International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 10, No. 02, 2026 ♦ PP. 18–26

Article DOI: https://doi.org/10.54216/IJWAC.100203

FULL LENGTH ARTICLE

Tiny Intelligence in Fog-Assisted Wireless Sensor IoT Networks: A Review of Deployment Patterns, Resource Trade-offs, and Open Challenges

Aygul Z. Ibatova[1,*] **Baumuratova Dilaram**[2]

[1] Tyumen Industrial University, Russia

[2] Astana International University, Kazakhstan

Emails: aigoul@rambler.ru · Baumuratova.d@gmail.com

Received: February 05, 2026 Revised: March 09, 2026 Accepted: May 12, 2026 ⋆ Corresponding author

ABSTRACT

Wireless sensor IoT networks are moving from simple measurement pipelines toward distributed systems where sensing, interpretation, filtering, and coordination are divided across devices, fog nodes, and cloud services. This review examines that transition through the lens of tiny intelligence, with special attention to how small models, local event filters, federated updates, service placement, and privacy controls reshape fog-assisted wireless sensor deployments. The paper does not treat fog computing as a generic latency layer. Instead, it studies fog as a governance and orchestration layer that decides which data should stay at the device, which events should be aggregated nearby, and which models require cloud-level supervision. A structured comparison of prior studies is provided across architecture, TinyML, federated learning, placement, security, benchmarking, and lifecycle coverage. The synthesis shows that the literature has matured in modelling fog resources and building lightweight inference functions, but remains fragmented in lifecycle management, cross-layer wireless awareness, privacy accounting, and reproducible evaluation. The review concludes with a research agenda for sensor-to-fog intelligence pipelines that are adaptive, auditable, energy-aware, and suitable for long-lived cyber-physical deployments.

Keywords: Fog computing ▪ Wireless sensor networks ▪ Internet of Things ▪ TinyML ▪ Edge intelligence ▪ Federated learning ▪ Resource-aware orchestration

1. INTRODUCTION

Wireless sensor IoT devices now sit at the front line of cyber-physical observation. They measure vibration, sound, light, pressure, motion, soil moisture, temperature, traffic, occupancy, air quality, and device telemetry. Foundational work by Akyildiz et al. framed wireless sensor networks around sensing, communication, and coordination constraints [1]. Later surveys extended this view by detailing deployment requirements and protocol issues [2], while energy-aware studies showed that network lifetime remains a central limitation in unattended sensing [3, 4]. The present generation of sensor systems adds a second requirement: the device, gateway, and fog node must decide what the data mean before all packets are moved to distant servers. This shift changes the role of fog computing from a mere relay tier into a distributed intelligence layer.

The conventional cloud-centred IoT stack is effective for large-scale storage and historical analytics, but it is less suitable when applications require local reaction or reduced bandwidth consumption. Atzori et al. established the IoT as a heterogeneous network of identifiable objects and services [5], and Gubbi et al. later connected that vision to cloud-

supported sensing and analytics [6]. Al-Fuqaha et al. clarified the enabling protocols and application layers of IoT systems [7], while Botta et al. discussed how cloud platforms integrate with IoT workloads [8]. Fog and edge computing were introduced to place computation, storage, and networking services closer to data-producing devices [9, 10]. Subsequent fog surveys refined this architecture by examining resource management, service composition, mobility, and the relationship between fog, cloudlets, and mobile edge computing [11, 12]. Bellavista et al. emphasised fog computing for IoT applications and pervasive environments [13]. Naha et al. further summarised the architectural requirements and research directions of fog systems [15]. Yet many sensor deployments still apply a coarse design rule: either transmit to a gateway or execute a fixed lightweight model locally. That rule is no longer sufficient.

Recent developments in TinyML have made it possible to deploy useful inference functions on microcontroller-class devices [27, 28]. In parallel, edge-intelligence studies have described how learning functions can be divided across nearby computation tiers instead of being confined to remote cloud servers [25, 26]. Federated learning adds another layer by allowing distributed devices or fog nodes to contribute model updates without transferring raw data [30, 29]. The technical challenge is not simply whether a model fits in memory. It is whether the model should be executed at the sensor, at a fog node, at an edge server, or in the cloud under changing wireless conditions, battery states, privacy constraints, and data novelty. In this review, the phrase *tiny intelligence* denotes this distributed decision capability across small devices and nearby fog resources.

This paper differs from prior fog-IoT reviews in three ways. First, it focuses on wireless sensor IoT devices rather than generic mobile or edge clients. Second, it treats TinyML, federated learning, service placement, security, and evaluation tools as parts of a single operational pipeline. Third, it uses compact mathematical expressions to relate latency, energy, memory, and privacy pressure without introducing an empirical classifier or performance test. The review is therefore intended as a design-oriented synthesis for researchers planning fog-assisted sensor intelligence rather than another broad catalogue of fog architectures.

The rest of the paper is organised as follows. Section 2 describes the review scope and synthesis lens. Section 3 compares representative prior studies. Section 4 positions the cloud-to-things continuum for wireless sensor IoT. Section 5 presents a taxonomy of intelligence placement. Section 6 reviews TinyML and federated learning for constrained devices. Section 7 discusses orchestration, placement, and offloading. Section 8 summarises security and privacy concerns. Section 9 reviews evaluation tools and benchmark evidence. Section 10 provides a critical analysis of the reviewed contributions. Section 11 identifies application lessons and open challenges, and Section 12 concludes the paper.

