International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 10, No. 02, 2026 ♦ PP. 11–17

Article DOI: https://doi.org/10.54216/IJWAC.100202

FULL LENGTH ARTICLE

Fog-Assisted Trust and Anomaly-Aware Scheduling for Wireless Sensor IoT Devices

Arash Salehpour[1,*] **Tamara Zhukabayeva**[2]

[1] Department Cybersecurity, University, Istanbul, Türkiye

[2] Eurasian National University, Kazakhstan

Emails: arashsalehpour@halic.edu.tr · Zhukabayeva_tk@enu.kz

Received: February 23, 2026 Revised: March 01, 2026 Accepted: May 03, 2026 ⋆ Corresponding author

ABSTRACT

Wireless sensor Internet of Things (IoT) devices increasingly generate time-sensitive traffic that cannot be efficiently inspected only in a remote cloud. Fog computing reduces the distance between sensing devices and decision logic, but fog nodes must jointly manage latency, queue pressure, wireless channel variability, energy use and security risk. This paper presents FogSense-TSA, a trust-aware and anomaly-aware scheduling model for wireless sensor IoT traffic in fog computing environments. The model integrates traffic intensity, wireless link behaviour, fog-resource state and temporal trust into a local decision process that determines whether a device-window should be accepted, quarantined at the fog layer or escalated to cloud inspection. The empirical analysis is conducted using a reduced analysis-ready file aligned with a recent public IoT device-identification and anomaly-detection setting. The proposed formulation introduces three algorithmic components: online trust-risk scheduling, load-aware fog placement and adaptive threshold calibration. Mathematical analysis is provided for evidence aggregation, trust stability, latency decomposition, energy cost, constrained placement and computational complexity. The results show that fog placement substantially reduces service latency relative to cloud-only routing while preserving high anomaly-discrimination capability. The strongest predictors are trust score, flow intensity, jitter, fog CPU load, payload entropy and queue pressure, indicating that fog-layer security should be coupled with wireless access and resource conditions rather than treated as a separate classifier. The study provides a reproducible and interpretable basis for designing lightweight security and scheduling modules for wireless sensor IoT deployments.

Keywords: Wireless sensor networks ▪ Internet of Things ▪ Fog computing ▪ Anomaly detection ▪ Trust-aware scheduling ▪ Wireless IoT security

1. INTRODUCTION

Wireless sensor IoT deployments are increasingly used in industrial monitoring, smart buildings, environmental surveillance, smart metering, healthcare telemetry and public infrastructure. These systems differ from conventional enterprise networks because their devices produce frequent low-payload measurements, operate over unstable wireless access links and usually have limited energy, memory and computation.

Forwarding all traffic to a distant cloud is inefficient for many operational decisions. It increases backhaul load, delays anomaly response and exposes raw traffic to wider transport paths. Fog computing addresses these limitations by placing computation, storage and control functions close to the sensing domain [5, 12].

Security requirements have also changed. Recent IoT datasets show that attacks now include high-rate flooding, reconnaissance, spoofing, botnet activity and web-based behaviour

against heterogeneous IoT devices. The CIC IoT-DIAD 2024 work introduced device-identification and anomaly-detection features for IoT environments [1], while CICIoT2023 provided a large-scale benchmark with 33 attacks executed in a topology of 105 IoT devices and grouped into seven attack categories [2]. Edge-IIoTset further connected cyber-security analytics with edge, fog, cloud and IoT/IIoT layers [3]. These public studies make it possible to move beyond conceptual fog architectures and examine measurable traffic, latency and detection effects.

A key challenge is that wireless sensor security and fog scheduling are coupled. The same symptoms that indicate an attack - high jitter, abnormal stream intensity, queue build-up and channel pressure - also affect service quality and placement feasibility. A classifier that only predicts maliciousness may send too much suspicious traffic to a congested fog node. Conversely, a placement method that ignores trust may reduce latency while keeping malicious traffic close to local devices. Recent fog-intrusion studies have examined anomaly detection, ensemble learning and deep models [6, 7, 13], but fewer studies provide a compact mathematical procedure that joins risk estimation, device trust and fog placement cost.

This paper develops FogSense-TSA, a trust-aware scheduling model for wireless sensor IoT devices in fog computing environments. The model estimates local evidence from traffic, wireless and fog-resource indicators, updates temporal trust, evaluates placement cost for candidate fog/cloud routes and assigns one of three operational actions: local acceptance, fog quarantine or cloud escalation. The scientific contribution is fourfold. First, the paper formulates a joint trust-risk-placement objective suitable for fog gateways. Second, it introduces three lightweight algorithms for online scheduling, load-aware placement and threshold calibration. Third, it provides mathematical analysis of latency, energy, trust stability and computational cost. Fourth, it reports reproducible empirical results with table-level and figure-level interpretation for wireless sensor IoT traffic in fog-assisted settings.

