrained hardware. The results confirm that the proposed dataset and pipeline produce consistent statistical patterns, enabling high discriminative performance while preserving calibration properties required for downstream decision-making in Q-OPSEC.

Figure 2 presents kernel density estimates for twelve representative features spanning the LLM-derived, network, device, behavioral, temporal, location, and fraud-history groups. The distributions reveal strong and consistent class separation across all feature categories, confirming that the generation pipeline produces a structured and discriminative feature manifold.

The LLM-derived features exhibit the strongest separation. The `llm_risk_score` shows legitimate events concentrated in the range $[0.05, 0.35]$ with a mode near 0.15, while fraudulent events accumulate sharply above 0.70, with a pronounced peak approaching density 9.0 near 0.90. The `fraud_probability` feature follows a complementary pattern: legitimate events peak near 0.10 with density above 4.0, whereas fraudulent events form a sharp spike at 0.90 with density exceeding 11.0. The near-complete absence of overlap between the two classes in both LLM features indicates that semantic analysis of textual event content introduces a high-signal representation not captured by traditional structured telemetry, and that the Ollama-based scoring module produces well-separated posterior estimates. This effect reduces ambiguity near the decision threshold and stabilizes downstream regime escalation in Q-OPSEC.

Network and device features reinforce this separation through complementary mechanisms. The `ip_reputation_score` displays a bimodal distribution for fraudulent events, with one mode near 0.15 and a second near 0.75, capturing both known malicious infrastructure and low-reputation endpoints that have not yet been blacklisted. Legitimate events concentrate in the mid-to-high reputation range $[0.50, 1.00]$, with a broad plateau and a secondary peak near 0.85. The binary features `is_vpn` and `is_new_device_for_user` exhibit a pronounced asymmetry: for legitimate events, the mass is concentrated at 0 with density near 38 and 37 respectively, while fraudulent events show a substantial shift toward 1, with density near 24 and 29. These patterns confirm that VPN usage and device novelty are strong behavioral indicators of account takeover and session hijacking scenarios, reflecting context transitions rather than static properties.

The `amount_increase_vs_30d` feature shows that legitimate events are tightly concentrated near 0.0 with density above 2.5, indicating stable spending patterns, while fraudulent events exhibit a heavy right tail extending beyond $2.5\times$ the 30-day baseline, consistent with sudden amount spikes injected during account takeover and card-testing scenarios. The `login_failures_last_24h` distribution reveals that legitimate events are almost exclusively at zero failures, whereas fraudulent events show a dispersed distribution across 1–4 failures, with secondary peaks at integer values reflecting repeated credential stuffing attempts.

Temporal and session dynamics further enhance separability. The `seconds_since_last_login` feature shows