2. REVIEW SCOPE AND SYNTHESIS LENS

This review follows a thematic mapping approach. The objective is not to count publications in a bibliometric manner, but to organise mature and recent studies around decisions that repeatedly appear in fog-assisted sensor systems: where to execute, what to transmit, when to aggregate, how to protect data, and how to evaluate deployments. Foundational IoT and wireless sensor network surveys are used to establish the problem background [5, 1]. The fog and MEC literature then defines the middle computing layer and its communication-computation constraints [17, 18]. Resource-management studies add the placement and load-balancing view needed for operational fog systems [20]. More recent work on TinyML clarifies how learning functions can move to small devices [28]. Security and privacy studies are used to interpret the governance risks that appear when models and data are distributed [32, 35].

The reviewed studies were coded into six themes: architecture, sensor intelligence, service placement, privacy and trust, tool support, and benchmark data. Table 1 summarises the lens used in the paper. The coding deliberately separates *device-level intelligence* from *fog-level orchestration*, because these are often merged under the generic phrase edge AI. In practice, a microcontroller executing a quantised anomaly detector has very different constraints from a gateway performing federated aggregation or a fog broker selecting a service placement.

Table 1. Review lens used to organise the literature.

Theme	Main question	Representative sources
Architecture	How is computation distributed across sensor, fog, edge, and cloud tiers?	Fog/edge surveys and architectural studies [11, 12]
Tiny intelligence	What can realistically run on small sensor nodes?	TinyML and edge intelligence reviews [27, 28]
Placement	Where should tasks and services be deployed?	Resource management and service placement surveys [20, 24]
Privacy and trust	How are data, models, and updates protected?	Security, privacy, and FL works [32, 35]
Evaluation	Which tools and datasets support reproducibility?	Simulators and IoT security datasets [36, 40]

3. COMPARATIVE REVIEW OF PRIOR STUDIES

The literature around fog-assisted wireless sensor IoT has expanded in several directions rather than along one unified line. Early wireless sensor and IoT surveys established the importance of energy, connectivity, protocol heterogeneity, and application diversity. Fog and edge computing studies then shifted the emphasis toward locality, distributed resource management, and low-latency service provisioning. More recent work on TinyML and federated learning has introduced a different question: how much intelligence can be moved closer to the signal without weakening privacy, increasing maintenance cost, or overloading small devices.

Table 2 consolidates these strands. The table is intentionally expressed as a compact yes/no mapping so that differences among the studies are visible without long descriptive entries. A “Yes” value indicates that the study gives explicit attention to the corresponding dimension; a “No” value indicates that the issue is outside the main scope or appears only marginally. The comparison shows that architecture surveys are strong on fog/cloud structure but weak on tiny on-device intelligence, while TinyML and federated-learning studies are strong on model execution but less detailed on fog placement and long-term orchestration.

Table 2. Comparative mapping of representative studies in fog-assisted wireless sensor IoT intelligence.

Study	WSN	Fog	TinyML	FL	Placement	Security	Benchmark	Lifecycle	Main orientation
Akyildiz et al. [1]	Yes	No	No	No	No	No	No	No	Wireless sensor foundations
Yick et al. [2]	Yes	No	No	No	No	Yes	No	No	Sensor network design issues
Rault et al. [3]	Yes	No	No	No	No	No	No	No	Energy efficiency in WSNs
Yetgin et al. [4]	Yes	No	No	No	No	No	No	No	Lifetime maximisation
Atzori et al. [5]	Yes	No	No	No	No	Yes	No	No	IoT enabling view
Gubbi et al. [6]	Yes	No	No	No	No	No	No	No	Cloud-centric IoT vision
Al-Fuqaha et al. [7]	Yes	No	No	No	No	Yes	No	No	IoT technologies and protocols
Shi et al. [9]	No	Yes	No	No	Yes	Yes	No	No	Edge computing vision
Satyanarayanan [10]	No	Yes	No	No	Yes	No	No	No	Edge/cloudlet perspective
Mouradian et al. [11]	No	Yes	No	No	Yes	Yes	No	No	Fog architecture and challenges
Yousefpour et al. [12]	No	Yes	No	No	Yes	Yes	No	No	Fog and related paradigms
Puliafito et al. [14]	Yes	Yes	No	No	Yes	No	No	No	Fog for IoT services
Hong and Varghese [20]	No	Yes	No	No	Yes	No	No	No	Fog/edge resource management
Kumari et al. [21]	No	Yes	No	No	Yes	No	No	No	Task offloading algorithms
Al-Asadi and Al-mamory [23]	No	Yes	No	No	Yes	No	No	No	Edge/fog node placement
Apat et al. [24]	No	Yes	No	No	Yes	No	No	No	Service placement optimisation
Dutta and Bharali [27]	Yes	No	Yes	No	No	Yes	No	No	TinyML and IoT
Heydari and Mahmoud [28]	Yes	No	Yes	No	No	No	No	No	On-device inference review
Nguyen et al. [30]	Yes	Yes	No	Yes	No	Yes	No	No	Federated learning for IoT
Imteaj et al. [29]	Yes	Yes	No	Yes	No	Yes	No	No	Resource-constrained FL
Mothukuri et al. [32]	No	Yes	No	Yes	No	Yes	No	No	FL security and privacy
Gupta et al. [36]	Yes	Yes	No	No	Yes	No	Yes	No	Fog simulation toolkit
Ferrag et al. [39]	Yes	Yes	No	Yes	No	Yes	Yes	No	IoT/IIoT security dataset
Neto et al. [40]	Yes	No	No	No	No	Yes	Yes	No	Large-scale IoT attacks

The comparison also reveals a gap that motivates the rest of this review. Many studies treat one layer of the continuum very carefully, but few connect device-level model constraints, wireless communication cost, fog placement, privacy policy, and model lifecycle into one deployable view. This separation is understandable because each topic has its own technical depth. However, in wireless sensor IoT deployments, these dimensions interact in a single operational pipeline: a smaller local model may save bandwidth, but it may also create update-management and drift-detection responsibilities for the fog tier.

