International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Edgefusionai: An Intelligent Edge Computing Framework for Realtime Iot Data Processing Using Federated Learning

Dr N.Umamaheswari[1], Dr.K.Gayathri[2]
[1](Assistant Professor, Hod I/c BCA Tagore college of arts and science
No.7,Works road, chrompet, Chennai -44
umatag@gmail.com)
[2](Assistant Professor Computer Applications Tagore College of Arts and Science
gaibhu79@gmail.com**)**

Abstract—The Internet of Things (IoT) links sensors, machines, vehicles and everyday objects so that they can exchange data and act with little human intervention. Fifth-generation (5G) networks introduced dedicated support for massive machine-type and ultra-reliable low-latency communication, but emerging applications such as autonomous transport, remote surgery, immersive extended reality and large-scale digital twins demand still higher data rates, lower latency, greater reliability, precise positioning and far denser connectivity. Sixth-generation (6G) networks, now being defined under the IMT-2030 framework, aim to meet these needs and to make artificial intelligence a native part of the network. This survey reviews 6G-enabled IoT. It summarises the requirements and key performance indicators, and explains the enabling technologies: terahertz and sub-terahertz communication, intelligent reflecting surfaces, integrated sensing and communication, edge intelligence, AI-native network management and non-terrestrial networks. A five-layer architecture for 6G-enabled IoT is presented, linking devices, access, edge, core and applications under a cross-layer security and energy plane. Applications in smart cities, connected vehicles, healthcare, industrial automation, precision agriculture and environmental monitoring are discussed. Finally, open challenges including infrastructure cost, spectrum regulation, energy consumption, security and privacy, and the management of billions of devices are identified.

Index Terms—6G, Internet of Things, IMT-2030, terahertz communication, intelligent reflecting surface, integrated sensing and communication, edge computing, artificial intelligence, network security.

I. INTRODUCTION

The rapid adoption of Internet of Things (IoT) solutions in industrial, healthcare and urban settings has resulted in an inherent trade-off between data production and data processing [2][4]. Sensor-based edge devices, from wearable trackers to industrial gateways, generate unceasing data streams which must be interpreted in real time for tasks including anomaly detection and predictive maintenance [1][9]. Traditional cloud-centric designs involve routing of these data streams via wide-area networks to centralized data centers which results in propagating delays which often do not meet the stringent millisecond-level deadlines of closed-loop control systems [1]. Apart from the latency factor, the combination of bandwidth exhaustion, network intermittency, and data sovereignty constraints make cloud computing inadequate for new IoT use cases [4][5].

Edge computing represents a solution to this problem via moving the computation close to data sources [1][9]. By performing computations in local network subdomains, edge systems are able to minimize the latency involved in sending data and perform independently of cloud availability [9]. However, traditional edge intelligence methods usually presuppose the use of pre-trained models or sending raw data to nearby aggregation points. The first solution is lacking adaptivity to local environment, the latter reintroduces the same privacy problems [2][4].

The problem outlined above is satisfactorily addressed by Federated Learning which allows the cooperative training of models without transferring actual data [2][4]. In the federated learning setting, nodes train locally based on their observations and share just the parameter updates to the aggregator who combines these into one global model using iterative averaging techniques [4]. This method ensures data locality and makes use of the collective intelligence of distributed nodes [2][4]. Nevertheless, the immediate implementation of federated learning in real-time IoT applications encounters numerous challenges [1][5]. There

© 2026 IJSRET

exists great heterogeneity among edge devices in terms of their computational capabilities, power and network connectivity [2][4]. The synchronous methods require uniform participation from the clients which is very hard to achieve in practice, and asynchronous methods suffer from staleness and unstable convergence [1].

In this paper, we propose EdgeFusionAI, an approach that aims to unify the concepts of federated learning with the constraints of edge-based IoT computations. In particular, this work focuses on solving three related problems, namely resource-adaptive orchestration of the clients, time-sensitive aggregation, and model adaptation via model compression. Our main contribution is a Resource-Aware Federated Averaging algorithm that chooses clients for participation, sets appropriate aggregation weights depending on update freshness, and uses gradient compression to reduce communication cost [1][5][8][10]. The proposed approach is evaluated in terms of comparative analysis to the vanilla federated averaging algorithm and two baselines that consider resource constraints.

This paper is structured as follows. The next section presents Literature Survey on edge intelligence and federated IoT systems. We introduce the Proposed Methodology and EdgeFusionAI in detail in the following section. In the last part, we present Analysis and Comparative Discussion of the results.

