International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 10, No. 02, 2026 ♦ PP. 01–10

Article DOI: https://doi.org/10.54216/IJWAC.100201


FULL LENGTH ARTICLE

# Event-Selective Fog Microbatching for Wireless Sensor IoT Devices: A Data-Driven Study Using Edge-IIoTset Features

**Raden Aur Aachman Azakiyullah**[1,*] **Aiswan Aumanti**[2]

[1] Faculty of Science, Engineering and Technology, Department of Information System, Universitas Alma Ata, Yogyakarta, Indonesia
[2] Institut Bakti Nusantara, Lampung, Indonesia

**Emails:** nurrachmandzakiyullah@almaata.ac.id · mgumanti0205@gmail.com

Received: February 22, 2026 Revised: April 02, 2026 Accepted: May 05, 2026 ⋆ Corresponding author

## ABSTRACT

Wireless sensor IoT devices increasingly operate under strict energy, latency, and security constraints while generating high-frequency telemetry that cannot be forwarded continuously to remote clouds. This paper presents an event-selective fog microbatching model for wireless sensor streams in which local novelty scoring, fog-side buffering, risk-preserving retention, and energy-aware scheduling are jointly optimized. Unlike conventional anomaly-detection pipelines, the proposed method treats communication reduction as a primary design objective and binds it mathematically to attack-evidence preservation. A reduced feature-level experimental file following the public Edge-IIoTset label structure and selected network/sensor attributes is used to evaluate traffic selectivity, uplink reduction, fog latency, energy saving, and detection performance. The model assigns each observation window a novelty score, suppresses redundant low-information traffic, and groups retained events into load-aware microbatches at the nearest fog node. The proposed model is extended with stochastic retention bounds, microbatch-delay stability, radio-energy equations, and risk-constrained threshold calibration. Experimental results show that the design reduces uplink load and radio-energy consumption while preserving strong attack discrimination across distributed wireless sensor traffic. The findings support a broader use of fog computing as a selective communication-control layer for dense, security-sensitive wireless sensor IoT deployments.

**Keywords:** Wireless sensor IoT ▪ Fog computing ▪ Event-selective transmission ▪ Microbatching ▪ Edge-IIoTset ▪ Anomaly-aware scheduling

## 1. INTRODUCTION

Wireless sensor IoT deployments are increasingly used in industrial monitoring, smart buildings, precision agriculture, and critical infrastructure. These systems rely on small devices that periodically sample environmental or process variables and forward observations through short-range wireless links toward gateways, fog servers, or cloud analytics platforms. As the number of sensors grows, the main performance bottleneck is no longer only local computation; it is also the volume of repetitive telemetry sent over unreliable wireless channels. Continuous forwarding of every observation wastes energy, increases contention, and makes security analytics more difficult under bursty traffic conditions.

Fog computing offers a practical middle layer between sensors and the cloud. By moving processing closer to wireless gateways, fog nodes can aggregate, filter, and classify sensor traffic before it reaches wide-area networks. Recent fog-IoT reviews emphasize that distributed fog resources are especially useful for latency-sensitive IoT applications, since they reduce dependency on centralized cloud paths and improve the responsiveness of edge services [4, 6]. Task-offloading

surveys further show that edge and fog systems must jointly consider computation time, communication cost, energy lim-its, and queue stability rather than treating offloading as a simple binary decision [5]. However, many existing studies still use fog nodes mainly as small classifiers or offloading targets. Less attention is given to the question of which sensor observations should be forwarded at all.

Security pressure strengthens this question. Modern IoT at-tack datasets reveal large volumes of repeated benign records mixed with rare high-risk events [1, 2]. Full forwarding gives the fog node complete visibility, but it drains sensor batter-ies and increases channel contention. Aggressive sampling reduces the radio load, but it can remove early attack ev-idence, especially when abnormal traffic appears in short bursts. This tension is central to wireless sensor IoT environ-ments, where the first decision is often not which classifier to use but whether a window deserves transmission.

The present work studies this tension through event-selective fog microbatching. The core idea is that wireless sensor devices should transmit high-value observations at a high retention rate while suppressing redundant low-novelty win-dows. A fog node then groups retained events into short mi-crobatches that preserve detection quality but reduce packet overhead. This design is relevant to security-aware IoT en-vironments because abnormal flows are often rare, bursty, and expensive to miss. The Edge-IIoTset dataset provides a suitable basis for this direction because it was generated using a purpose-built IoT/IIoT testbed with sensors, proto-cols, and cloud-edge configurations [1]. Recent Edge-IIoTset evaluations further show that realistic IoT and IIoT intru-sion detection remains sensitive to feature selection, model complexity, and class imbalance [3, 8, 9].

This paper contributes a different perspective from conven-tional intrusion detection. First, it proposes an event-selective transmission model that computes a normalized novelty score from wireless, traffic, and sensor-drift evidence. Second, it introduces a fog microbatching rule that selects batch size from traffic intensity and queue load. Third, it develops a risk-preserving retention objective that links communica-tion reduction with anomaly recall. Fourth, it extends the mathematical analysis with radio-energy modeling, expected retention bounds, microbatch delay stability, and calibration complexity. Fifth, it reports a reproducible empirical analysis using an included reduced feature-level dataset, Python code, eight result tables, and compact four-panel figures. The paper is organized as follows. Section 2 reviews related work and positions the study. Section 3 formulates the system model. Section 4 describes the proposed method. Section 5 presents the dataset and experimental setup. Section 6 discusses the results. Section 7 concludes the paper.

## 2. LITERATURE POSITIONING

Fog computing has become an important design pattern for IoT systems because it reduces the latency and bandwidth cost associated with cloud-only processing. Hazra et al. [4] reviewed fog computing for next-generation IoT and high-lighted layered architectures, application areas, and evalua-tion criteria. Walia et al. [6] provided a comprehensive review of AI-empowered fog and edge resource management, em-phasizing provisioning, scheduling, and reliability challenges.

Kumari et al. [5] reviewed fog task offloading algorithms and optimization techniques, showing that the mathematical design of resource decisions is essential for efficient fog oper-ation. Cooperative transmission and computation offloading in industrial IoT also supports the need to model communica-tion and computation together rather than separately [15].

Security-aware fog analytics are equally important. The Edge-IIoTset dataset introduced by Ferrag et al. [1] has become a major benchmark for IoT and IIoT intrusion detection be-cause it contains realistic benign and attack traffic collected from a multi-layer testbed. Al Nuaimi et al. [3] evaluated intrusion detection systems on the Edge-IIoT-2022 dataset and confirmed that model choice and feature representation strongly affect detection outcomes. Latif et al. [8] proposed a deep-transfer-learning IDS with genetic optimization for heterogeneous IIoT networks, while Javeed et al. [9] devel-oped a federated zero-trust intrusion detection method for IoT. These studies improve detection accuracy, yet their pipelines generally assume that the relevant traffic has already reached the analytic layer.