4. CLOUD-TO-THINGS CONTINUUM FOR SENSOR IOT

A wireless sensor IoT system can be interpreted as a continuum rather than a stack. At one end, sensors have severe power and memory limits but direct access to raw physical signals. At the other end, cloud platforms provide long-term storage and compute capacity but suffer from delay, backhaul dependence, and privacy exposure. The fog tier mediates between these extremes by hosting gateways, roadside units, industrial controllers, base-station servers, local micro-data centres, and cooperative edge nodes. This interpretation is consistent with the edge-computing vision of Shi et al. [9] and the cloudlet-oriented view of Satyanarayanan [10].

For sensor IoT, the key design issue is the cost of moving information. A raw stream $s_i(t)$ from sensor i may contain redundant samples, local noise, private data, and event-free periods. Sending all samples to the cloud imposes communication energy and latency cost. Keeping all analysis on the sensor may reduce traffic but can increase local computation cost and make model maintenance difficult. A simple placement score for a task k can be written as:

$$C_k(p) = \alpha L_k(p) + \beta E_k(p) + \gamma M_k(p) + \delta R_k(p), \quad (1)$$

where p is the execution place, L_k is latency, E_k is energy, M_k is memory pressure, and R_k is privacy or risk exposure. The weights reflect the application. A fall detector, for example, should assign high weight to latency; an industrial condition-monitoring system may give greater weight to false alarms, model drift, and maintenance windows.

Figure 1 summarises the continuum used in this review. It avoids the common binary cloud-versus-edge distinction by separating sensor, fog, and cloud responsibilities. In this reading, the fog layer is not only a smaller cloud. It performs context fusion, model caching, local policy enforcement, and selective aggregation.

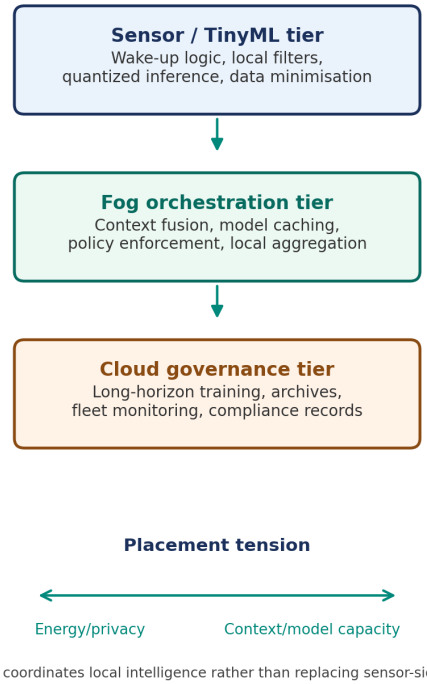

Figure 1. Placement taxonomy for tiny intelligence in fog-assisted wireless sensor IoT systems.

The continuum view also helps explain why fog computing is frequently paired with mobile edge computing and edge intelligence. MEC studies emphasise radio-computation in-

teraction and offloading decisions [17, 18]. Broader MEC surveys add the role of 5G and beyond integration [19], while edge-intelligence research connects these resources to local learning functions [25]. Sensor IoT systems need both: wireless-aware scheduling and resource-aware orchestration.

5. TAXONOMY OF INTELLIGENCE PLACEMENT

The first family of deployment patterns is *sensor-first intelligence*. A small model, rule set, or thresholding mechanism runs on the sensing device and transmits only events or compressed descriptors. TinyML surveys show that this pattern can reduce latency and bandwidth while improving privacy because raw data remain close to the source [27, 28]. However, the pattern is fragile when models must adapt to new device behaviour, environmental change, or rare events.

The second family is *fog-mediated intelligence*. Sensor data are first filtered locally and then interpreted at gateways or fog nodes. This pattern is common in smart buildings, industrial monitoring, and environmental sensing where several sensors contribute to one decision. Fog nodes can fuse multiple streams, maintain local state, and execute heavier models than microcontrollers. Puliafito et al. describe this fog role from an IoT service perspective [14], while Habibi et al. review the architectural components that support such deployments [16]. In practice, this pattern also depends on offloading and task-allocation methods such as those surveyed by Kumari et al. [21]. It also enables rapid model replacement without reflashing every device.

The third family is *collaborative learning*. Federated learning and split learning distribute model training or inference without centralising raw data. Nguyen et al. present the broader IoT federated-learning landscape [30], while Imteaj et al. focus on resource-constrained devices [29]. Brecko et al. further position federated learning in edge-computing environments [31]. In wireless sensor IoT, this is attractive for privacy-preserving analytics but difficult under unstable participation, small batteries, non-IID sensor data, and constrained uplinks. The literature therefore suggests a hybrid arrangement: sensors perform feature extraction or local inference, fog nodes aggregate or personalise models, and the cloud maintains long-horizon learning.

Table 3. Comparison of intelligence placement patterns.