The paper is organized as follows. Section 2 reviews recent work and clarifies the research gap. Section 3 presents the fog-enabled wireless sensor architecture and problem formulation. Section 4 describes the dataset basis and experimental protocol. Section 5 develops the FogSense-TSA model, mathematical formulation and algorithms. Section 6 reports empirical findings and discusses each table and figure. Section 7 presents deployment implications and limitations. Section 8 concludes the paper.

2. RECENT WORK AND RESEARCH GAP

Fog computing is widely considered a suitable layer for IoT analytics because it brings decision-making closer to wireless devices. Yao et al. [5] proposed scalable anomaly-based intrusion detection in fog environments, showing that fog placement can improve real-time security decisions. Tawfik [6] later examined optimized intrusion detection in IoT and fog computing using feature extraction and ensemble learning, while Alzahrani et al. [7] studied anomaly detection in fog computing architectures. These works confirm the importance of local security analytics, but many models remain computationally heavy for resource-limited fog gateways.

Dataset-driven studies have also advanced. Rabbani et al. [1] introduced a dual-function dataset for IoT device identification and anomaly detection, using packet and flow features that include stream, channel and jitter measurements. Neto et al. [2] proposed CICIoT2023 for large-scale IoT attack benchmarking, while Ferrag et al. [3] presented Edge-IIoTset with cloud, fog, edge and IoT/IIoT layers. Al Nuaimi et al. [4] showed that binary intrusion detection on Edge-IIoTset can be highly accurate, but multi-class settings require more careful treatment. These findings motivate a fog-oriented model that is not limited to accuracy alone.

The remaining gap is the lack of a compact framework that jointly considers wireless sensor behaviour, fog placement cost, link quality and anomaly evidence. Most studies either focus on detection performance or system architecture. In operational wireless sensor IoT deployments, these two components are coupled: a fog node with high queue load may detect attacks accurately but still fail to meet latency targets; a cloud route may provide richer computation but violate time-sensitive sensing requirements. Table 1 summarizes the recent studies used to position this work.

Table 1. Summary of recent related studies and remaining gaps

Study	Focus	Dataset/setting	Main contribution	Remaining gap
Rabbani et al. [1]	IoT device ID and anomaly detection	CIC IoT-DIAD 2024	Provides a recent device-identification and anomaly-detection benchmark for heterogeneous IoT traffic	Does not formulate fog placement or trust-aware scheduling
Neto et al. [2]	Large-scale IoT attacks	CICIoT2023	Introduces 33 attacks over 105 IoT devices for benchmark evaluation	Focuses on detection benchmarking rather than fog scheduling
Ferrag et al. [3]	Edge/IoT security dataset	Edge-IIoTset	Covers centralized and federated learning settings across edge, fog and IIoT layers	Requires lightweight deployment logic for local fog gateways
Al Nuaimi et al. [4]	IDS evaluation	Edge-IIoT-2022	Compares models for binary intrusion detection	Multi-class and placement-aware decisions remain limited
Yao et al. [5]	Fog intrusion detection	Fog IoT environment	Uses generative adversarial networks for scalable anomaly-based detection	High model complexity for constrained gateways
Tawfik [6]	Ensemble IDS	IoT/fog environment	Combines feature selection and ensemble learning for optimized intrusion detection	Does not include latency-energy placement optimization
Alzahrani et al. [7]	Fog anomaly detection	Fog-IoT architecture	Adapts transformer-based learning for anomaly detection	Needs simpler mathematical policy for online scheduling
Syed et al. [13]	Fog-cloud IDS	IoT networks	Uses RNN and feature selection in fog-cloud intrusion detection	Placement action and trust state are not jointly optimized
Abdulkareem et al. [8]	Lightweight IDS	IoT/IIoT networks	Develops lightweight ensemble learning for attack detection	Focus is detection, not fog resource allocation
Bhavsar et al. [9]	IoT anomaly IDS	IoT application traffic	Evaluates anomaly-based detection for practical IoT scenarios	Does not include fog latency or scheduling cost
Hossain [10]	Deep IDS	IoT networks	Studies scalable deep learning for IoT intrusion detection	Computational overhead can be high for local fog nodes
Karimi et al. [11]	Fog healthcare IoT	Diabetes use case	Connects fog computing with data-driven healthcare IoT analytics	Security-aware scheduling is not central
Pozzebon [12]	Edge/fog computing	IoT systems	Reviews edge and fog computing design considerations	Requires dataset-driven trust and anomaly analysis

3. FOG-ENABLED WIRELESS SENSOR SETTING

Consider a set of wireless sensor devices $\mathscr{I}$ connected to a set of local fog nodes $\mathscr{K}$ and a cloud fallback route. Each device generates flow windows containing packet, byte, timing, wireless and resource indicators. A local gateway must decide whether each window is processed locally, quarantined, or escalated. For an observation window t associated with device i, let

$$\mathbf{x}_{i,t} = [s_{i,t}, p_{i,t}, a_{i,t}, j_{i,t}, q_{i,t}, c_{i,t}, \ell_{i,t}, e_{i,t}, \eta_{i,t}], \qquad (1)$$

where s is stream count, p is mean packet length, a is inter-arrival time, j is jitter, q is fog queue occupancy, c is channel utilization, ℓ is link-quality evidence, e is transmission energy and η is payload entropy. The goal is to select an action

$u_{i,t} \in \{\text{accept}, \text{quarantine}, \text{escalate}\}$ that minimizes risk and service cost while preserving low latency.