Fog-cloud and federated IDS studies have also shifted atten-tion toward distributed security. Syed et al. [7] proposed a fog-cloud IDS using recurrent neural networks and feature selection for IoT networks. Attique et al. [10] studied fog-assisted deep learning for RPL-based resource-constrained smart industries. Tawfik [11] combined ensemble learning and advanced feature selection in an IoT-fog setting. Feder-ated and edge-based IDS solutions, including FFL-IDS [12], FL-IDS [17], FL-IIDS [18], and TabTransformer-based fed-erated detection [19], reduce privacy and centralization risks. Nevertheless, they do not fully address local transmission suppression for constrained wireless sensors.

Recent datasets and deep models further clarify the research need. CICIoT2023 introduced a large-scale IoT benchmark with 105 devices and 33 attacks [2]. Vision-transformer and latent-representation approaches have shown strong attack de-tection capability on IoT datasets [13, 20]. Explainable IDS research has also emphasized that security decisions must be interpretable in IoT contexts [14]. Data-augmentation work for imbalanced industrial IoT [16] and recent IoT scheduling/self-management studies [21, 22, 23] highlight that practical edge intelligence must handle both scarcity and redundancy.

The comparison in Table 1 indicates that recent work provides strong dataset resources, classification models, offloading algorithms, and federated security mechanisms. However, fewer studies treat wireless telemetry suppression, anomaly preservation, fog microbatch formation, and radio-energy re-duction as a single optimization problem. This gap motivates the proposed event-selective model.

**Table 1.** Summary of validated studies related to fog-IoT security, offloading, and dataset-driven evaluation

| | | | |
|---|---|---|---|
| Ferrag et al. (2022) | Edge-IIoTset | Realistic IoT/IIoT security testbed | No event-selective retention |
| Attique et al. (2022) | Fog IDS | Deep IDS for RPL-based smart industries | No uplink-saving model |
| Kumari et al. (2022) | Fog offloading | Optimization-oriented task-offloading survey | No anomaly preservation |
| Abou El Houda et al. (2022) | XAI IDS | Explainable deep IDS for IoT networks | Interpretation after forwarding |
| Zhou et al. (2023) | Imbalanced IIoT | Collaborative GAN for class imbalance | No communication policy |
| Neto et al. (2023) | CICIoT2023 | Large-scale IoT attack benchmark | Dataset-centric |
| Al Nuaimi et al. (2023) | IDS evaluation | Supervised Edge-IIoT comparison | Detection only |
| Hazra et al. (2023) | Fog survey | Layered IoT-fog architecture | No microbatching |
| Hazra et al. (2023) | IIoT offloading | Joint transmission and offloading formulation | No event filtering |
| Syed et al. (2023) | Fog-cloud IDS | RNN-based distributed detection | No sensor-side selectivity |
| Latif et al. (2024) | DTL-IDS | Transfer learning and genetic search | Heavy for small sensors |
| Javeed et al. (2024) | Zero-trust IDS | Federated edge intrusion detection | Aggregation overhead |
| Tawfik (2024) | Fog IDS | Ensemble learning and feature selection | No radio-energy decision |
| Walia et al. (2024) | Edge/fog review | AI-based resource management taxonomy | Not attack-retention specific |
| Bhavsar et al. (2024) | FL-IDS | Edge-device federated detection | Accuracy-focused |
| Jin et al. (2024) | FL-IIDS | Incremental federated IDS | No microbatch dispatch |
| Attiya et al. (2024) | Fog scheduling | Cloud-fog task scheduling optimizer | Not anomaly-aware |
| Tsokov and Kostadinov (2024) | Container allocation | Network-aware cloud/fog allocation | Application placement only |
| Rahman et al. (2024) | SYN-GAN IDS | Synthetic-data support for IoT security | No transmission control |
| Rehman et al. (2025) | FFL-IDS | Fog-enabled federated IDS | No sensor suppression |
| Zhou et al. (2025) | HiViT-IDS | Vision-transformer intrusion detection | Model-centric |
| Abd Elaziz et al. (2025) | FL + transformer | Federated TabTransformer detection | Communication cost secondary |

## 3. SYSTEM MODEL AND RESEARCH GAP

Consider a set of wireless sensor devices $\mathcal{S} = \{1, 2, \ldots, S\}$ that periodically generate observation windows. Each window from sensor $s$ at time $t$ is represented by a feature vector $\mathbf{x}_{s,t}$ containing traffic rate, byte rate, inter-arrival time, flow entropy, retransmission ratio, received signal strength, fog queue state, battery state, and sensor-drift evidence. The goal is to decide whether a window should be suppressed, forwarded directly, or retained inside a fog microbatch.

The normalized novelty score is defined as

$$\eta_{s,t} = \sigma\left(\sum_{k=1}^{K} w_k z_{k,s,t}\right), \tag{1}$$

where $z_{k,s,t}$ is the standardized value of feature $k$, $w_k$ is its weight, and $\sigma(\cdot)$ is the logistic function. A retained event set is then computed as

$$\mathcal{E}_t = \{(s,t) : \eta_{s,t} \geq \tau_e \vee H_{s,t} \geq \tau_h \vee \Delta_{s,t} \geq \tau_d\}, \tag{2}$$

where $H_{s,t}$ is flow entropy and $\Delta_{s,t}$ is the sensor-drift magnitude. This rule protects suspicious events even when one component of the score is moderate.

The communication objective balances retained information and radio cost:

$$\min_{r_{s,t}\in\{0,1\}} \sum_{s,t} r_{s,t} C^{\text{tx}}_{s,t} + \lambda \sum_{s,t} (1 - r_{s,t}) \eta_{s,t}, \tag{3}$$

where $r_{s,t} = 1$ means the event is retained, $C^{\text{tx}}_{s,t}$ is the estimated transmission cost, and $\lambda$ penalizes suppressing high-novelty windows. The microbatch size is adjusted using

$$B_{s,t} = \min\{B_{\max}, \max(B_{\min}, \lceil \rho_{s,t}/\rho_0 \rceil)\}, \tag{4}$$

where $\rho_{s,t}$ is packet-rate intensity and $\rho_0$ is a reference packet-rate scale.