Pattern	Best use	Main limitation	Typical tier
Sensor-first	Event detection, wake-up logic, privacy-sensitive signals	Memory and retraining limits	Sensor
Fog-mediated	Multi-sensor fusion and local coordination	Placement and load management	Gateway/fog node
Collaborative	Privacy-preserving model improvement	Communication and drift overhead	Sensor-fog-cloud
Cloud-governed	Fleet-scale analytics and archival learning	Delay and data movement cost	Cloud

A useful placement rule is to compare the cost of local inference with the cost of transmission plus remote inference:

$$D_{local} = T^{s}_{inf} + E^{s}_{cpu}, \quad D_{remote} = T_{tx} + T^{f}_{queue} + T^{f}_{inf} + E^{s}_{radio}. \tag{2}$$

Sensor-first inference is preferred when $D_{local} < D_{remote}$ and the confidence of the local model is adequate. Fog-assisted inference is preferred when the device lacks model capacity or when a decision requires more than one sensor stream.

6. TINYML AND FEDERATED LEARNING IN THE FOG

TinyML expands the feasible design space for wireless sensors by moving inference to microcontroller-class hardware [27, 28]. The main benefit is not only low latency; it is the ability to convert continuous sensing into sparse semantic events. For example, a vibration sensor can transmit a bearing-health state instead of a raw waveform; an acoustic sensor can transmit a detected pattern rather than an audio trace. This changes network load from sample-rate dependent to event-rate dependent.

Federated learning addresses a different part of the lifecycle. It allows multiple devices or fog nodes to contribute to a model without sharing raw samples. This principle is well documented for IoT-scale federated learning [30] and has been examined more specifically for constrained edge and IoT devices [29, 31]. In principle, this is well aligned with distributed IoT. In practice, the sensor setting is challenging because data are heterogeneous and device participation is intermittent. A fog node can reduce this burden by acting as a local aggregator, caching updates, filtering unreliable participants, and coordinating communication rounds.

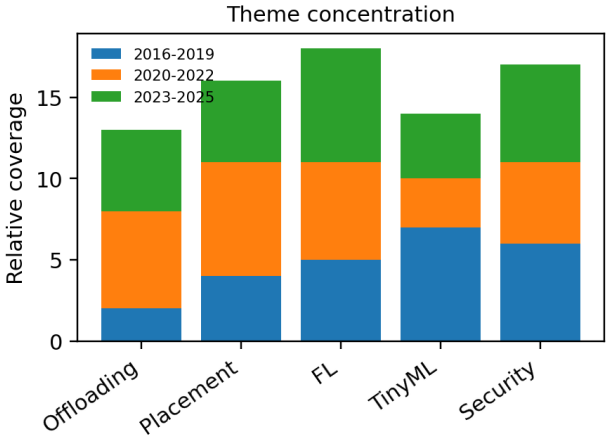

Figure 2. Distribution of major themes across three literature periods.

Figure 2 shows that the reviewed literature moved from infrastructure and offloading questions toward learning, privacy, and security. Placement and federated learning receive stronger attention after 2020, which reflects the need to manage intelligence across sensor, fog, and cloud tiers rather than treating fog nodes as passive gateways.

Figure 3 separates the execution tiers by the decision pressure they are best suited to absorb. The cloud remains valuable for long-horizon analysis, but it is least suitable when autonomy and privacy dominate. Sensor-side inference is strongest under autonomy and privacy constraints, while fog computing occupies the middle region where low delay, energy control, and local coordination must be balanced.

Figure 4 highlights why tiny intelligence is attractive but incomplete. Sensor-first execution is strong for privacy and energy-aware filtering because raw signals can be reduced before transmission. Its weaker coverage and maintainability scores indicate that fog nodes are still needed for neighbourhood context, model versioning, rollback, and long-term coordination.

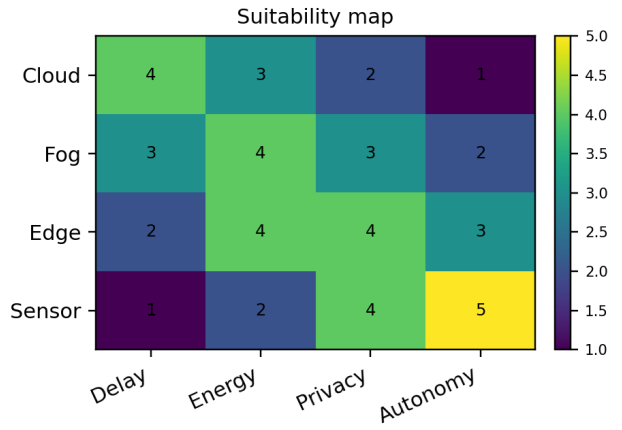

Figure 3. Suitability of execution tiers under delay, energy, privacy, and autonomy concerns.

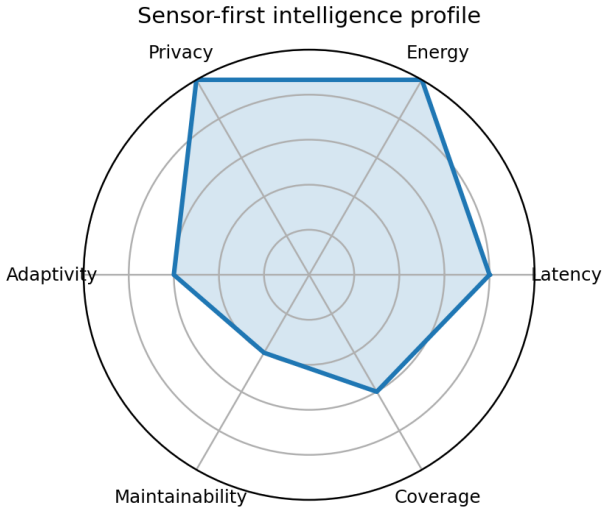

Figure 4. Sensor-first intelligence profile across six design criteria.