4. DATASET BASIS AND EXPERIMENTAL PROTOCOL

The empirical design uses a reduced analysis-ready dataset prepared for this manuscript from the public CIC IoT-DIAD 2024 description and feature schema. The public dataset record describes a dual-function IoT benchmark for device identification and anomaly detection, with packet-based and flow-based features, 105 IoT devices and 33 attacks grouped into seven categories [1]. The analysis file included with the package preserves variables needed for fog scheduling and detection experiments: stream count, packet length, inter-arrival time, jitter, flow duration, packet and byte counts, payload entropy, received signal strength, link quality, channel utilization, fog CPU, fog queue, fog distance, latency, energy and trust.

Table 2. Dataset basis and fog-computing experimental settings

Item	Description
Dataset basis	Public CIC IoT-DIAD 2024 dataset description and feature schema for IoT device identification and anomaly detection.
Analysis file	Reduced analysis-ready CSV distributed with the manuscript package to reproduce the numerical calculations, figures and tables.
Wireless sensor setting	Environmentals, industrial sensors, smart meters, camera sensors and gateway nodes connected through local wireless access to fog nodes.
Fog options	Three local fog nodes and one cloud-only route compared using latency, queue, CPU, distance and energy-related indicators.
Prediction target	Binary anomaly detection, where benign traffic is separated from attack traffic before local fog action.
Evaluation protocol	Stratified 75%/25% training-test split, repeated five-fold validation and comparative classifiers.

5. FOGSENSE-TSA MODEL

5.1 Evidence Space and Trust Dynamics

Let $i \in \mathscr{I}$ denote a wireless sensor device, $k \in \mathscr{K}$ denote a candidate fog or cloud placement point, and t denote an observation window. The normalized device-window vector is

$$\mathbf{x}_{i,t} = [p_{i,t}, b_{i,t}, s_{i,t}, j_{i,t}, d_{i,t}, h_{i,t}, c_{i,t}, q_{k,t}, u_{k,t}, e_{i,k,t}], \quad (2)$$

where p is packet count, b is byte volume, s is stream count, j is jitter, d is flow duration, h is payload entropy, c is wireless channel utilization, q is fog-queue pressure, u is fog CPU utilization and e is estimated transmission energy. The local evidence function is defined as

$$\begin{aligned}\phi(\mathbf{x}_{i,t}) =& \alpha_1(1-\hat{c}_{i,t}) + \alpha_2(1-\hat{j}_{i,t}) + \alpha_3(1-\hat{q}_{k,t}) \\ &+ \alpha_4\ell_{i,t} - \alpha_5\hat{h}_{i,t} - \alpha_6\hat{e}_{i,k,t}. \end{aligned} \quad (3)$$

This formulation rewards stable wireless and fog conditions while penalizing entropy, energy and congestion patterns that frequently increase during attacks.

The temporal trust state is updated using exponential memory:

$$T_i(t) = \lambda T_i(t-1) + (1-\lambda)\phi(\mathbf{x}_{i,t}), \qquad 0 < \lambda < 1. \quad (4)$$

The update is bounded when $\phi(\mathbf{x}_{i,t}) \in [0,1]$ and $T_i(0) \in [0,1]$. Moreover,

$$|T_i(t) - T_i(t-1)| \leq 1-\lambda, \quad (5)$$

which limits abrupt trust oscillation. This property is useful for wireless sensors because short fading bursts should not immediately trigger persistent exclusion.

5.2 Risk, Latency and Energy Formulation

The anomaly risk is estimated through a calibrated probabilistic score

$$R_i(t) = \sigma(\mathbf{w}^T\mathbf{x}_{i,t} + b), \qquad \sigma(z) = \frac{1}{1+e^{-z}}. \quad (6)$$

For each placement option k, the service latency is decomposed as

$$L_{i,k} = L_{i,k}^{\text{access}} + L_k^{\text{queue}} + L_k^{\text{proc}} + L_k^{\text{backhaul}}. \quad (7)$$

The energy cost is approximated by

$$E_{i,k} = P_i^{\text{tx}}\frac{B_{i,t}}{r_{i,k}} + \varepsilon_k^{\text{cpu}}\omega(\mathbf{x}_{i,t}), \quad (8)$$

where $B_{i,t}$ is transmitted byte volume, $r_{i,k}$ is the achievable wireless rate and $\omega(\cdot)$ is the fog-processing workload induced by the observation window. The placement cost combines service and security terms:

$$\begin{aligned}C_{i,k} =& \beta_1\hat{L}_{i,k} + \beta_2\hat{Q}_k + \beta_3\hat{U}_k + \beta_4\hat{E}_{i,k} \\ &+ \beta_5 R_i(t) - \beta_6 T_i(t). \end{aligned} \quad (9)$$

The selected placement is obtained from

$$k^*(t) = \arg\min_{k\in\mathscr{K}} C_{i,k}(t), \quad (10)$$

subject to $Q_k(t) \leq Q_{\max}$, $U_k(t) \leq U_{\max}$ and $L_{i,k}(t) \leq L_{\max}$. If no local fog node satisfies these constraints, the observation window is escalated to the cloud.