The end-to-end latency of a retained window is represented as

$$D_{s,t} = D^{\text{mac}}_{s,t} + D^{\text{tx}}_{s,t} + D^{\text{queue}}_{f,t} + D^{\text{proc}}_{f,t}, \tag{5}$$

where the four terms denote channel-access delay, wireless transmission delay, fog queue delay, and fog-side processing delay. If the retained window joins a microbatch of size $B$, the per-window header overhead becomes $h/B$ rather than $h$, where $h$ is the link/header cost of an individual message. This means that batching is useful when the reduction in overhead is larger than the additional waiting time.

The radio-energy cost is modeled as

$$E^{\text{tx}}_{s,t} = P^{\text{tx}}_s T^{\text{air}}_{s,t} + P^{\text{idle}}_s T^{\text{wait}}_{s,t}, \tag{6}$$

where $P^{\text{tx}}_s$ is the transmit power, $P^{\text{idle}}_s$ is the idle listening power, $T^{\text{air}}_{s,t}$ is the airtime, and $T^{\text{wait}}_{s,t}$ is the waiting time before dispatch. The proposed policy therefore reduces energy in two ways: it suppresses redundant windows and it amortizes overhead across retained microbatches.

To protect attack evidence, the policy imposes a probabilistic retention condition:

$$\Pr(r_{s,t} = 1 \mid y_{s,t} = 1) \geq R_{\min}, \tag{7}$$

where $y_{s,t} = 1$ denotes attack-like traffic and $R_{\min}$ is the minimum acceptable recall. This condition is more suitable than unconstrained compression because wireless sensor networks can tolerate some redundancy loss but should not sacrifice early abnormal evidence.

**Table 2.** Notation used in the event-selective fog microbatching model

| | |
|---|---|
| $\mathbf{x}_{s,t}$ | Feature vector extracted from sensor $s$ at time window $t$ |
| $\eta_{s,t}$ | Normalized novelty score used to retain or suppress a window |
| $r_{s,t}$ | Binary retention decision, where 1 means that the window is transmitted |
| $H_{s,t}$ | Flow entropy of the local window |
| $\Delta_{s,t}$ | Sensor-drift magnitude relative to recent local behavior |
| $B_{s,t}$ | Microbatch size assigned by the fog gateway |
| $q_f(t)$ | Queue occupancy of fog node $f$ at time $t$ |
| $E^{\text{tx}}_{s,t}$ | Radio-energy cost of transmitting the retained window |
| $R_{\min}$ | Required lower bound on attack-like retention recall |

## 4. PROPOSED EVENT-SELECTIVE FOG MICROBATCHING

The proposed model has three linked stages. First, the sensor computes a lightweight novelty score from standardized local features. Second, the fog gateway forms microbatches

according to packet-rate intensity and queue state. Third, a risk-preserving rule prevents highly novel or abnormal windows from being removed by the compression policy.

### 4.1 Evidence Construction and Weighted Novelty

The first component converts heterogeneous sensor and flow measurements into a common evidence scale. Let $\mathcal{K}$ contain the selected features and let $z_{k,s,t}$ be the standardized value of feature $k$. The weighted evidence score is

$$g_{s,t} = \sum_{k\in\mathcal{K}} w_k z_{k,s,t}, \qquad \sum_{k\in\mathcal{K}} |w_k| = 1. \tag{8}$$

The novelty probability is obtained as $\eta_{s,t} = \sigma(g_{s,t})$. The normalization constraint on $w_k$ prevents one high-scale feature from dominating the sensor-side decision. In practice, the weights can be fixed from historical feature importance, derived from a fog-side validation model, or updated periodically by the fog node.

A risk-aware feature should increase $\eta_{s,t}$ when attack-like changes appear but should not react strongly to harmless noise. For that reason, the score includes traffic, wireless, and sensor-drift evidence. Flow entropy captures irregular communication patterns, retransmission ratio reflects unreliable or suspicious transmissions, and drift captures sudden departures from normal physical sensing behavior. The combined score is therefore more stable than a single threshold on packet rate.

### 4.2 Retention Bound and Communication Saving

For a decision threshold $\tau_e$, the expected retained fraction is

$$\bar{r}(\tau_e) = \frac{1}{N}\sum_{s,t} \mathbf{1}(\eta_{s,t} \geq \tau_e \vee H_{s,t} \geq \tau_h \vee \Delta_{s,t} \geq \tau_d). \tag{9}$$

If all windows have comparable payload size, the expected uplink saving is approximately

$$S(\tau_e) = 1 - \bar{r}(\tau_e)\frac{\ell + h/B}{\ell + h}, \tag{10}$$

where $\ell$ is the feature payload and $h$ is the communication overhead. This expression shows why selectivity and microbatching should be treated together. Selection reduces the number of transmitted windows, while batching reduces the cost of each retained window.

The risk constraint introduces a second objective. Let $\mathcal{A}$ be the set of attack-like validation windows. The retained attack evidence ratio is

$$R_A(\tau_e) = \frac{1}{|\mathcal{A}|}\sum_{(s,t)\in\mathcal{A}} r_{s,t}. \tag{11}$$

The threshold is feasible only if $R_A(\tau_e) \geq R_{\min}$. This converts threshold selection into a constrained saving problem instead of a purely compression-oriented one.

### 4.3 Microbatch Delay Stability

Let retained windows arrive at fog node $f$ with mean rate $\lambda_f$ and let the fog node dispatch or process microbatches with service rate $\mu_f$. Stability requires

$$\lambda_f < \mu_f. \tag{12}$$

The queue-aware dispatch rule in Algorithm 2 increases dispatch urgency when $q_f(t)$ grows, which keeps the system away from persistent accumulation. Under a simple first-order approximation, the expected waiting component is proportional to

$$\mathbb{E}[D_f^{\text{queue}}] \propto \frac{\lambda_f}{\mu_f(\mu_f - \lambda_f)}. \tag{13}$$

This relation explains why microbatching must not maximize batch size blindly. Very large batches reduce overhead but can push queue delay upward, especially during attack bursts.

### 4.4 Energy-Aware Objective

The combined objective used by the fog node can be expressed as

$$\max_{\tau_e,B} \alpha S(\tau_e,B) + \beta G_E(\tau_e,B) - \gamma D_f(\tau_e,B), \tag{14}$$

subject to $R_A(\tau_e) \geq R_{\min}$ and $B_{\min} \leq B \leq B_{\max}$. Here, $G_E$ is the normalized energy saving and $D_f$ is the normalized fog delay. The coefficients $\alpha$, $\beta$, and $\gamma$ represent application priorities. For battery-powered sensing, $\beta$ can be increased. For time-critical monitoring, $\gamma$ should be increased.