Figure 5 reflects the increasing emphasis on distributed learning from 2016 onward. The trend does not imply that every sensor should train a model. Rather, it shows that the field increasingly treats intelligence as a cooperative function distributed across devices, fog brokers, and cloud services. This is why model placement, update scheduling, and privacy-aware aggregation are now core design issues in fog-assisted wireless sensor IoT.

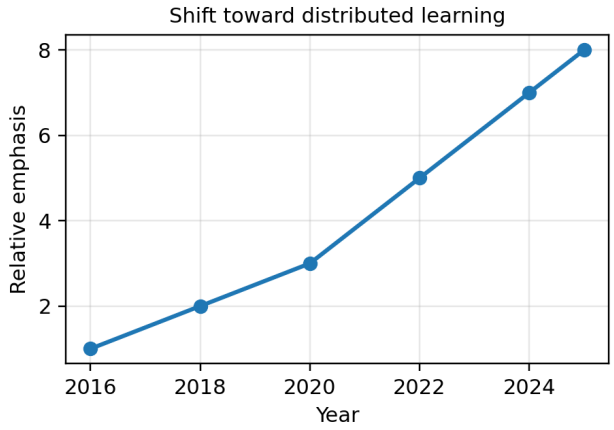

Figure 5. Shift in the literature toward distributed learning in sensor-to-fog systems.

A compact model-update budget for federated sensor learning can be written as:

$$B_{round} = \sum_{i \in \mathcal{S}_r} \left(|\theta_i| q_i + h_i \right), \tag{3}$$

where $\mathcal{S}_r$ is the set of participating devices in round r, $|\theta_i|$ is the number of transmitted parameters or compressed update entries, q_i is the number of bits per entry, and h_i represents protocol overhead. This expression shows why sparse updates, quantisation, and fog-level aggregation are central to resource-constrained federated sensor systems.

7. ORCHESTRATION, OFFLOADING, AND SERVICE PLACEMENT

Fog orchestration concerns where tasks should be placed, when they should migrate, and how limited resources are shared across applications. Surveys of resource management show that fog systems must jointly consider computation, communication, storage, and mobility [20]. Task-offloading and continuum-allocation studies extend this argument to optimisation under heterogeneous capacity and demand [21, 22]. Service placement studies further show that placement is often multi-objective: minimising latency alone can overload nearby nodes or increase energy consumption [23, 24].

Table 4 summarises representative studies across the continuum. The table is intentionally compact to preserve the journal layout while still distinguishing architecture, learning, security, and evaluation contributions.

Table 4. Representative studies and their relevance to fog-assisted sensor intelligence.

Study	Focus	Contribution to this review	Tier
Mouradian et al. [11]	Fog survey	Architecture and algorithms	Fog
Yousefpour et al. [12]	Fog taxonomy	Links fog, MEC, cloudlet	Fog-cloud
Puliafito et al. [14]	Fog for IoT	IoT service perspective	Fog
Hong and Varghese [20]	Resource mgmt.	Placement and load issues	Fog-edge
Kumari et al. [21]	Offloading	Optimisation process	Fog
Vergara et al. [22]	Allocation	Fog-cloud continuum	Fog-cloud
Dutta and Bharali [27]	TinyML	On-device inference	Sensor
Imteaj et al. [29]	FL for IoT	Device constraints	Sensor-fog
Neto et al. [40]	IoT dataset	Security benchmark scale	Evaluation
Lera et al. [38]	Simulator	Fog modelling support	Evaluation

A practical orchestration model can be expressed as a constrained placement problem. Let $x_{k,p} \in \{0,1\}$ indicate whether task k is assigned to place p. A simplified objective is:

$$\min_{x} \sum_{k} \sum_{p} x_{k,p} \left(\alpha L_{k,p} + \beta E_{k,p} + \gamma U_p \right), \tag{4}$$

subject to memory, CPU, and deadline constraints:

$$\sum_{k} x_{k,p} m_k \le M_p, \quad \sum_{k} x_{k,p} c_k \le C_p, \quad L_{k,p} \le d_k. \tag{5}$$

Here U_p represents utilisation or congestion at place p. The formulation is intentionally simple but exposes a recurring theme: fog orchestration is a balancing problem, not a single-metric optimisation problem.

Figure 6 shows the operational decision loop in a full-width form. The key point is that placement should be revisited when device energy, link quality, queue length, or privacy constraints change. Static placement is suitable for pre-

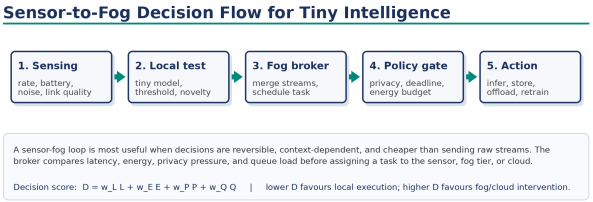

Figure 6. Decision flow for choosing local, fog, or cloud execution in sensor-intelligence pipelines.

dictable periodic sensing, but event-driven systems require adaptive switching between local inference, fog aggregation, and cloud-assisted retraining.