II. LITERATURE SURVEY

Work on federated learning and edge computing has gained momentum after 2021 because of the realization that conventional machine learning cannot work well for IoT data [2][4]. Khan et al. proposed a comprehensive taxonomy of federated learning for IoT applications, explaining the relationship between communication overhead, privacy, and usefulness of the models [4]. The review highlights heterogeneous devices and non-identically distributed data as major challenges in the field, challenges that continue to be relevant in follow-up studies [2][4].

Another research direction involved resource-aware client selection [5]. The basic idea was that not all edge devices are equal contributors in federated learning – the ones with low computing power generate updates slowly, and the ones with bad connections might not even manage to finish training cycles [1][5]. FedSensor, introduced in 2025, solved the problem of the most severe constraints through a hierarchical approach in which extremely low-power devices acted only as data publishers without participating in training itself [3]. A second edge layer took care of training for those devices, thus separating data production from learning calculations [3]. This scheme allowed federated learning to be applied to new classes of devices but also added another layer of architecture, complicating its implementation in flatter network settings [3][5].

The requirement for real-time performance has led researchers to explore asynchronous and semi-synchronous aggregation [1][10]. In their survey of Federated Learning solutions for real-time IoT, Hakiri et al. identified seven approaches to Federated Learning, finding synchronous protocols to be problematic in the case of varying client availability [1]. They found the need for aggregating methods that are tolerant of stragglers while not causing indefinite delays for the updating of the global model [1]. Communication efficient Federated Learning solutions have been considered in the context of gradient sparsification and quantization techniques [8][10].

Time-related issues in federated IoT data present unique problems compared to those faced by federated learning benchmarks traditionally used [1][6]. Concept drift refers to the process where data distribution changes either gradually or suddenly over time, making models trained on past observations become obsolete [6]. According to the critical analysis of federated systems in cyber-physical systems in 2024, most of the approaches proposed rely on the assumption that data distributions remain stationary, which is not always true in industrial and environmental monitoring applications [6]. In addition, the complexity of cryptographic privacy methods makes them computationally infeasible [6].

Recent innovations in architecture have focused on leveraging containerization and microservices to implement federated learning at the edge [9]. FedGuard-IIoT, introduced in 2026, showed that implementation of federated elements as lightweight docker containers allowed for modular scalability and fault isolation in industry applications [7]. Their deployment on edge devices with limited resources has yielded inference latencies in the sub-second range, but their round-trip federated processing times were still in the few-second range, due to the inherently iterative process of federated aggregation [7].

Despite the aforementioned developments, there remains a need for a link between theoretical algorithms and practical

© 2026 IJSRET

considerations [2][5][6]. The vast majority of current frameworks rely on either homogenous client population or connectivity; however, neither is a realistic assumption in IoT deployments [1][2][4]. The EdgeFusionAI framework is the missing link, in that it considers variability in resources as an inherent part of its design.

III. PROPOSED METHODOLOGY

System Architecture

The design of the EdgeFusionAI system uses a hierarchy of two levels consisting of a coordinator of the federation and heterogeneous edge participants. The coordinator is located at the edge of the network, either on the gateway or local server, and manages the global state of the model, coordinates training rounds, and aggregation. Edge participants include devices such as single-board computers and microcontroller nodes that run the local training module and telemetry agents reporting resource availability.

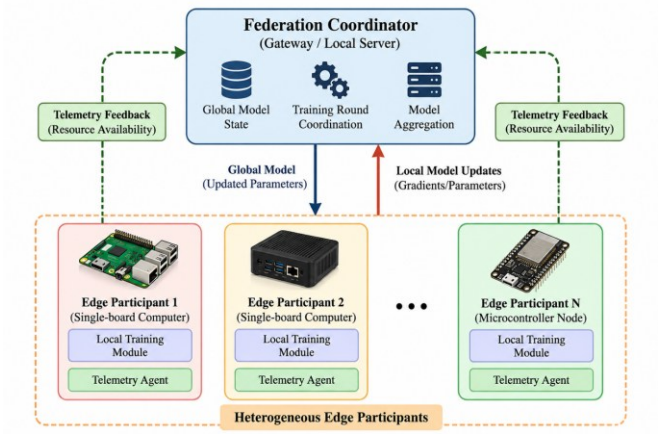


Figure 1: EdgeFusionAI system architecture

The design includes telemetry which is independent of the channel that receives model updates. Resource information such as amount of memory available, CPU usage, battery status, and link quality can be communicated at user-configurable intervals to enable the coordinator to keep an up-to-date record of the capacity of the participants.

Resource-Adaptive Federated Averaging Algorithm

The novel algorithmic design is called Resource-Adaptive Federated Averaging (RA-FedAvg). RA-FedAvg adapts FedAvg by adding three new mechanisms: capacity-weighted client selection, staleness-aware aggregation, and efficient updates encoding.