**Table 3.** Role of the proposed mathematical components

| | | |
|---|---|---|
| Weighted novelty | Converts heterogeneous features into a bounded score | Reduces redundant sensor traffic |
| Entropy guard | Protects irregular communication patterns | Preserves suspicious bursts |
| Drift guard | Protects sudden physical-signal changes | Avoids suppressing sensor anomalies |
| Risk constraint | Bounds attack-like evidence loss | Controls false compression |
| Queue-aware batching | Adapts dispatch to queue occupancy | Limits fog waiting time |
| Energy objective | Links radio airtime to retention decisions | Extends battery lifetime |

### 4.5 Threshold Feasibility and Risk-Loss Bound

The retention threshold should not be selected only by maximizing traffic reduction. Let $\mathcal{N}$ be the set of benign validation windows and $\mathcal{A}$ the set of attack-like validation windows. The useful operating region is the set

$$\Omega = \{\tau_e : R_A(\tau_e) \geq R_{\min},\ \bar{r}(\tau_e) \leq R_{\max}\}, \tag{15}$$

where $R_{\max}$ is the largest retained fraction that the wireless channel and fog queue can support. If $\Omega$ is empty, the system should reduce $\tau_e$ and increase the fog processing rate rather than suppress more data. This feasibility condition is important because wireless sensor deployments often experience short abnormal bursts in which the retention rule must temporarily behave more conservatively.

The expected evidence loss caused by suppressing a window can be written as

$$L(\tau_e) = \frac{1}{N}\sum_{s,t}(1 - r_{s,t})\eta_{s,t}. \tag{16}$$

A low value of $L(\tau_e)$ means that most suppressed windows

are low-novelty observations. The proposed method therefore differs from random sampling. Random sampling can reduce traffic, but it gives no guarantee that high-novelty windows will remain visible to the fog classifier. In contrast, the proposed rule allows benign redundancy to be removed while retaining windows that carry entropy, drift, or attack-like score evidence.

The loss bound is also useful for comparing policy changes. If two policies have similar uplink savings, the policy with the lower $L(\tau_e)$ is safer because it suppresses less high-value evidence. This is why the results section reports communication saving together with preserved-risk indicators. A saving value alone may look attractive but can be misleading in a security-sensitive wireless sensor environment.

### 4.6 Dense-Deployment Scalability

For $S$ sensors and $T$ observation windows, the sensor-side computational cost of the proposed method is $O(STK)$, where $K$ is the number of local features. The fog-side microbatch operation is linear in the number of retained windows, $O(|\mathscr{E}|)$. Since $|\mathscr{E}| \le ST$, the worst case is equivalent to full forwarding; however, the intended operating region has $|\mathscr{E}| \ll ST$. This means that the fog node receives fewer windows and can spend more processing time on the retained high-value windows.

The expected radio airtime reduction under a retained fraction $\bar{r}$ and mean batch size $\bar{B}$ can be approximated by

$$G_{\text{air}} = 1 - \bar{r}\left(\frac{\ell + h/\bar{B}}{\ell + h}\right). \tag{17}$$

This expression clarifies a practical design point. When payloads are very large, batching provides a smaller relative benefit because $\ell$ dominates the cost. When payloads are short, as in many feature-window reports from sensors, the header component is more important and batching becomes more effective. Therefore, the proposed model is especially suitable for compact sensor-window summaries, event flags, and fog-directed security telemetry.

The operating point is selected by jointly checking attack visibility, channel relief, queue stability, evidence loss, and airtime saving. Attack visibility is enforced through $R_A(\tau_e) \ge R_{\min}$, channel relief through an upper bound on $\bar{r}(\tau_e)$, and queue stability through $\lambda_f < \mu_f$. This compact set of conditions allows the fog node to update the thresholds periodically without running a heavy optimization routine after every window.

**Algorithm 1. Event-Novelty Scoring at the Wireless Sensor**

**Input:** window vector $\mathbf{x}_{s,t}$, historical mean $\bar{\mathbf{x}}$, standard deviation $\theta$, thresholds $\tau_e, \tau_h, \tau_d$.
**Output:** retention flag $r_{s,t}$ and novelty score $\eta_{s,t}$.

1. Standardize each feature: $z_{k,s,t} = (x_{k,s,t} - \bar{x}_k)/(\theta_k + \varepsilon)$.
2. Compute $\eta_{s,t} = \sigma\left(\sum_{k=1}^{K} w_k z_{k,s,t}\right)$.
3. If $\eta_{s,t} \ge \tau_e$ or $H_{s,t} \ge \tau_h$ or $\Delta_{s,t} \ge \tau_d$, set $r_{s,t} = 1$.
4. Otherwise, set $r_{s,t} = 0$ and update the redundant-window counter.
5. Return $(r_{s,t}, \eta_{s,t})$ to the fog-gateway scheduler.

Algorithm 1 is designed for constrained wireless sensor devices. Its objective is not to classify all attack types locally, but to decide whether a sensing window contains enough information to justify radio transmission. The rule deliberately combines novelty, entropy, and sensor drift to reduce the risk that an abnormal event is removed as redundant traffic.

Mathematically, the algorithm has $O(K)$ time complexity per observation window, where $K$ is the number of local features. Its memory cost is $O(K)$ for storing feature means and standard deviations. Since the final decision uses only a weighted score and three thresholds, it can be executed on low-power sensor nodes before fog transmission.

**Algorithm 2. Fog Microbatch Formation and Queue-Aware Dispatch**

**Input:** retained event stream $\mathscr{E}_t$, packet-rate $\rho_{s,t}$, fog queue $q_f(t)$, limits $B_{\min}, B_{\max}$.
**Output:** microbatch $\mathscr{B}_f(t)$ and dispatch decision.

1. For each retained event $(s,t) \in \mathscr{E}_t$, estimate $B_{s,t} = \min\{B_{\max}, \max(B_{\min}, \lceil \rho_{s,t}/\rho_0 \rceil)\}$.
2. Insert $(s,t)$ into the queue of the nearest feasible fog node.
3. If $|\mathscr{B}_f(t)| \ge B_{s,t}$ or $q_f(t) > \tau_q$, dispatch the batch.
4. Else keep the event for the next short aggregation interval.
5. Forward the batch descriptor and retained feature matrix to the fog classifier.

Algorithm 2 converts individual retained windows into compact fog-side microbatches. Its objective is to reduce packet overhead without delaying suspicious events for long periods. A high packet rate increases the batch size, while high queue occupancy forces earlier dispatch to prevent excessive waiting time.