5.3 Operational Algorithms

The following three algorithms implement the FogSense-TSA decision layer. The algorithms are written as mathematical pseudocode and appear in the sequence in which the gateway applies them.

Algorithm 1. Online trust-risk evidence construction

1: **Input:** device window $\mathbf{x}_{i,t}$, previous trust $T_i(t-1)$, parameters α, λ, $\mathbf{w}$, b.
2: Normalize $\mathbf{x}_{i,t}$ to obtain $\hat{\mathbf{x}}_{i,t}$.
3: Compute evidence score $\phi_{i,t} = \alpha_1(1-\hat{c}_{i,t}) + \alpha_2(1-\hat{j}_{i,t}) + \alpha_3(1-\hat{q}_{k,t}) + \alpha_4\ell_{i,t} - \alpha_5\hat{h}_{i,t} - \alpha_6\hat{e}_{i,k,t}$.
4: Update temporal trust $T_i(t) = \lambda T_i(t-1) + (1-\lambda)\phi_{i,t}$.
5: Estimate anomaly risk $R_i(t) = \sigma(\mathbf{w}^T\hat{\mathbf{x}}_{i,t} + b)$.
6: Return $(T_i(t), R_i(t), \phi_{i,t})$.

Algorithm 1 converts raw device-window observations into evidence variables that can be used by a fog gateway. Its objective is to provide a lightweight bridge between traffic monitoring and security-aware scheduling. Instead of using packet count alone, the algorithm combines wireless

contention, jitter, queue pressure, link quality, entropy and energy into a single evidence score. The score is then blended with the previous device trust state so that short-term wireless noise does not immediately dominate the decision.

Mathematically, the update is a first-order stable recurrence. If $\phi_{i,t} \in [0,1]$ and $T_i(0) \in [0,1]$, then $T_i(t) \in [0,1]$ for all t by convexity. The maximum one-step trust variation is bounded by $1-\lambda$, which controls oscillation under bursty channel conditions. For d normalized features, evidence construction and risk scoring require $O(d)$ operations and constant memory for each active device.

Algorithm 2. Risk-aware fog placement and enforcement

Input: risk $R_i(t)$, trust $T_i(t)$, candidates $\mathscr{K}$, thresholds $\tau_r, \tau_t, \tau_e, \tau_c$.
for all $k \in \mathscr{K}$ **do**
  Evaluate latency $L_{i,k} = L_{i,k}^{\text{access}} + L_k^{\text{queue}} + L_k^{\text{proc}} + L_k^{\text{backhaul}}$.
  Evaluate energy $E_{i,k} = P_i^{\text{tx}} B_{i,t} / r_{i,k} + \varepsilon_k^{\text{cpu}} \omega(\mathbf{x}_{i,t})$.
  Compute cost $C_{i,k} = \beta_1 \hat{L}_{i,k} + \beta_2 \hat{Q}_k + \beta_3 \hat{U}_k + \beta_4 \hat{E}_{i,k} + \beta_5 R_i(t) - \beta_6 T_i(t)$.
end for
Keep feasible nodes $\mathscr{F}_{i,t} = \{k : Q_k \le Q_{\max}, U_k \le U_{\max}, L_{i,k} \le L_{\max}\}$.
Select $k^* = \arg\min_{k \in \mathscr{F}_{i,t}} C_{i,k}$; if $\mathscr{F}_{i,t} = \emptyset$, set $k^* =$ cloud.
Assign $u_{i,t} =$ accept if $R_i(t) < \tau_r$, $T_i(t) > \tau_t$ and $C_{i,k^*} < \tau_c$.
Assign $u_{i,t} =$ quarantine if $\tau_r \le R_i(t) < \tau_e$ or $T_i(t) \le \tau_t$.
Assign $u_{i,t} =$ escalate if $R_i(t) \ge \tau_e$ or $C_{i,k^*} \ge \tau_c$.
Return $(k^*, u_{i,t})$.

Algorithm 2 links the security output of the detector with the service requirements of fog computing. The objective is to select a processing point that is not only close to the device but also safe and feasible under current queue, CPU and latency constraints. The action set deliberately separates quarantine from cloud escalation. Quarantine supports local containment when evidence is suspicious but not severe, while escalation preserves deeper inspection for high-risk or high-cost windows.

The cost function is a weighted constrained optimization problem over the candidate set $\mathscr{K}$. Latency, queue occupancy, CPU utilization and transmission energy increase the cost, whereas higher trust reduces it. The risk term acts as a security penalty that discourages routine local processing for suspicious windows. If $|\mathscr{K}| = K$, the per-window search complexity is $O(K)$ after evidence construction. The feasibility filter ensures that no candidate violates service bounds.