8. SECURITY, PRIVACY, AND TRUST DIMENSIONS

Fog-assisted sensor intelligence introduces security and privacy benefits, but it also expands the attack surface. Keeping raw data near the source can reduce exposure; however, gateways and fog nodes become attractive targets because they aggregate many sensor streams. Alwakeel highlights the security and privacy issues that arise when fog and edge resources are deployed outside tightly controlled data centres [34]. Sicari et al. further connect these issues to authentication, trust management, privacy preservation, secure communication, and availability under distributed conditions [35]. Federated learning adds a separate set of concerns, including poisoning, inference attacks, malicious model updates, and aggregation robustness [32, 33].

Security analysis in this field should therefore distinguish between data protection and model protection. Data protection asks whether raw measurements, identifiers, and locations can be inferred. Model protection asks whether learned parameters or local updates can leak private information or be manipulated by an adversary. Table 5 summarises the main risks.

Table 5. Security and privacy risks in fog-assisted sensor intelligence.

Risk	Where it appears	Mitigation direction
Raw-data leakage	Sensor-to-fog traffic and storage	Local features, encryption, minimisation
Gateway compromise	Fog nodes aggregating many devices	Trust scoring, isolation, attestation
Model poisoning	Federated or shared model updates	Robust aggregation, update screening
Inference attack	Model parameters and confidence outputs	Differential privacy, output control
Resource exhaustion	Wireless channel and fog queues	Rate control, admission, scheduling

A simple privacy pressure term can be introduced for task placement:

$$R_k(p) = \eta_1 S_k(p) + \eta_2 I_k(p) + \eta_3 V_k(p), \tag{6}$$

where S_k denotes sensitivity of the raw signal, I_k represents identity or location exposure, and V_k captures vulnerability of the receiving tier. A high value of $R_k(p)$ discourages sending raw data to that tier and favours feature extraction or local decision-making.

9. EVALUATION TOOLS AND BENCHMARK EVIDENCE

Evaluation remains a weak point in fog-enabled sensor intelligence. iFogSim enabled repeatable resource-management experiments for IoT, edge, and fog environments [36]. EdgeCloudSim extended this type of evaluation toward mobility and edge-system performance [37], while YAFS supports graph-based modelling of IoT and fog scenarios [38]. These tools are most valuable when researchers report placement assumptions, node capacities, data rates, mobility patterns, and scheduling policies. Without these details, fog results become difficult to compare.

Security and anomaly-detection evaluations depend on datasets. Edge-IIoTset and CICIoT2023 provide examples of IoT-oriented traces and attack scenarios [39, 40]. These datasets are useful for studying traffic-level anomalies, but they do not fully represent the closed-loop behaviour of sensor-fog systems where decisions change the future traffic. Future benchmarks should include both network traces and orchestration decisions.

Table 6. Evaluation resources used in fog and IoT security studies.

Resource	Type	Use in future studies
iFogSim [36]	Fog simulation toolkit	Placement and latency scenarios
EdgeCloudSim [37]	Edge simulation environment	Mobility and load analysis
YAFS [38]	Fog simulation framework	Graph-based IoT experiments
Edge-IIoTset [39]	Cybersecurity dataset	Centralised/federated IDS studies
CICIoT2023 [40]	Large-scale IoT attack dataset	Security analytics benchmark

Reproducible evaluation should include at least four layers of reporting: device-class constraints, wireless link assumptions, fog-resource capacities, and learning lifecycle assumptions. The absence of any one layer can make a result look better than it would be in deployment. For example, a model may appear energy-efficient if communication cost is ignored, or privacy-preserving if metadata and model-update leakage are not considered.

10. CRITICAL ANALYSIS OF REVIEWED CONTRIBUTIONS

The review indicates that the field has accumulated many useful components, but it has not yet converged on a complete engineering methodology. Architecture studies define where fog resources may be placed; placement studies optimise where services should run; TinyML studies show that inference can be moved onto small devices; federated-learning studies reduce raw-data centralisation; and simulator papers provide controlled environments. The weakness is that these contributions often stop at their own boundary. A service-placement paper may not model model drift; a TinyML paper may not include fog queueing; a security paper may not account for sensor battery limits; and a dataset paper may not include orchestration decisions.

Table 7 provides a critical yes/no assessment of contribution coverage. The intention is not to rank papers mechanically, but to expose what is commonly missing when the studies are viewed from the perspective of an end-to-end fog-assisted wireless sensor deployment. The recurring weaknesses are lifecycle governance, realistic wireless awareness, and privacy accounting beyond encryption or raw-data avoidance.

Table 7. Critical assessment of contribution coverage in the reviewed literature.