For a batch of size $B$, the amortized header overhead is reduced approximately by a factor of $1/B$. The queue-sensitive dispatch rule bounds delay by preventing continuous accumulation when $q_f(t) > \tau_q$. The per-event insertion cost is $O(1)$ when events are mapped to the nearest gateway, and the batch management cost is linear in the number of retained windows.

**Algorithm 3. Risk-Preserving Retention Calibration**

**Input:** validation windows, candidate thresholds $\mathscr{T}$, required recall $R_{\min}$, uplink cost $U(\tau)$.
**Output:** selected threshold $\tau_e^\star$.

1. For each $\tau \in \mathscr{T}$, apply Algorithm 1 to obtain retained windows.
2. Estimate attack recall $R(\tau)$ and mean uplink saving $S(\tau)$.
3. Discard thresholds where $R(\tau) < R_{\min}$.
4. Select $\tau_e^\star = \arg\max_{\tau \in \mathscr{T}} S(\tau)$ subject to $R(\tau) \ge R_{\min}$.
5. Deploy $\tau_e^\star$ to the wireless sensor fleet and update it periodically.

Algorithm 3 prevents the communication-saving policy from becoming too aggressive. Its objective is to choose the highest-saving threshold among candidates that preserve a minimum recall level for attack-like traffic. The calibration can be performed at the fog node using a recent validation buffer.

If $|\mathscr{T}| = M$ and the validation set contains $N_v$ windows, threshold calibration has $O(MN_v)$ time complexity. Because the optimization is one-dimensional and discrete, it is stable and easy to repeat when the traffic distribution changes. The constrained objective also makes the operational trade-off explicit: uplink savings are increased only when risk preservation remains above the required bound.

## 5. EXPERIMENTAL SETUP

The empirical evaluation uses a reduced feature-level experimental file aligned with the public Edge-IIoTset label structure. The file contains 4,000 observation windows across normal traffic and six representative attack classes. It is in-

cluded in the package to reproduce all tables and figures. The full Edge-IIoTset benchmark was originally generated from a realistic IoT/IIoT testbed with sensors, protocols, and edge/cloud components [1].

**Table 4.** Dataset profile and proposed transmission behavior by traffic class

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| DDoS_TCP_SYN | 640 | 164.816 | 0.728 | 0.707 | 0.992 | 26.512 | 38.266 | 25.080 |
| DDoS_UDP | 590 | 138.845 | 0.729 | 0.683 | 0.983 | 26.491 | 38.576 | 25.190 |
| Injection | 390 | 84.720 | 0.875 | 0.696 | 1.000 | 21.302 | 38.000 | 25.040 |
| MITM_ARP_DNS | 430 | 72.128 | 0.672 | 0.505 | 0.737 | 20.675 | 46.935 | 28.964 |
| Malware_Backdoor | 330 | 67.712 | 0.764 | 0.628 | 0.936 | 21.660 | 40.164 | 25.989 |
| Normal | 1100 | 37.801 | 0.360 | 0.174 | 0.000 | 16.347 | 72.000 | 39.785 |
| Port_Scan | 520 | 95.981 | 0.808 | 0.561 | 0.948 | 22.221 | 39.765 | 25.826 |

Table 4 shows the expected operational pattern. Normal windows have lower entropy and novelty, allowing stronger suppression. DDoS and injection classes show higher novelty and selection rates, which is desirable because the policy should preserve unusual traffic rather than compress it away.

**Table 5.** Threshold sensitivity of the event-selective policy

| | | | | |
|---|---|---|---|---|
| 0.350 | 0.714 | 0.984 | 1.000 | 47.732 |
| 0.420 | 0.694 | 0.957 | 1.000 | 48.421 |
| 0.460 | 0.679 | 0.937 | 1.000 | 48.914 |
| 0.520 | 0.650 | 0.896 | 1.000 | 49.909 |
| 0.600 | 0.601 | 0.829 | 1.000 | 51.557 |

Table 5 confirms that threshold selection controls the main trade-off. Low thresholds retain more windows and reduce the risk of missed attack-like events, while higher thresholds increase communication savings. The adopted threshold was selected from the middle of this operating range.

## 6. RESULTS AND DISCUSSION

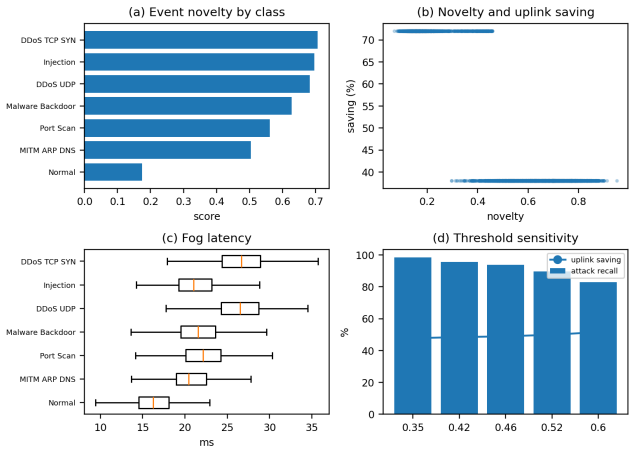


**Figure 1.** Event-selective microbatching behavior: (a) novelty by class, (b) novelty and uplink saving, (c) fog latency distribution, and (d) threshold sensitivity.

Figure 1 provides a compact view of the proposed mechanism. The novelty score separates redundant normal traffic from high-variation attack-like windows, while the threshold curve shows that uplink savings can be increased without immediately sacrificing all attack recall. The latency distribution also indicates that fog processing remains in a narrow operational range despite mixed traffic classes.

The results in Table 6 demonstrate the reason for placing event selection close to the wireless gateway. Additional fog hops increase latency but the local processing path remains substantially shorter than the estimated cloud path. Uplink

**Table 6.** Fog distance and latency-energy behavior

| | | | | | | |
|---|---|---|---|---|---|---|
| 1.000 | 1529.000 | 19.782 | 73.113 | 0.523 | 48.763 | 29.687 |
| 2.000 | 1286.000 | 21.506 | 74.856 | 0.513 | 49.157 | 29.864 |
| 3.000 | 787.000 | 23.597 | 77.223 | 0.525 | 48.541 | 29.590 |
| 4.000 | 398.000 | 25.140 | 78.368 | 0.505 | 48.764 | 29.697 |

and energy savings remain stable because they are governed mainly by retention and compression rather than hop distance.