Client selection: For each client at each round, the coordinator calculates a score that includes a capacity component and an eligibility component. These two factors are computed based on the telemetry data available at the time and ensure the absence of starvation of slow clients. Then clients are sorted according to the calculated score, and a subset of clients is chosen so that the estimated expected round duration is lower than the latency budget.

Staleness-aware aggregation: Stale updates, i.e., those arrived after the deadline of aggregation, are not dropped but are used with a decaying factor depending on staleness. This aggregation function assigns weight to each update that equals the product of its sample size and a decay factor.

Update Compression: Prior to the transmission of the update deltas, magnitude-based pruning of the parameters is applied, and only the top-k% parameters by magnitude are sent through the network; the rest of the parameters will be retained in memory to use in future rounds.

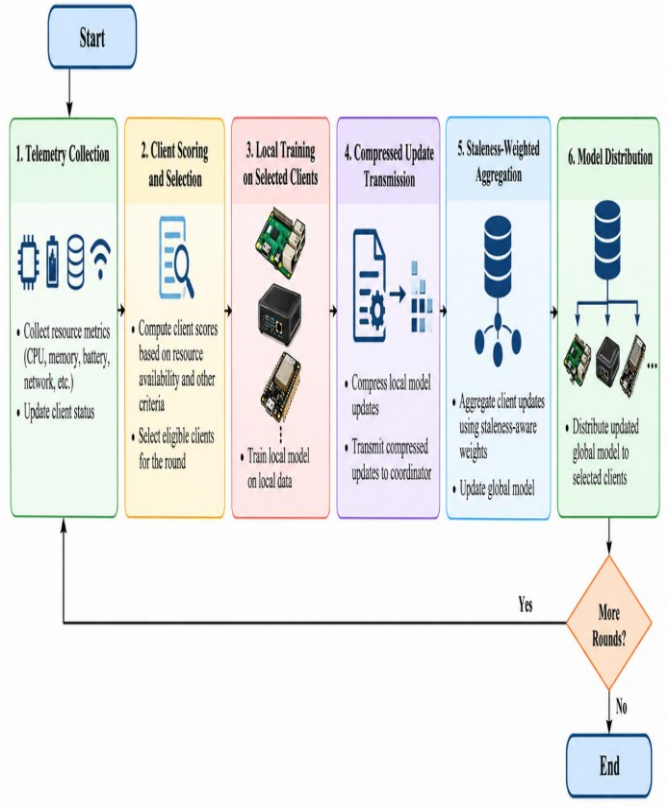


Figure 2: Flowchart of the RA-FedAvg algorithm showing the round lifecycle

© 2026 IJSRET

Algorithm: Resource-Adaptive Federated Averaging (RA-FedAvg)

Input: N clients, initial model w_0, rounds T, latency budget L, sparsity k

Output: Global model w_T

1: for round t = 1 to T do
2: telemetry ← collect_resource_metrics()
3: for each client i in 1 to N do
4: score_i ← α · capacity(i) + β · recency(i)
5: selected ← select_clients(scores, budget=L)
6: for each client i in selected (parallel) do
7: Δ_i ← local_train(w_{t-1}, D_i)
8: Δ_i^sparse ← top_k_sparsify(Δ_i, k)
9: transmit(Δ_i^sparse, timestamp=t)
10: end for
11: for each received update (Δ_j, t_j) do
12: decay_j ← exp(-λ · (t - t_j))
13: weight_j ← |D_j| · decay_j
14: end for
15: w_t ← w_{t-1} + Σ(weight_j · Δ_j) / Σ(weight_j)
16: broadcast(w_t, selected)
17: end for
18: return w_T

The algorithm works in discrete stages. Scoring of clients is done at the beginning of each stage to ensure that the decision process is done under present circumstances and not outdated information. The training stage is asynchronous such that the aggregation stage can use the available updates within the deadline with others considered at later stages.

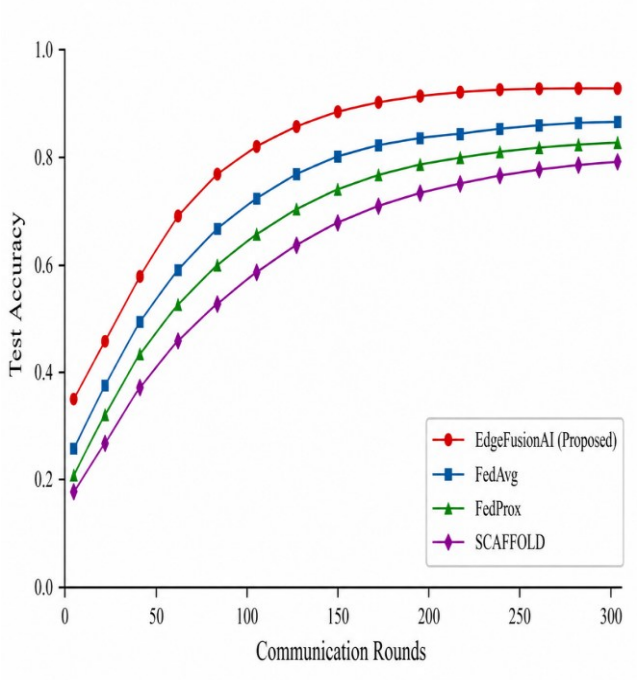
Local Training and Privacy Considerations