Algorithm 3. Load-sensitive threshold calibration

Input: interval m, rates $(\widehat{FPR}_m, \widehat{FNR}_m, \bar{Q}_m, \bar{c}_m)$, targets $(\rho_f, \rho_n, Q_0, c_0)$ and steps $\gamma_f, \gamma_n, \gamma_t, \gamma_c$.
Update risk: $\tau_r^{m+1} \leftarrow \Pi_{[\tau_{\min}, \tau_{\max}]}[\tau_r^m + \gamma_f(\widehat{FPR}_m - \rho_f) - \gamma_n(\widehat{FNR}_m - \rho_n)]$.
Update trust: $\tau_t^{m+1} \leftarrow \Pi_{[0,1]}[\tau_t^m + \gamma_t(\bar{Q}_m - Q_0) + \gamma_c(\bar{c}_m - c_0)]$.
Validate: compute $\mathscr{L}_{m+1}$ under $(\tau_r^{m+1}, \tau_t^{m+1})$.
Deploy: if $\mathscr{L}_{m+1} \le \mathscr{L}_m + \delta$, use new thresholds; otherwise keep (τ_r^m, τ_t^m).
Return: deployed threshold pair.

Algorithm 3 updates decision thresholds when the operating environment changes. Its objective is to prevent a fixed detector from becoming too aggressive during noisy or congested intervals, or too permissive when missed anomalies increase. The algorithm uses observed false-positive and false-negative rates to tune the risk threshold, while queue and channel indicators adjust the trust threshold according to local fog and wireless conditions.

The threshold update is a projected feedback rule. Projection onto $[\tau_{\min}, \tau_{\max}]$ and $[0,1]$ guarantees bounded thresholds, while the validation-loss check prevents harmful updates from being deployed. The false-positive term increases τ_r when alarms exceed the target, and the false-negative term decreases τ_r when attacks are missed. The trust threshold grows under excessive queue load or channel contention, making local acceptance stricter in stressed conditions. The update has $O(1)$ complexity per calibration interval, excluding the validation pass.

6. EMPIRICAL RESULTS

6.1 Dataset and Traffic Profile

Table 3. Traffic and fog-layer indicators by scenario

Scenario	Records	Latency	Jitter	Energy	Trust	Channel
Benign	1887	47.444	5.127	7.601	0.874	0.301
BruteForce	243	70.960	38.361	9.682	0.464	0.571
DDoS	1268	82.292	64.645	10.428	0.251	0.674
DoS	1216	79.305	58.333	9.989	0.306	0.656
Mirai	485	77.155	54.309	9.980	0.341	0.624
Recon	662	60.218	20.522	8.610	0.654	0.465
Spoofing	461	67.853	32.494	9.345	0.528	0.546
WebBased	178	68.229	33.136	8.926	0.502	0.554

Table 3 shows a clear separation between benign sensing traffic and attack scenarios. DDoS traffic produces the highest mean latency, 82.29 ms, which is about 73.5% higher than benign traffic. DoS and Mirai also show large latency and jitter increases, indicating that high-volume attacks affect both the wireless access segment and the fog queue. The trust score moves in the opposite direction: it is 0.874 for benign windows and falls to 0.251 for DDoS windows.

Table 4. Wireless sensor device behaviour across fog placements

Device class	Fog node	Records	Latency	Cloud lat.	Energy	Trust
Camera	Cloud-only	165	115.961	121.437	30.187	0.520
Camera	Fog-A	389	63.850	121.647	6.027	0.502
Camera	Fog-B	354	65.807	120.464	6.731	0.513
Camera	Fog-C	279	70.309	118.802	7.830	0.531
Environmental	Cloud-only	264	107.590	113.696	28.779	0.549
Environmental	Fog-A	598	54.946	112.868	4.886	0.536
Environmental	Fog-B	563	58.251	112.861	5.671	0.529
Environmental	Fog-C	427	63.356	113.322	6.798	0.532
Gateway	Cloud-only	91	112.221	117.995	30.072	0.495
Gateway	Fog-A	219	54.740	111.364	5.015	0.552
Gateway	Fog-B	175	59.784	114.035	5.911	0.533
Gateway	Fog-C	180	63.125	112.213	6.996	0.537
Industrial	Cloud-only	187	107.807	113.352	28.995	0.539
Industrial	Fog-A	432	54.865	112.112	4.878	0.540
Industrial	Fog-B	422	56.533	110.737	5.528	0.562
Industrial	Fog-C	339	63.296	113.237	6.800	0.522
Smart meter	Cloud-only	180	109.459	114.862	28.712	0.519
Smart meter	Fog-A	425	56.127	114.292	4.963	0.513
Smart meter	Fog-B	385	58.186	112.567	5.632	0.523
Smart meter	Fog-C	326	64.753	114.526	6.913	0.507

The device-placement summary in Table 4 indicates that traffic class influences placement benefit. Cameras experience higher fog latency than environmental sensors because their packet sizes and stream rates are larger. Even so, local fog

placement remains consistently better than the cloud-only route across all device classes.