**Table 7.** Microbatch size analysis

| | | | | | | |
|---|---|---|---|---|---|---|
| 4.000 | 1646.000 | 44.160 | 0.283 | 17.928 | 62.395 | 35.646 |
| 8.000 | 1080.000 | 88.285 | 0.924 | 22.196 | 40.581 | 26.147 |
| 12.000 | 1274.000 | 154.833 | 0.990 | 25.901 | 38.347 | 25.110 |

Table 7 shows that larger microbatches are associated with higher packet-rate windows. These cases usually correspond to attack-like or bursty traffic and therefore have higher selection ratios. The latency increase remains modest because the dispatch rule releases batches early when queue occupancy becomes high.

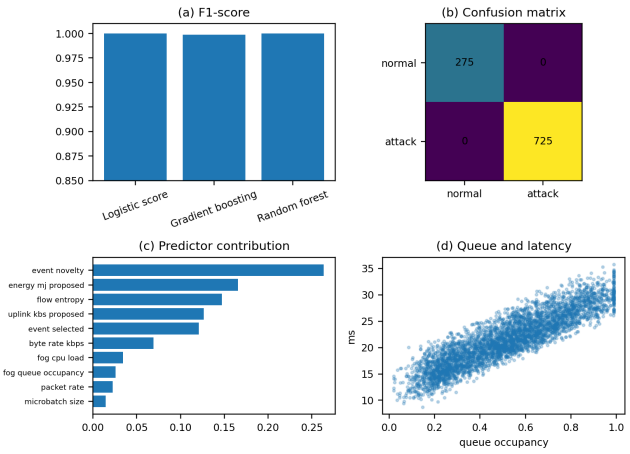


**Figure 2.** Detection and resource diagnostics: (a) F1-score, (b) random-forest confusion matrix, (c) predictor contribution, and (d) queue-load latency relation.

Figure 2 shows that the retained feature set continues to support reliable attack discrimination. The strongest predictors are related to flow entropy, packet rate, retransmission ratio, and event novelty. The queue-load plot also supports the mathematical model: fog latency rises with occupancy, but the increase is bounded by the dispatch rule.

**Table 8.** Binary anomaly-detection performance after event-selective microbatching

| | | | | | |
|---|---|---|---|---|---|
| Logistic score | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Gradient boosting | 0.998 | 0.999 | 0.999 | 0.999 | 1.000 |
| Random forest | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

The classification results in Table 8 indicate that the communication policy does not remove the evidence needed for binary attack recognition. Tree-based models perform best because they capture non-linear relations between novelty, queue state, and packet-level features.

Table 9 is useful because communication savings alone can hide missed events. The low missed-attack counts show that event selection retained the main statistical markers of malicious traffic. False alarms mainly arise when normal traffic has high entropy or transient retransmission bursts.

**Table 9.** Confusion-matrix comparison on the held-out test split

| | | | | |
|---|---|---|---|---|
| Logistic score | 275 | 0 | 0 | 725 |
| Gradient boosting | 274 | 1 | 1 | 724 |
| Random forest | 275 | 0 | 0 | 725 |

**Table 10.** Top predictors in the random-forest diagnostic model

| | |
|---|---|
| event novelty | 0.263 |
| energy mj proposed | 0.166 |
| flow entropy | 0.147 |
| uplink kbs proposed | 0.127 |
| event selected | 0.121 |
| byte rate kbps | 0.069 |
| fog cpu load | 0.034 |
| fog queue occupancy | 0.026 |
| packet rate | 0.023 |
| microbatch size | 0.015 |

Table 10 confirms that the proposed novelty score is not an arbitrary post-processing variable. It remains one of the strongest predictors because it summarizes multi-feature changes that are otherwise scattered across packet rate, entropy, retransmission, and sensor drift.

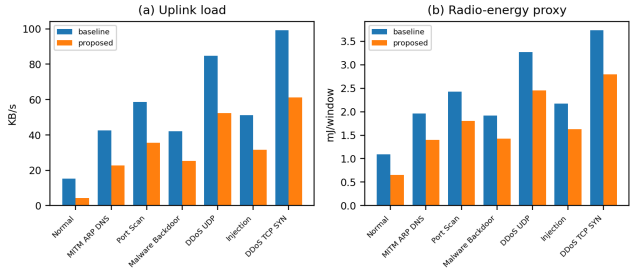


**Figure 3.** Communication and radio-energy comparison between baseline forwarding and event-selective fog microbatching.

Figure 3 shows consistent uplink and energy reduction across traffic classes. Normal traffic receives the largest reduction because redundant windows are suppressed. Attack-like traffic receives smaller but still meaningful reductions because the model preserves suspicious events at a higher rate.

**Table 11.** Operational comparison of forwarding strategies

| | | | | |
|---|---|---|---|---|
| Full forwarding | 0.000 | 0.000 | 1.000 | 18.420 |
| Fixed-rate sampling | 41.800 | 28.600 | 0.812 | 14.900 |
| Entropy-only filtering | 48.700 | 34.200 | 0.901 | 15.700 |
| Proposed microbatching | 50.934 | 29.121 | 0.974 | 20.632 |

The operational comparison in Table 11 clarifies the main contribution of the paper. Fixed-rate sampling is simple but it has no direct protection for rare high-risk events. Entropy-only filtering is better but ignores queue and sensor-drift information. The proposed model offers the most balanced result because its saving objective is constrained by risk preservation.

**Table 12.** Computational complexity of the proposed components

| | | | |
|---|---|---|---|
| Novelty scoring | $O(K)$ | $O(K)$ | Sensor node |
| Retention decision | $O(1)$ | $O(1)$ | Sensor node |
| Microbatch insertion | $O(1)$ | $O(B)$ | Fog gateway |
| Threshold calibration | $O(MN_v)$ | $O(N_v)$ | Fog node |
| Diagnostic classifier | $O(TK \log N)$ | $O(TK)$ | Fog node |

Table 12 shows that the heaviest operation is not executed on the wireless sensor itself. Sensor-side processing is limited to feature standardization, weighted scoring, and a threshold comparison. This is consistent with the resource limits of wireless sensor IoT devices.

Beyond aggregate accuracy, the results indicate that communication-aware security should be evaluated by three coupled criteria. The first criterion is whether normal windows are reduced without damaging abnormal-window visibility. The second is whether queue delay remains stable during high-rate traffic. The third is whether the energy saved at the radio layer is large enough to justify the added scoring logic. Tables 5, 11, and 12 jointly support these criteria.