Study group	Clear taxonomy	Resource model	Wireless-aware	Fog orchestration	Privacy depth	Reproducible tools	Lifecycle view	Deployment guidance	Critical observation
WSN foundations [1, 2]	Yes	Yes	Yes	No	No	No	No	Yes	Strong sensing basis but pre-fog framing
Energy/lifetime studies [3, 4]	Yes	Yes	Yes	No	No	No	No	Yes	Energy is rigorous, intelligence lifecycle is absent
General IoT surveys [5, 7]	Yes	No	Yes	No	Yes	No	No	Yes	Broad but not placement-specific
Cloud-IoT integration [8]	Yes	Yes	No	No	Yes	No	No	Yes	Cloud value is clear, local decision logic is limited
Edge/fog visions [9, 10]	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Foundational but not sensor-specific
Fog surveys [11, 12]	Yes	Yes	No	Yes	Yes	No	No	Yes	Good architecture coverage, weak model lifecycle detail
MEC surveys [17, 18]	Yes	Yes	Yes	Yes	No	No	No	Yes	Strong communication-computation view, less sensor governance
Resource management [20, 21]	Yes	Yes	No	Yes	No	No	No	Yes	Optimisation is rich, but privacy and drift are secondary
Placement studies [23, 24]	Yes	Yes	No	Yes	No	No	No	Yes	Mature placement taxonomy, limited tiny-device constraints
Edge intelligence [25, 26]	Yes	Yes	No	Yes	Yes	No	No	Yes	Good AI-at-edge view, not enough WSN specificity
TinyML reviews [27, 28]	Yes	Yes	Yes	No	Yes	No	No	Yes	Device inference is strong, fog orchestration is thin
Federated-learning reviews [30, 29]	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Privacy-aware learning, but high coordination overhead
FL security reviews [32, 33]	Yes	No	No	Yes	Yes	No	No	Yes	Good threat discussion, less resource realism
Fog security surveys [34, 35]	Yes	No	No	Yes	Yes	No	No	Yes	Security coverage is high, control-loop analysis is limited
Simulation tools [36, 38]	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Useful experimentation, but model governance is not central
Dataset studies [39, 40]	Yes	No	Yes	No	Yes	Yes	No	Yes	Strong security traces, weak orchestration context

The critical reading suggests that future review and experimental work should avoid treating each contribution type as sufficient by itself. TinyML without fog governance becomes difficult to update. Fog placement without wireless and device constraints risks unrealistic optimisation. Federated learning without energy and attack modelling can overstate privacy-preserving benefits. A stronger research design would report the sensing task, local model budget, communication policy, fog placement rule, privacy objective, and maintenance plan together.

11. APPLICATION DOMAINS AND DESIGN LESSONS

Wireless sensor IoT devices appear in industrial monitoring, healthcare, agriculture, smart buildings, environmental observation, transportation, and infrastructure maintenance. In industrial settings, fog nodes are valuable because they can align multi-sensor evidence with local process states. In healthcare, sensor-first processing can reduce the exposure of personal signals, while fog nodes can support context fusion and emergency response. In agriculture and environmental sensing, energy and coverage constraints dominate; intelligence should reduce transmissions without hiding rare events.

Across these domains, the reviewed literature suggests five lessons. First, not every model should be pushed to the sensor; some decisions require neighbourhood context. Second, fog nodes should manage model versions and policies, not only computation. Third, wireless-link quality must be part of placement decisions. Fourth, privacy cannot be added after deployment; it changes where data and models should move. Fifth, benchmark datasets should be paired with realistic orchestration assumptions.

Table 8. Domain-specific implications for wireless sensor IoT deployments.

Domain	Fog value	Main caution
Industrial monitoring	Multi-sensor fusion and local alarms	Avoid model drift under process changes
Healthcare sensing	Privacy-preserving early decision	Secure identity and metadata leakage
Agriculture	Bandwidth reduction and long battery life	Rare events must not be suppressed
Smart buildings	Local comfort and energy control	Heterogeneous device ownership
Infrastructure	Field-side inspection and alert routing	Connectivity gaps and maintenance cost

12. OPEN CHALLENGES AND RESEARCH AGENDA

The first open challenge is lifecycle management. Most papers evaluate model deployment or placement at one point in time. Real deployments require monitoring, versioning, recalibration, rollback, and retirement of models. Fog nodes are well positioned to manage this lifecycle, but standard interfaces for model provenance and update control are still immature.

The second challenge is cross-layer adaptation. Wireless sensor IoT systems are affected by physical signal quality, MAC scheduling, routing, queueing, model confidence, and application deadlines. Treating these layers separately produces brittle deployments. A promising direction is to design cross-layer controllers that observe link quality and application confidence together.

The third challenge is privacy accounting. Local inference and federated learning reduce raw data movement, but they do not automatically guarantee privacy. Metadata, model updates, and confidence scores can still leak information. Future work should combine differential privacy, secure aggregation, and fog-level policy enforcement without overwhelming constrained devices.

The fourth challenge is benchmark realism. Current datasets are valuable but mainly support traffic or classification experiments. Future benchmarks should describe how sensor events affect network traffic and how fog decisions affect subsequent samples. Such benchmarks would support fairer evaluation of event-driven sensing, selective transmission, and adaptive offloading.

The final challenge is sustainability. Tiny intelligence can reduce bandwidth, but added computation, model updates, and device maintenance can increase lifecycle cost. Future review and experimental work should include energy, carbon, device replacement, and update overhead as first-class metrics, especially for large-scale environmental and smart-city deployments.

A practical research agenda should therefore include four reporting habits. First, the device class should be stated precisely, including memory, sampling rate, radio type, and duty cycle. Second, the fog tier should be described as a finite resource with queueing, contention, and policy decisions rather than as an abstract server. Third, the learning lifecycle

should include update frequency, drift detection, rollback criteria, and privacy assumptions. Fourth, benchmarks should combine network traces with orchestration traces so that researchers can evaluate not only whether an event was detected, but also whether the system made an efficient decision about where to process it.

13. CONCLUSION

This review examined fog-assisted wireless sensor IoT systems through the lens of tiny intelligence. The central argument is that future sensor deployments should not be designed as simple pipelines that move data from devices to the cloud. They should be designed as adaptive continuums in which sensing, inference, transmission, aggregation, and retraining are placed according to latency, energy, memory, privacy, and lifecycle constraints.