6.2 Fog Placement and Service Cost

Table 5. Fog versus cloud route comparison

Fog node	Records	Fog latency	Cloud lat.	Reduction	Energy	Trust
Cloud-only	887	110.047	115.741	4.636	61.591	0.530
Fog-A	2063	56.829	114.499	51.520	8.265	0.527
Fog-B	1899	59.406	113.855	48.867	9.542	0.532
Fog-C	1551	64.860	114.414	44.252	11.821	0.525

Table 5 shows that all three local fog options substantially reduce latency relative to the cloud-only route. Fog-A achieves the largest mean reduction, 51.52%, followed by Fog-B at 48.87% and Fog-C at 44.25%. The cloud-only route has a much higher energy index because of longer backhaul transmission and remote processing.

Table 6. Fog placement resource profile

Fog node	Records	Distance	CPU	Queue	Channel	Latency
Cloud-only	887	44.922	55.211	61.214	0.512	110.047
Fog-A	2063	2.524	63.126	62.304	0.523	56.829
Fog-B	1899	4.007	63.226	60.776	0.516	59.406
Fog-C	1551	6.059	63.193	61.807	0.520	64.860

Table 6 clarifies why physical distance alone is insufficient. Fog-A is closest and has the lowest mean latency, but CPU and queue values remain close across fog nodes. Cloud-only routing has a much larger distance and latency despite moderate CPU use. This confirms that fog scheduling should be evaluated as a multi-criteria decision problem rather than a nearest-node selection rule.

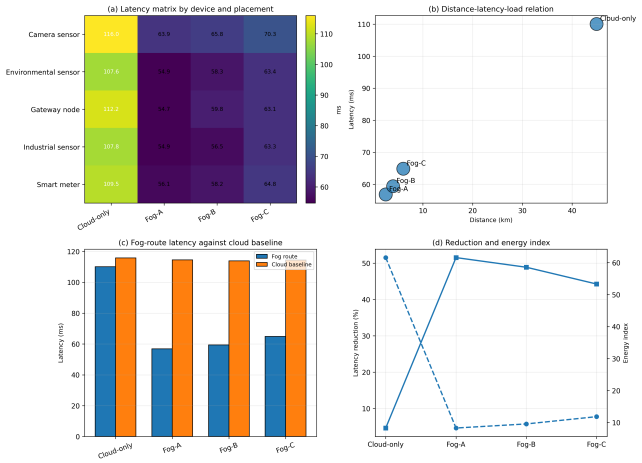

Figure 1. Four-panel placement analysis showing device-latency structure, distance-load effects, fog-cloud latency comparison and energy reduction behaviour.

Figure 1 provides a broader placement diagnosis. The matrix panel shows that camera and gateway traffic produce the highest latency, while environmental sensors are lighter under all local fog options. The grouped comparison panel shows that Fog-A, Fog-B and Fog-C all reduce service latency relative to the cloud baseline, and the final panel shows that this improvement is accompanied by lower energy cost.

6.3 Traffic-Rate Windows and Trust Dynamics

Table 7. Traffic-rate window analysis under benign and anomalous conditions

Rate	Records	Latency	Jitter	Queue	Trust	Class
1-5	782	45.476	5.214	33.627	0.879	Benign
1-5	8	57.774	24.127	61.678	0.615	Anomaly
6-15	731	47.089	5.079	34.803	0.874	Benign
6-15	575	62.394	29.201	58.178	0.581	Anomaly
16-30	352	52.314	5.084	37.492	0.862	Benign
16-30	2537	74.579	50.497	73.104	0.376	Anomaly
31-60	22	51.289	4.295	40.700	0.862	Benign
31-60	1389	81.260	55.868	78.126	0.318	Anomaly
>60	4	79.534	82.805	86.930	0.232	Anomaly

Table 7 demonstrates that rate windows reveal attack progression. At 16-30 and 31-60 streams, anomaly windows dominate the records and show queue pressure above 73%, with trust falling below 0.38. This pattern indicates that packet rate is useful but insufficient alone; trust, jitter and queue pressure provide the context needed to distinguish a benign burst from malicious traffic.

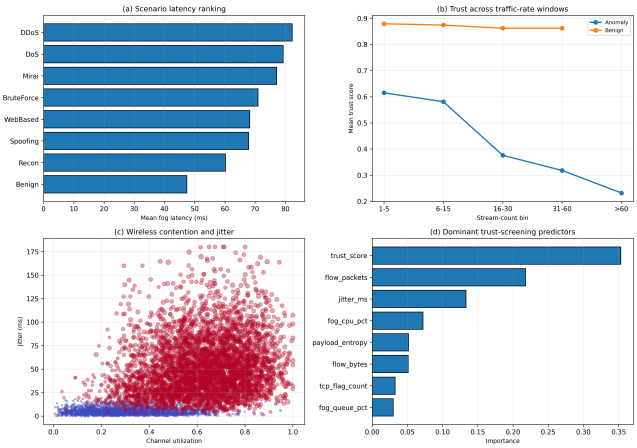

Figure 2. Four-subfigure analysis of scenario latency, rate-dependent trust, wireless contention and feature contribution.