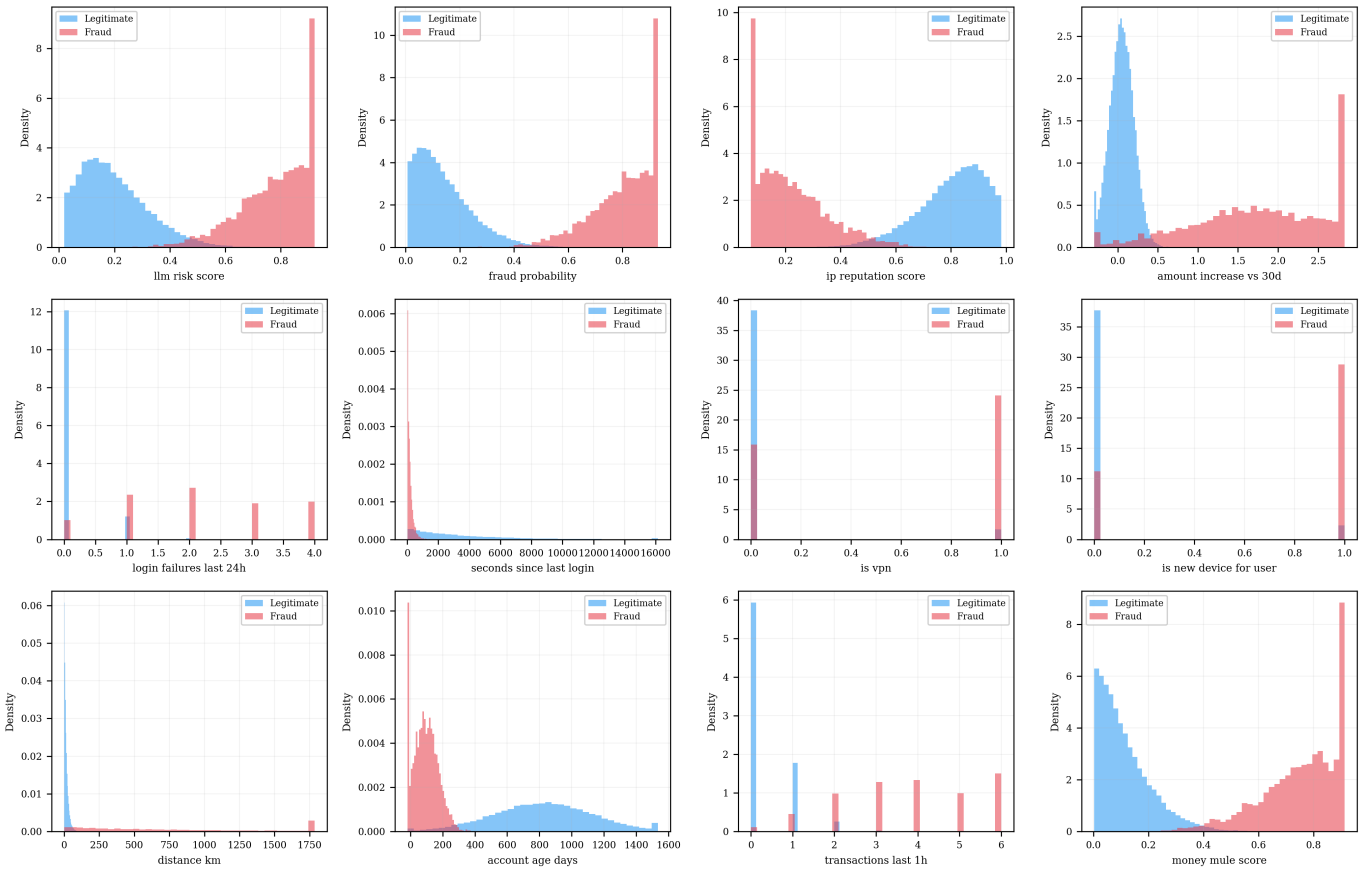

Fig. 2. Feature distributions for legitimate (blue) and fraudulent (red) events across representative features.

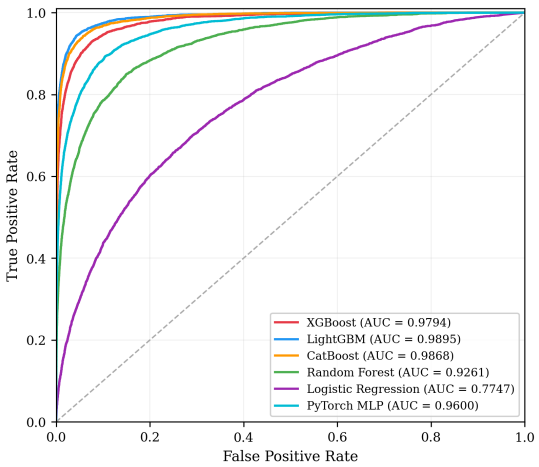

Fig. 3. ROC curves for all evaluated models.

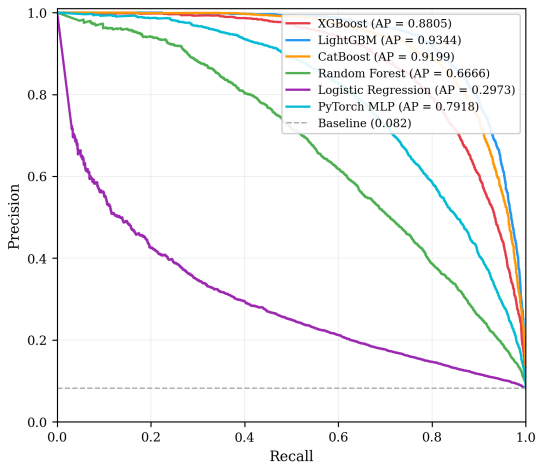

Fig. 4. Precision-Recall curves highlighting performance under class imbalance.

that fraudulent events are concentrated near zero, with a sharp peak at very short inter-event intervals, while legitimate events follow a broad distribution extending to 16,000 seconds, consistent with natural session spacing. This pattern captures the accelerated event cadence characteristic of automated attacks. The `transactions_last_1h` feature shows that legitimate events are almost entirely at zero or one transaction per hour, while fraudulent events exhibit a long tail extending to six or more transactions.