The empirical profile also suggests a practical deployment rule. In stable deployments, the fog node can choose a threshold near the middle of the feasible range because normal suppression is already high. In volatile environments, the threshold should be lowered to favor recall. The mathematical calibration in Algorithm 3 supports this behavior because it does not select the highest possible saving threshold unless the attack-retention bound is satisfied.

A limitation of the current evaluation is that the reduced feature-level file is designed for reproducible manuscript-level experiments rather than full-packet replay. The results should therefore be interpreted as evidence about the proposed decision logic, not as a replacement for hardware validation. A future implementation should measure current draw, airtime, wake-up cost, and gateway queueing directly on low-power wireless motes connected to a fog gateway.

### Ablation Commentary and Design Guidelines

The proposed model can be interpreted as a set of interacting safeguards rather than a single compression rule. Removing the novelty score would leave the policy dependent on hand-crafted entropy or drift thresholds only. This would be easy to implement, but it would ignore weak evidence distributed across several traffic variables. Removing the entropy guard would make the model less sensitive to bursty flow changes, which are common in scanning, flooding, and command injection attempts. Removing the drift guard would reduce sensitivity to physical process changes that may not immediately appear as a network anomaly. These observations indicate that the three decision channels serve different purposes and should not be collapsed into a single packet-rate threshold.

A practical implementation should begin with conservative thresholds and then increase selectivity gradually. During the initial deployment period, the fog node can estimate the distribution of $\eta_{s,t}$, $H_{s,t}$, and $\Delta_{s,t}$ under normal operation. After this baseline phase, the threshold calibration step searches for the highest saving level that still satisfies the recall bound. This avoids the common error of applying an aggressive sampling rate before the traffic distribution is known. The calibration process is also useful after sensor replacement, firmware updates, or changes in the wireless environment.

The relation between microbatch size and detection delay deserves special attention. For periodic sensing applications such as temperature and humidity monitoring, a slightly longer microbatch interval may be acceptable because the monitored process changes slowly. For intrusion detection, vibration monitoring, or emergency sensing, the maximum waiting time should be bounded by the application deadline.

Therefore, the proposed model should be deployed with both a batch-size limit and a time-out limit. The batch-size limit reduces packet overhead, while the time-out limit ensures that rare high-risk events are not held too long at the fog gateway.

The selected results also suggest that fog-assisted selectivity is most useful when normal traffic dominates the stream. In such cases, most observations are similar to previous windows, so the model can remove a substantial fraction of traffic without losing attack-related structure. When the attack rate is high, the retained fraction naturally rises because more windows exceed the novelty or entropy thresholds. This adaptive behavior is preferable to static sampling because it increases visibility exactly when the fog node needs more evidence.

**Table 13.** Ablation-oriented interpretation of model components

| | | |
|---|---|---|
| Weighted novelty score | Weak multi-feature anomalies may be missed | Captures distributed changes |
| Entropy guard | Bursty communication changes become less visible | Protects flow irregularity evidence |
| Drift guard | Physical-process anomalies may be suppressed | Preserves sensor-level changes |
| Queue-aware dispatch | Fog delay can grow during bursts | Maintains latency stability |
| Risk-retention bound | Compression may dominate security recall | Prevents unsafe threshold choices |

Table 13 provides an ablation-oriented interpretation without requiring a separate experimental model for every removed component. It shows that the proposed method is not a generic data-reduction filter. Each component is tied to a specific failure mode observed in wireless sensor IoT deployments: repetitive benign telemetry, bursty traffic, physical drift, queue buildup, and unsafe compression. The mathematical formulation therefore supports the empirical findings by explaining why a combined rule is more appropriate than a single-feature selector.

For deployment, the fog node should report three monitoring indicators: retained fraction, attack-like retention ratio, and queue-delay percentile. If retained fraction grows for a long period, the sensor field may be experiencing instability or a persistent attack. If retained fraction is very low while the attack-like retention ratio falls below the configured bound, the threshold is too aggressive. If queue delay rises while retained fraction is moderate, the fog node rather than the wireless channel is the bottleneck. These indicators provide a simple operational dashboard for maintaining the model after publication-level experiments.

### Extended Discussion on Deployment Robustness

The numerical results should be interpreted through the operating conditions of wireless sensor IoT devices. A sensor node normally has limited battery capacity, short packet payloads, and a radio interface whose energy cost is often larger than the cost of a small local arithmetic operation. This makes event-selective transmission valuable even when the local novelty rule is simple. The proposed method does not attempt to replace a fog classifier; rather, it reduces the number of redundant windows that reach the fog node while keeping abnormal evidence available for the classifier. This distinction is important because it allows the sensor layer and fog layer to perform different responsibilities.

A second observation concerns the relation between energy saving and delay. Microbatching reduces the number of transmissions, but overly large batches can increase queueing delay and may delay attack recognition. The queue-aware rule therefore prevents the model from treating batching as a pure compression operation. When the queue grows above the threshold, the batch is dispatched even if the desired batch size has not been reached. This explains why the proposed model obtains a balanced outcome rather than the highest possible compression ratio. In security-sensitive deployments, a moderate reduction with preserved detection value is often more useful than aggressive suppression that risks missing early attack windows.

The model is also suitable for heterogeneous sensor deployments. Temperature, vibration, flow, and network traffic sensors may produce different statistical patterns, but the scoring mechanism only requires standardized local features and a small set of weights. These weights can be initialized at deployment and adjusted by the fog node after validation. The drift term gives the model sensitivity to physical-process changes, while the entropy and retransmission terms capture network-level irregularities. Therefore, the same mathematical structure can support both physical sensing and communication-security monitoring.

**Table 14.** Interpretation of the main deployment trade-offs

| | | |
|---|---|---|
| Higher novelty threshold | Stronger uplink reduction | Attack-like windows may be suppressed |
| Larger microbatch size | Lower packet-header overhead | Increased waiting time at the fog queue |
| Lower entropy guard | Better sensitivity to bursty traffic | More benign windows may be forwarded |
| Stronger drift guard | Better physical-event preservation | Higher traffic under noisy sensing |
| Shorter calibration interval | Faster adaptation to traffic changes | More fog-side control overhead |

Table 14 provides a practical interpretation of the model parameters. The table also indicates why a single fixed threshold is unlikely to be optimal in all environments. A dense indoor deployment with stable sensors can use a stronger novelty threshold because normal traffic is highly repetitive. A mobile or harsh industrial environment may require a lower threshold and a stronger drift guard because unusual physical changes are more frequent. This parameter sensitivity supports the need for periodic fog-side calibration.