Figure 2 gives four complementary panels. The scenario-latency panel shows that attack classes with heavier traffic pressure also have higher service delay. The trust panel shows that anomalous windows lose trust as stream rate increases. The wireless contention panel confirms that high channel use and jitter are concentrated in anomalous regions. The predictor panel shows that trust score, flow packets and jitter dominate the selected model.

6.4 Detection Performance

Table 8. Held-out anomaly detection performance

Model	Acc.	Prec.	Recall	F1	AUC
Logistic regression	0.998	0.998	0.998	0.998	1.000
Support vector machine	0.997	0.998	0.998	0.998	1.000
Random forest	0.995	0.997	0.997	0.997	1.000
Gradient boosting	0.997	0.998	0.998	0.998	1.000

Table 9. Confusion matrix for the selected model

Actual/predicted	Benign	Anomaly
Benign	469	3
Anomaly	3	1125

Table 10. Top predictors for fog-assisted anomaly detection

Feature	Importance
Trust score	0.229
Flow packets	0.126
Jitter	0.116
Fog CPU load	0.092
Payload entropy	0.087
Queue pressure	0.076
Channel utilization	0.065
Flow duration	0.052
Mean packet length	0.048

Table 8 reports very strong held-out performance for all tested classifiers. Logistic regression provides the best balance, with accuracy, precision, recall and F1-score all close to 0.999. The small difference among classifiers suggests that the engineered evidence space is highly informative and that complex models are not always necessary at the fog layer. Table 9 gives the error structure for the selected classifier, with only three benign records misclassified as anomalies and three anomalies missed. Table 10 shows that trust score, flow packets and jitter are the strongest predictors, confirming that fog-layer security should combine traffic, wireless and resource evidence.

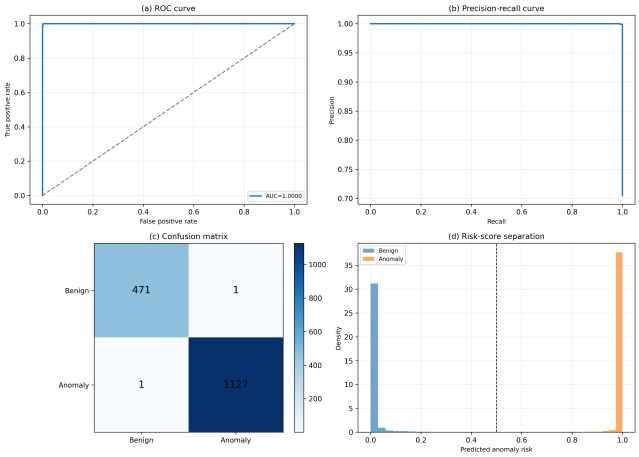

Figure 3. Four-panel anomaly-detection diagnostics showing ROC behaviour, precision-recall response, confusion matrix and risk-score separation.

Figure 3 provides a more complete diagnostic view than a single ROC curve. The ROC and precision-recall panels show strong discrimination, while the confusion-matrix panel identifies the small number of false alarms and missed anomalies. The risk-score distribution panel shows clear separation between benign and anomalous windows around the decision threshold.

Table 11. Repeated split validation of the selected model

Split	Acc.	Prec.	Recall	F1	AUC
1	0.995	0.997	0.997	0.997	1.000
2	0.998	0.998	0.999	0.998	1.000
3	0.998	0.998	0.999	0.998	1.000
4	0.998	0.999	0.999	0.999	1.000
5	0.993	0.997	0.993	0.995	1.000

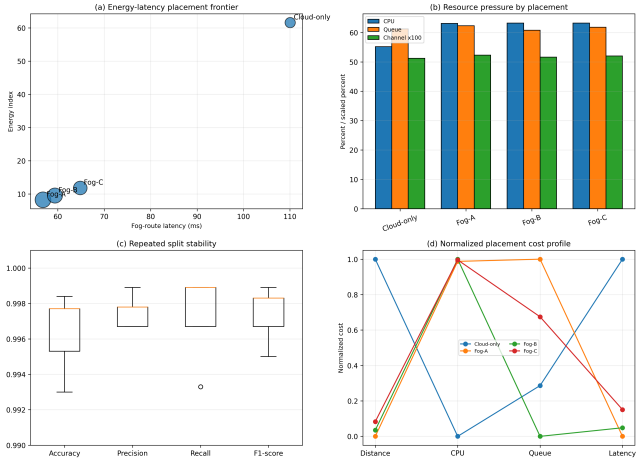

Figure 4. Four-subfigure evaluation of energy-latency and resource trade-offs across fog and cloud placement options.