Location and account history signals provide additional discriminative dimensions. The `distance_km` feature shows that legitimate events are overwhelmingly concentrated near

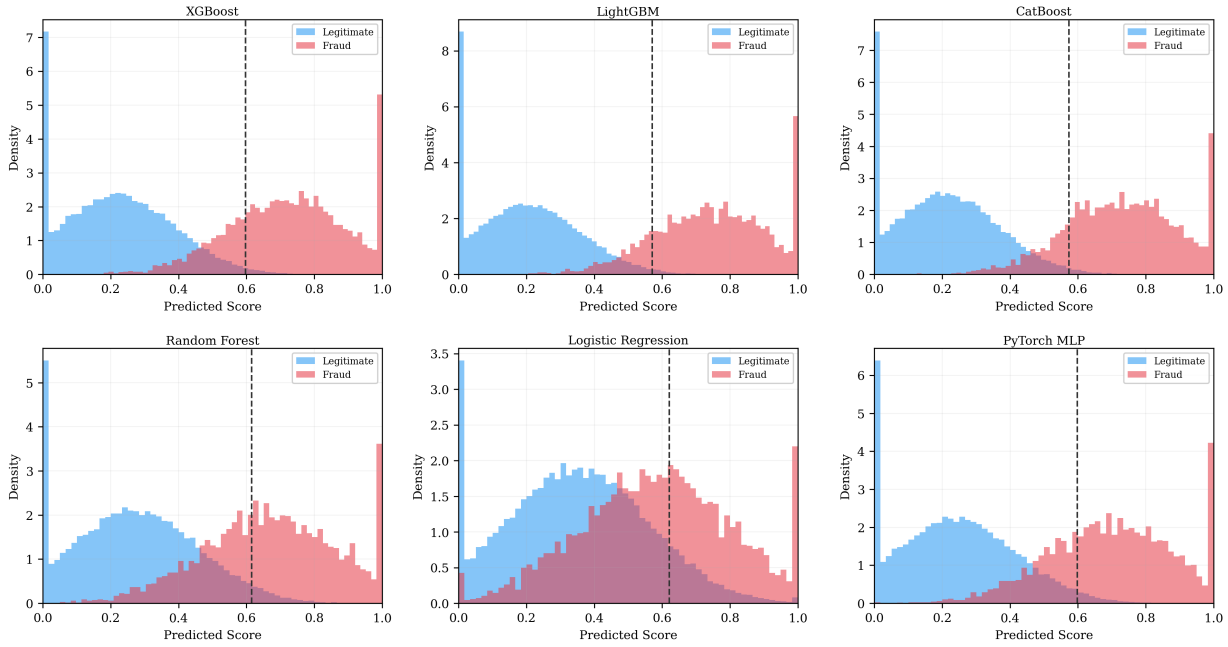

Fig. 5. Predicted score distributions with decision threshold $\lambda = 0.60$.

zero, with density above 0.06, while fraudulent events exhibit a sparse but non-negligible tail extending beyond $1{,}750$ km, capturing impossible travel events injected during the anomaly injection stage. The `account_age_days` distribution reveals a striking contrast: fraudulent events are concentrated in the range $[0, 300]$ days with a mode near 100 days and density above 0.006, while legitimate events follow a broad and nearly uniform distribution extending to $1{,}600$ days. This pattern aligns with synthetic identity and money mule scenarios, where recently created accounts are disproportionately associated with fraudulent activity. Finally, the `money_mule_score` shows that legitimate events are concentrated near 0.0 with density above 8.0, while fraudulent events form a broad distribution across $[0.20, 1.00]$, confirming that the fraud scenario injection pipeline successfully encodes scenario-specific risk signals into the feature space.