The reviewed literature shows that fog computing remains essential even as TinyML makes sensors more capable. Sensors can suppress redundant traffic and provide immediate reactions, but fog nodes provide context fusion, orchestration, model governance, and privacy-aware coordination. Federated learning further extends this continuum by enabling collaborative model improvement without raw-data centralisation, although it introduces communication and security challenges. The comparative tables in this review also show that a large part of the literature remains component-centred. A deployment-ready research direction should therefore combine device-level inference, fog-level orchestration, wireless-aware placement, privacy control, and lifecycle governance in one coherent design.

A mature research agenda for fog-enabled sensor intelligence should therefore integrate resource management, model lifecycle control, privacy accounting, and reproducible evaluation. The most useful future systems will be those that decide not only what a sensor reading means, but also where that meaning should be computed, how long it should be retained, and when it should reshape the behaviour of the network.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002, doi: 10.1016/S1389-1286(01)00302-4.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008, doi: 10.1016/j.comnet.2008.04.002.

[3] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014, doi: 10.1016/j.comnet.2014.03.027.

[4] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017, doi: 10.1109/COMST.2017.2650979.

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[8] A. Botta, W. de Donato, V. Persico, and A. Pescape, "Integration of cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016, doi: 10.1016/j.future.2015.09.021.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.

[10] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017, doi: 10.1109/MC.2017.9.

[11] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.

[12] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.

[13] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the Internet of Things," *Pervasive and Mobile Computing*, vol. 52, pp. 71–99, 2019, doi: 10.1016/j.pmcj.2018.12.007.

[14] C. Puliafito, E. Mingozzi, C. Vallati, F. Longo, and G. Merlino, "Fog computing for the Internet of Things: A survey," *ACM Transactions on Internet Technology*, vol. 19, no. 2, Art. 18, 2019, doi: 10.1145/3301443.

[15] R. K. Naha *et al.*, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018, doi: 10.1109/ACCESS.2018.2866491.

[16] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog computing: A comprehensive architectural survey," *IEEE Access*, vol. 8, pp. 69105–69133, 2020, doi: 10.1109/ACCESS.2020.2983253.

[17] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017, doi: 10.1109/COMST.2017.2682318.

[18] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017, doi: 10.1109/COMST.2017.2745201.

[19] Q.-V. Pham *et al.*, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020, doi: 10.1109/ACCESS.2020.3001277.

[20] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys*, vol. 52, no. 5, Art. 97, 2019, doi: 10.1145/3326066.

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

[22] J. Vergara, M. J. Santofimia, J. C. Moya, and J. Carretero, "A comprehensive survey on resource allocation in the fog-cloud continuum," *Sensors*, vol. 23, no. 9, Art. 4413, 2023, doi: 10.3390/s23094413.

[23] S. A. Al-Asadi and S. Al-mamory, "A survey on edge and fog nodes' placement methods, techniques, parameters, and constraints," *IET Networks*, 2023, doi: 10.1049/ntw2.12087.

[24] H. K. Apat, V. Goswami, B. Sahoo, R. K. Barik, and M. J. Saikia, "Fog Service Placement Optimization: A Survey of State-of-the-Art Strategies and Techniques," *Computers*, vol. 14, no. 3, Art. 99, 2025, doi: 10.3390/computers14030099.

[25] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019, doi: 10.1109/JPROC.2019.2918951.

[26] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019, doi: 10.1109/JPROC.2019.2921977.

[27] L. Dutta and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet of Things*, vol. 16, Art. 100461, 2021, doi: 10.1016/j.iot.2021.100461.

[28] S. Heydari and Q. H. Mahmoud, "Tiny machine learning and on-device inference: A survey of applications, challenges and future directions," *Sensors*, vol. 25, no. 10, Art. 3191, 2025, doi: 10.3390/s25103191.

[29] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2022, doi: 10.1109/JIOT.2021.3095077.

[30] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021, doi: 10.1109/COMST.2021.3075439.

[31] A. Brecko, E. Kajati, J. Koziorek, and I. Zolotova, "Federated learning for edge computing: A survey," *Applied Sciences*, vol. 12, no. 18, Art. 9124, 2022, doi: 10.3390/app12189124.

[32] V. Mothukuri *et al.*, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021, doi: 10.1016/j.future.2020.10.007.

[33] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021, doi: 10.1109/ACCESS.2021.3075203.

[34] A. M. Alwakeel, "An overview of fog computing and edge computing security and privacy issues," *Sensors*, vol. 21, no. 24, Art. 8226, 2021, doi: 10.3390/s21248226.

[35] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Insights into security and privacy towards fog computing evolution," *Computers & Security*, vol. 117, Art. 102682, 2022, doi: 10.1016/j.cose.2022.102682.

[36] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017, doi: 10.1002/spe.2509.

[37] C. Sonmez, A. Ozgovde, and C. Ersoy, "EdgeCloudSim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, Art. e3493, 2018, doi: 10.1002/ett.3493.

[38] I. Lera, C. Guerrero, and C. Juiz, "YAFS: A simulator for IoT scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91745–91758, 2019, doi: 10.1109/ACCESS.2019.2927895.

[39] M. A. Ferrag *et al.*, "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.

[40] E. C. P. Neto *et al.*, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, Art. 5941, 2023, doi: 10.3390/s23135941.