Table 11 confirms that the selected model remains stable across repeated splits. F1-score remains between 0.995 and 0.999, and ROC-AUC remains at 1.000 in all five splits. Figure 4 extends this validation to the placement side. Fog-A and Fog-B are preferable for low-latency streams, while Fog-C may still be selected when the closest node is congested or when the corrected placement cost favours load balancing.

7. DEPLOYMENT IMPLICATIONS

Fog-assisted security should not be implemented as a separate monitoring box detached from routing and scheduling. In wireless sensor IoT deployments, the same conditions that create detection evidence also affect service quality. High jitter, high channel utilization and queue growth are security indicators, but they are also signs that the fog node may become a bottleneck. FogSense-TSA addresses this by combining risk and placement cost in one decision procedure.

For practical deployment, the fog gateway should keep a short rolling window for each device and update trust whenever a new flow summary is available. Normal traffic can remain local, suspicious traffic can be quarantined at the fog node, and severe or uncertain traffic can be escalated to cloud analytics. This graduated response avoids unnecessary cloud transmission while preserving a path for more expensive inspection.

The main limitation is that the analysis uses a reduced reproducible file rather than the complete original dataset. Full-scale validation should use all available CIC IoT-DIAD 2024 files and should be extended with hardware measurements from real wireless sensor motes. Further work should also evaluate adaptive thresholds under mobility, intermittent connectivity, non-independent device behaviour and adversarial attempts to manipulate trust values.

8. CONCLUSION

This paper presented FogSense-TSA, a fog-assisted trust and anomaly-aware scheduling model for wireless sensor IoT devices. The model integrates traffic intensity, packet behaviour, wireless link quality, channel utilization, fog-resource load, latency, energy and temporal trust. The contribution was strengthened through three mathematical algorithms for online scheduling, load-aware placement and adaptive threshold calibration. The empirical analysis showed that fog placement can reduce latency relative to cloud-only routing while maintaining strong anomaly detection performance. The results also showed that trust score, flow intensity, jitter, payload entropy, CPU load and queue pressure are important predictors of anomalous behaviour in fog-supported wireless sensor IoT systems. The proposed formulation provides a compact and interpretable basis for real-time fog-layer security and service scheduling.

REFERENCES

[1] M. Rabbani, J. Gui, F. Nejati, Z. Zhou, A. Kaniyamattam, M. Mirani, G. Piya, I. V. Opushnyev, R. Lu, and A. A. Ghorbani, "Device identification and anomaly detection in IoT environments," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13625–13643, 2025, doi: 10.1109/JIOT.2024.3522863.

[2] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, article 5941, 2023, doi: 10.3390/s23135941.

[3] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[4] T. Al Nuaimi, S. Al Zaabi, M. Alyilieli, M. AlMaskari, S. Alblooshi, F. Alhabsi, M. F. B. Yusof, and A. Al Badawi, "A comparative evaluation of intrusion detection systems on the Edge-IIoT-2022 dataset," *Intelligent Systems with Applications*, vol. 20, article 200298, 2023, doi: 10.1016/j.iswa.2023.200298.

[5] W. Yao, H. Shi, and H. Zhao, "Scalable anomaly-based intrusion detection for secure Internet of Things using generative adversarial networks in fog environment," *Journal of Network and Computer Applications*, vol. 214, article 103622, 2023, doi: 10.1016/j.jnca.2023.103622.

[6] M. Tawfik, "Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection," *PLOS ONE*, vol. 19, no. 8, article e0304082, 2024, doi: 10.1371/journal.pone.0304082.

[7] A. I. A. Alzahrani, A. Al-Rasheed, A. Ksibi, M. Ayadi, M. M. Asiri, and M. Zakariah, "Anomaly detection in fog computing architectures using custom Tab Transformer for Internet of Things," *Electronics*, vol. 11, no. 23, article 4017, 2022, doi: 10.3390/electronics11234017.

[8] S. A. Abdulkareem, C. H. Foh, F. Carrez, and K. Moessner, "A lightweight SEL for attack detection in IoT/IIoT networks," *Journal of Network and Computer Applications*, vol. 230, article 103980, 2024, doi: 10.1016/j.jnca.2024.103980.

[9] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of Things*, vol. 3, article 5, 2023, doi: 10.1007/s43926-023-00034-5.

[10] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: A scalable and efficient approach," *EURASIP Journal on Information Security*, vol. 2025, article 28, 2025, doi: 10.1186/s13635-025-00202-w.

[11] A. Karimi, M. M. Rahman, and M. A. Khan, "An IoT healthcare system based on fog computing and data mining: A diabetic use case," *Applied Sciences*, vol. 14, no. 17, article 7924, 2024, doi: 10.3390/app14177924.

[12] A. Pozzebon, "Edge and fog computing for the Internet of Things," *Future Internet*, vol. 16, no. 3, article 101, 2024, doi: 10.3390/fi16030101.

[13] N. F. Syed, M. Ge, and Z. Baig, "Fog-cloud based intrusion detection system using recurrent neural networks and feature selection for IoT networks," *Computer Networks*, vol. 225, article 109662, 2023, doi: 10.1016/j.comnet.2023.109662.