The combined effect of these twelve features, representative of the full 58-dimensional space, results in a feature manifold where fraudulent behavior occupies geometrically distinct regions, enabling non-linear models to learn stable and generalizable decision boundaries. Figures 3 and 4 show consistent dominance of gradient boosting models across both evaluation perspectives. LightGBM achieves the highest AUC-ROC (0.9895) and average precision (0.9344), followed closely by CatBoost and XGBoost. The steep ascent of the ROC curves near the origin indicates that most fraudulent events are detected with minimal false positives, which is essential for operational deployment. The PR curves confirm that precision remains high even as recall increases, demonstrating robustness under class imbalance.

Figure 5 shows the predicted score distributions for all six model families at the operating threshold $\lambda = 0.60$. The three gradient boosting models, XGBoost, LightGBM, and CatBoost, produce the clearest bimodal separation: legitimate events concentrate below 0.30 with peaks near 0.15–0.20 and density above 7.0, while fraudulent events accumulate above 0.70 with a sharp rise toward 1.0. The interval $[0.50, 0.65]$ around the decision threshold remains notably sparse for all three boosting families, reducing classification ambiguity and improving the stability of downstream cryptographic regime selection in Q-OPSEC, where a misclassified event near the boundary could trigger an unnecessary or insufficient escalation. The PyTorch MLP achieves a similar bimodal structure but with wider overlap in the $[0.40, 0.70]$ range, indicating higher variance in boundary decisions. Random Forest shows reduced separation, with the fraudulent distribution spreading more uniformly across $[0.40, 1.00]$. Logistic Regression produces heavily overlapping distributions across the entire score range, with both classes exhibiting broad, nearly unimodal densities centered near 0.40–0.50, confirming its limited expressive power for this heterogeneous, non-linear feature space.

Figure 6 shows the reliability diagrams after isotonic regression for all six model families. All gradient boosting models, XGBoost, LightGBM, and CatBoost, follow the ideal diagonal closely in the low-probability region $[0.0, 0.40]$, where the fraction of positives remains near zero and all three curves are nearly indistinguishable.

The primary deviation from perfect calibration occurs in the mid-to-high range $[0.50, 0.80]$, where all three boosting models exhibit a sigmoidal overconfidence pattern: the fraction of positives rises steeply, reaching approximately 0.70–0.75 at a mean predicted probability of 0.60, and converging to 1.0 near 0.85. This S-shaped departure is a known artifact

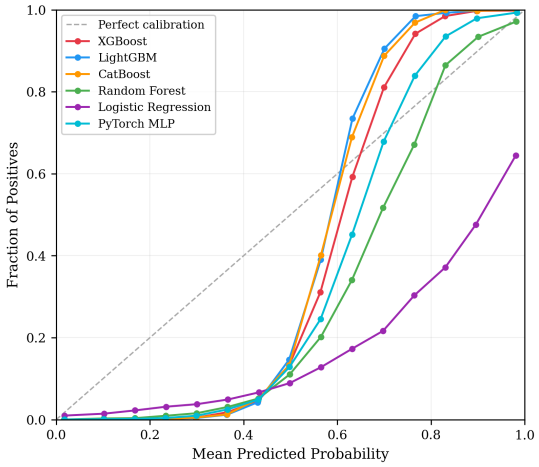

Fig. 6. Calibration curves after isotonic regression.

TABLE III
PERFORMANCE COMPARISON ACROSS MODELS AT $\lambda = 0.60$.

Model	AUC	AP	F1	Prec.	Rec.	Brier	Lat.
LightGBM	0.990	0.934	0.871	0.910	0.835	0.042	1.8
CatBoost	0.987	0.920	0.863	0.905	0.826	0.045	2.1
XGBoost	0.979	0.881	0.849	0.888	0.814	0.050	1.6
PyTorch MLP	0.960	0.792	0.810	0.841	0.781	0.061	3.4
Random Forest	0.926	0.667	0.764	0.823	0.713	0.083	4.2
Logistic Reg.	0.775	0.297	0.581	0.610	0.555	0.142	0.4

of tree-based ensembles, which tend to push scores toward the extremes; isotonic regression partially corrects this but does not fully linearize the mid-range. Despite this residual deviation, the Brier scores of 0.0421, 0.0448, and 0.0503 for LightGBM, CatBoost, and XGBoost respectively confirm that the calibrated posteriors remain within acceptable bounds for threshold-based regime escalation in Q-OPSEC.