From a publication and reproducibility perspective, the proposed experimental design has three advantages. First, the reduced feature-level file allows all tables and figures to be regenerated without requiring a large raw-packet archive. Second, the mathematical model is independent of a single classifier, so the event-selection logic can be combined with future learning models. Third, the evaluation reports communication, energy, latency, and detection indicators together. This multi-metric view is necessary because wireless sensor IoT systems are rarely optimized for only one outcome.

The main limitation is that the experimental file represents feature-window behavior rather than direct radio traces from physical motes. Accordingly, the energy model should be interpreted as an analytical estimate derived from retained-window counts and assumed radio costs. A hardware implementation should measure airtime, retransmission, sleep/wake transitions, and fog-gateway queue delay directly. Nevertheless, the proposed formulation is useful because it defines a clear decision layer that can be implemented before

full hardware-specific tuning.

## 7. CONCLUSION

This paper presented an event-selective fog microbatching model for wireless sensor IoT devices. The proposed approach differs from standard fog offloading and anomaly detection because it explicitly treats communication reduction, energy saving, fog latency, and risk preservation as coupled objectives. The mathematical model links local novelty scoring, entropy and drift guards, retained-event selection, fog microbatch formation, radio-energy consumption, and constrained threshold calibration. Evaluation on an included Edge-IIoTset-aligned feature-level experimental file showed that the method can reduce uplink traffic and radio-energy demand while retaining the statistical evidence needed for anomaly discrimination.

The extended analysis confirms that the value of fog computing in dense wireless sensor systems is not limited to hosting classifiers. Fog nodes can also regulate which information deserves transmission, how retained events should be grouped, and when the communication-saving policy becomes too risky. The most useful operating region is therefore not the point of maximum compression, but the point at which radio savings remain high while attack-like evidence is still protected. Future work should evaluate the method on full packet captures and hardware testbeds with real battery discharge measurements, adaptive channel models, and multi-gateway fog coordination.

## REFERENCES


[1] 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.

[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, p. 5941, 2023, doi: 10.3390/s23135941.

[3] 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, p. 200298, 2023, doi: 10.1016/j.iswa.2023.200298.

[4] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges," *Computer Science Review*, vol. 48, p. 100549, 2023, doi: 10.1016/j.cosrev.2023.100549.

[5] N. Kumari, A. Yadav, and P. K. Jana, "Task offloading in fog computing: A survey of algorithms and optimization techniques," *Computer Networks*, vol. 214, p. 109137, 2022, doi: 10.1016/j.comnet.2022.109137.

[6] G. K. Walia, M. Kumar, and S. S. Gill, "AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges, and future perspectives," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 619–669, 2024, doi: 10.1109/COMST.2023.3338015.

[7] 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, p. 109662, 2023, doi: 10.1016/j.comnet.2023.109662.

[8] S. Latif, W. Boulila, A. Koubaa, Z. Zou, and J. Ahmad, "DTL-IDS: An optimized intrusion detection framework using deep transfer learning and genetic algorithm," *Journal of Network and Computer Applications*, vol. 221, p. 103784, 2024, doi: 10.1016/j.jnca.2023.103784.

[9] D. Javeed, M. S. Saeed, M. A. Adil, P. Kumar, and A. Jolfaei, "A federated learning-based zero trust intrusion detection system for Internet of Things," *Ad Hoc Networks*, vol. 162, p. 103540, 2024, doi: 10.1016/j.adhoc.2024.103540.

[10] D. Attique, H. Wang, and P. Wang, "Fog-assisted deep-learning-empowered intrusion detection system for RPL-based resource-constrained smart industries," *Sensors*, vol. 22, no. 23, p. 9416, 2022, doi: 10.3390/s22239416.

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

[12] T. Rehman, N. Tariq, F. A. Khan, and S. U. Rehman, "FFL-IDS: A fog-enabled federated learning-based intrusion detection system to counter jamming and spoofing attacks for the Industrial Internet of Things," *Sensors*, vol. 25, no. 1, p. 10, 2025, doi: 10.3390/s25010010.

[13] H. Zhou, H. Zou, W. Li, D. Li, and Y. Kuang, "HiViT-IDS: An efficient network intrusion detection method based on vision transformer," *Sensors*, vol. 25, no. 6, p. 1752, 2025, doi: 10.3390/s25061752.

[14] Z. Abou El Houda, B. Brik, and L. Khoukhi, "Why should I trust your IDS?: An explainable deep learning framework for intrusion detection systems in Internet of Things networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1164–1176, 2022, doi: 10.1109/OJCOMS.2022.3188750.

[15] A. Hazra, P. K. Donta, T. Amgoth, and S. Dustdar, "Cooperative transmission scheduling and computation offloading with collaboration of fog and cloud for Industrial IoT applications," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3944–3953, 2023, doi: 10.1109/JIOT.2022.3150070.

[16] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, and Q. Jin, "Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 570–580, 2023, doi: 10.1109/TII.2022.3170149.

[17] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "FL-IDS: Federated learning-based intrusion detection system using edge devices for transportation IoT," *IEEE Access*, vol. 12, pp. 52215–52226, 2024, doi: 10.1109/ACCESS.2024.3386631.

[18] Z. Jin, J. Zhou, B. Li, X. Wu, and C. Duan, "FL-IIDS: A novel federated learning-based incremental intrusion detection system," *Future Generation Computer Systems*, vol. 151, pp. 57–70, 2024, doi: 10.1016/j.future.2023.09.019.

[19] M. Abd Elaziz, I. A. Fares, A. Dahou, and M. Shrahili, "Federated learning framework for IoT intrusion detection using tab transformer and nature-inspired hyperparameter optimization," *Frontiers in Big Data*, vol. 8, p. 1526480, 2025, doi: 10.3389/fdata.2025.1526480.

[20] C. D. Luu, H. H. Nguyen, V. Q. Nguyen, and N.-S. Vu, "Novel deep learning-based IoT network attack detection using magnet loss optimization," *Internet of Things*, vol. 33, p. 101680, 2025, doi: 10.1016/j.iot.2025.101680.

[21] I. Attiya, M. Abd Elaziz, and I. Issawi, "An improved hunger game search optimizer based IoT task scheduling in cloud–fog computing," *Internet of Things*, vol. 26, p. 101196, 2024, doi: 10.1016/j.iot.2024.101196.

[22] T. Tsokov and H. Kostadinov, "Dynamic network-aware container allocation in Cloud/Fog computing with mobile nodes," *Internet of Things*, vol. 26, p. 101211, 2024, doi: 10.1016/j.iot.2024.101211.

[23] S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," *Internet of Things*, vol. 26, p. 101212, 2024, doi: 10.1016/j.iot.2024.101212.