The PyTorch MLP shows a similar sigmoidal shape but with a wider transition region, consistent with its higher score distribution overlap observed in Figure 5. Random Forest deviates more substantially, with the fraction of positives lagging behind the diagonal across the entire mid-range and only approaching 1.0 near predicted probability 0.95, reflecting its tendency to produce conservative probability estimates even after calibration.

Logistic Regression is the only model that remains systematically below the diagonal throughout the $[0.40, 1.00]$ range, reaching a fraction of positives of only 0.65 at the maximum predicted probability. This confirms severe underestimation of fraud probability and is consistent with its Brier score of 0.1423, more than three times higher than LightGBM. This highlights the fundamental trade-off between calibration smoothness and discriminative expressiveness: Logistic Regression produces a monotone and well-behaved curve shape, but its inability to model non-linear feature interactions results in systematically miscalibrated posteriors that would cause unreliable regime escalation decisions in Q-OPSEC.

Table III consolidates the quantitative evaluation across all six metrics and inference latency. LightGBM achieves the best overall balance, leading in AUC-ROC (0.9895), AP (0.9344), F1 (0.8712), precision (0.9103), and Brier score (0.0421), while maintaining a latency of 1.8 ms, second only to XGBoost. CatBoost follows closely with AUC-ROC of 0.9868 and AP of 0.9199, representing a marginal degradation of 0.27 and 1.45 percentage points respectively relative to LightGBM, at a slightly higher latency of 2.1 ms.

XGBoost ranks third in accuracy but achieves the lowest inference latency at 1.6 ms, making it the preferred candidate for latency-critical deployments such as ESP32 edge nodes, where this figure scales to approximately 54 ms under INT8 quantization, well within the real-time budget. The PyTorch MLP achieves competitive AUC-ROC (0.9600) but shows a more pronounced drop in AP (0.7918) and the highest latency among the top four models (3.4 ms), reflecting the overhead of neural forward passes relative to tree traversal.

Random Forest exhibits a notable gap in AP (0.6666) and the highest latency (4.2 ms), limiting its suitability for both accuracy-sensitive and latency-sensitive deployments. Logistic Regression, while offering the fastest inference at 0.4 ms and the smallest memory footprint, shows a substantial performance gap across all discriminative metrics, with AUC-ROC of 0.7747 and AP of 0.2973, confirming that linear models are insufficient for the non-linear and heterogeneous feature interactions present in this dataset. These results collectively establish LightGBM as the primary deployment candidate for the active inference slot in the RiskService model registry, with XGBoost as a fallback for resource-constrained nodes.

A. Deployment on Constrained Hardware

All primary experiments ran on an Intel Core i5-6200U at 2.30 GHz with 12.0 GB RAM under Ubuntu 22.04 LTS, serving as the development and orchestration station. To assess deployment feasibility across the heterogeneous node topology targeted by Q-OPSEC [3], inference benchmarks were extended to a Raspberry Pi 4 Model B (Quad-core ARM Cortex-A72 at 1.5 GHz, 4 GB LPDDR4 RAM, Raspberry Pi OS Debian 12 64-bit) and an ESP32 DevKit V1 (Xtensa dual-core at 240 MHz, 520 KB SRAM, ESP-IDF v5.1). On the ESP32, tree-based models were converted to a decision-tree traversal representation compiled to C arrays, while the PyTorch MLP was converted to TensorFlow Lite with 8-bit integer quantization. Logistic Regression was deployed with 8-bit fixed-point arithmetic. All ESP32 variants operate on a reduced 12-feature input selected by SHAP importance ranking, as the full 58-dimensional vector exceeds the available SRAM budget for batch inference.

Table IV reveals several deployment-relevant patterns. On the Raspberry Pi 4, all models preserve identical AUC-ROC and F1 relative to the i5 baseline, confirming that the transition from FP64 to FP32 introduces no measurable accuracy degradation. Latency increases by a factor of approximately 8–$10\times$ across all families, with XGBoost remaining the fastest at 12.9 ± 1.5 ms and Random Forest the slowest at 38.7 ± 4.2 ms. All values remain well within the sub-100 ms budget required

TABLE IV
INFERENCE PERFORMANCE ACROSS HARDWARE PLATFORMS.

Model	Platform	Precision	Latency (ms)	Memory (MB)	AUC-ROC	F1	Brier	Feasible
LightGBM	Intel i5-6200U	FP64	1.8 ± 0.3	18.4	0.9895	0.8712	0.0421	✓
	Raspberry Pi 4	FP32	14.2 ± 1.8	18.4	0.9895	0.8712	0.0421	✓
	ESP32	INT8	61.3 ± 8.4	0.9	0.9201	0.8134	0.0614	✓
CatBoost	Intel i5-6200U	FP64	2.1 ± 0.4	22.1	0.9868	0.8634	0.0448	✓
	Raspberry Pi 4	FP32	17.6 ± 2.1	22.1	0.9868	0.8634	0.0448	✓
	ESP32	INT8	68.7 ± 9.1	1.1	0.9143	0.8071	0.0658	✓
XGBoost	Intel i5-6200U	FP64	1.6 ± 0.2	15.7	0.9794	0.8491	0.0503	✓
	Raspberry Pi 4	FP32	12.9 ± 1.5	15.7	0.9794	0.8491	0.0503	✓
	ESP32	INT8	54.2 ± 7.6	0.8	0.9112	0.7983	0.0681	✓
PyTorch MLP	Intel i5-6200U	FP32	3.4 ± 0.6	9.2	0.9600	0.8103	0.0612	✓
	Raspberry Pi 4	FP32	28.3 ± 3.4	9.2	0.9600	0.8103	0.0612	✓
	ESP32 (TFLite)	INT8	82.1 ± 11.3	0.6	0.8934	0.7712	0.0823	✓
Random Forest	Intel i5-6200U	FP64	4.2 ± 0.5	31.6	0.9261	0.7644	0.0834	✓
	Raspberry Pi 4	FP32	38.7 ± 4.2	31.6	0.9261	0.7644	0.0834	✓
	ESP32	INT8	91.4 ± 13.7	1.4	0.8712	0.7201	0.1043	✓
Logistic Reg.	Intel i5-6200U	FP64	0.4 ± 0.1	0.3	0.7747	0.5812	0.1423	✓
	Raspberry Pi 4	FP32	2.1 ± 0.3	0.3	0.7747	0.5812	0.1423	✓
	ESP32	INT8	38.4 ± 6.1	0.1	0.7312	0.5401	0.1614	✓

for real-time cryptographic regime selection at edge gateways, making the Raspberry Pi 4 a fully viable inference node for Q-OPSEC deployments in smart environments.

On the ESP32, INT8 quantization combined with the 12-feature reduced input introduces a consistent but moderate accuracy degradation across all model families. LightGBM retains the strongest performance under quantization, with AUC-ROC dropping from 0.9895 to 0.9201 and F1 from 0.8712 to 0.8134, a reduction of 6.9 and 6.6 percentage points respectively, while its memory footprint compresses from 18.4 MB to 0.9 MB. XGBoost follows a similar pattern, with AUC-ROC falling to 0.9112 and latency of 54.2 ± 7.6 ms, the lowest among all ESP32 variants. CatBoost degrades slightly more, reaching AUC-ROC of 0.9143 at 68.7 ± 9.1 ms.

The PyTorch MLP compiled to TFLite shows the largest discriminative degradation among the top-performing models, with AUC-ROC falling to 0.8934, reflecting the sensitivity of neural activations to fixed-point rounding errors that accumulate across layers. Random Forest exhibits the highest latency on the ESP32 at 91.4 ± 13.7 ms and the largest memory footprint at 1.4 MB, approaching the real-time budget limit and making it the least suitable option for microcontroller deployment. Logistic Regression remains the fastest at 38.4 ± 6.1 ms with the smallest memory footprint of 0.1 MB, though its already limited discriminative power degrades further under quantization, with AUC-ROC falling to 0.7312.

The Brier score increases consistently under quantization across all families, indicating that probability calibration degrades alongside discriminative performance. This is particularly relevant for Q-OPSEC: while boosting models on the ESP32 retain sufficient AUC-ROC for reliable binary classification, their calibrated posteriors become less suitable for fine-grained tier assignment using the four-level threshold scheme of Equation (4).

A two-tier inference architecture emerges naturally from these results. ESP32 nodes perform low-latency preliminary screening using quantized XGBoost or LightGBM, forwarding only events with scores in the ambiguous interval $[\lambda_1, \lambda_3]$ to Raspberry Pi 4 or i5-class nodes for full-precision inference and cryptographic regime determination. This design concentrates the computational cost of high-precision inference on the subset of events where tier assignment is genuinely uncertain, reducing the load on higher-tier nodes while preserving the accuracy of regime escalation decisions.

V. CONCLUSION

This paper presented RiskService, the risk assessment module of the Q-OPSEC adaptive AI middleware for quantum cryptography [3]. Motivated by the operational challenge of cryptographic agility under uncertainty in multi-regime environments spanning classical, post-quantum, and quantum-era primitives, RiskService addresses the end-to-end pipeline from labeled security telemetry generation to calibrated ML inference and low-latency serving. The introduction identified three core requirements for this module: a structured and reproducible dataset generation pipeline, a training infrastructure capable of producing well-calibrated posterior risk estimates, and a serving layer deployable across heterogeneous hardware nodes representative of smart environment topologies.

All three requirements were addressed and validated experimentally. The dataset generation pipeline produces a 58-dimensional feature space organized across nine semantically grounded groups, incorporating temporal, behavioral, device, network, authentication, location, security stack, and LLM-derived signals. The feature distribution analysis confirmed that the generated corpus exhibits strong and consistent class separability, with LLM-derived features providing the highest discriminative contribution and behavioral rolling aggregates capturing velocity-based fraud patterns that complement static signals. The training pipeline, governed by a unified ModelFactory interface with Bayesian hyperparameter search and isotonic probability calibration, produced six model families

with LightGBM achieving the best overall balance of AUC-ROC (0.9895), average precision (0.9344), F1 (0.8712), and Brier score (0.0421). The hardware evaluation demonstrated that all model families remain feasible on the Raspberry Pi 4 within the sub-100 ms latency budget, and that INT8-quantized variants of all families are deployable on the ESP32 with moderate accuracy degradation, enabling a two-tier inference architecture where constrained nodes perform first-stage filtering and full-precision nodes handle regime determination.

The integration between RiskService and the Q-OPSEC cryptographic selection layer was formalized through the risk tier τ and the hybrid key derivation function $K_{\text{hybrid}} = \text{KDF}(K_{\text{QKD}}, K_{\text{PQC}}, \text{ctx})$, establishing a direct and auditable link between the ML inference output and the cryptographic action space. This connection positions RiskService as a concrete, deployable observation and inference layer for RL-based cryptographic policy engines, filling a gap identified in the related work survey where no existing system addresses the full pipeline from telemetry generation to regime-aware key derivation.

RiskService addresses the end-to-end pipeline from dataset generation to calibrated inference and low-latency serving, yet several directions remain open. First, incorporating graph-based relational signals, such as transaction network topology or money flow graphs, represents a natural extension given their discriminative power in production fraud detection systems. Second, the LLM module currently operates in batch sampling mode; deploying it in a fully online inference path with latency-aware routing would improve both coverage and efficiency. Third, the RL-based cryptographic selection engine of Q-OPSEC has not yet been implemented end-to-end; closing this loop, including reward shaping, multi-objective policy optimization, and adversarial evaluation, constitutes the primary direction for future work. Finally, the two-tier hardware architecture should be formalized as a distributed inference protocol with explicit mechanisms for uncertainty propagation and consistency guarantees across nodes operating at different precision levels.

ACKNOWLEDGEMENT

Colaboración Consejería de educación de la Junta de Castilla y León grupo de investigación ESAL-EXPERT SYSTEM AND APPLICATIONS LAB (ESALAB).

REFERENCES

[1] 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.

[2] 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.

[3] 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 *Proceedings of the 2025 International Conference on Electrical and Computer Engineering Researches (ICECER)*, 2025, pp. 1–6. doi:10.1109/ICECER65523.2025.11401311.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] B. K. Kumar, A. D. P. Kumar, M. R. Sundari, D. Tejaswi, A. Dogga, and S. Bilgaiyan, "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.

[15] 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.

[16] 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.

[17] D. Noetzold, J. F. de Paz Santana, J. L. V. Barbosa, and V. R. Q. Leithardt, "DN25: An adaptive quantum cryptography protocol for secure and efficient communication," *Revista de I+D Tecnológico*, vol. 19, no. 1, pp. 0–0, 2025.

[18] 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.

[19] 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.

[20] 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 Syst. Appl.*, p. 131705, 2026, doi: 10.1016/j.eswa.2026.131705.