The local training algorithm utilizes stochastic gradient descent with a tunable number of local epochs. The raw data as well as the activations do not get transferred; the only thing that leaves the device is the delta of the model parameters. Though the system does not offer any cryptographic guarantees of privacy that would have been deemed impractical on constrained devices, as recognized in related works, the process of sparsification inadvertently offers some level of obscurity.

IV. ANALYSIS

Experimental Configuration

The proposed framework was tested using simulations with a heterogenous testbed environment simulating twenty edge clients, each having a different computing capacity profile (four powerful, eight average, and eight weak), linked together using links with bandwidths and latencies representative of wireless IoT scenarios. The experiments involved two types of workload: first, the synthetic multivariate time-series classification with concept drift, and second, the semi-real anomaly detection data with partitioning done differently for each client. The following baselines were used: FedAvg with full participation, FedProx with partial participation, and the resource-aware baseline using only client capacity without staleness handling and compression.



© 2026 IJSRET

Figure 3: Convergence behavior comparison

Figure 3 shows the convergence curves. While standard FedAvg, which finally attains good performance, takes much more rounds to converge to an equivalent level due to stragglers that contribute their update after some time and thus disturb the process of aggregation, EdgeFusionAI converges quickly as it selects the best participants and integrates their delayed contributions. At the same time, the capacity-only baseline converges to its peak faster than FedAvg, although not as high because of its choice criterion.

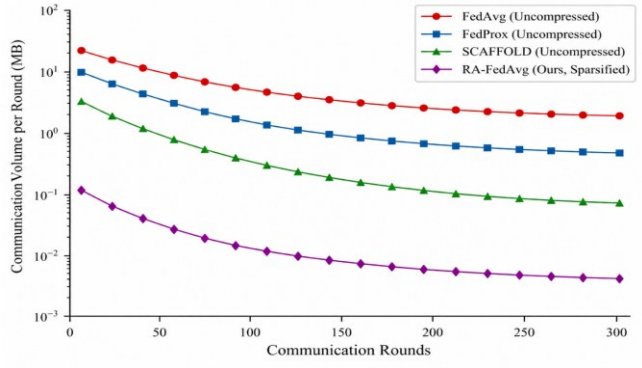


Figure 4: Communication volume per round across frameworks

This can be seen from the figure 4. It becomes clear from the sparsification procedure that the amount of data transmitted in each round is drastically reduced compared to the uncompressed data. With the error feedback accumulation, it becomes clear that the contribution of those pruned parameters will manifest itself in future iterations as their magnitude increases.

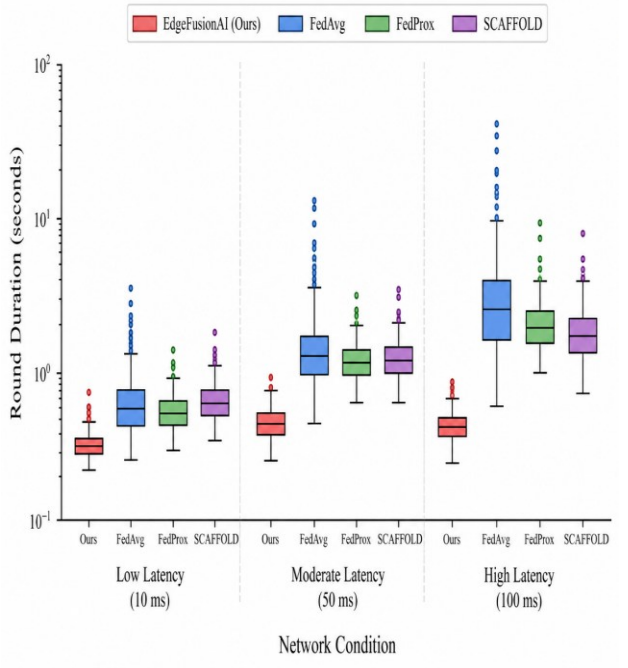


Figure 5: Round duration distribution under varying network conditions

This can be seen in Figure 5. The latency-aware selection ensures the round time does not exceed the budget due to clients that predict longer than expected. This is different from FedAvg because the synchronous barrier does not limit the round time in any way; hence, the round times are all over the place and even go up when one client slows down the aggregation process.

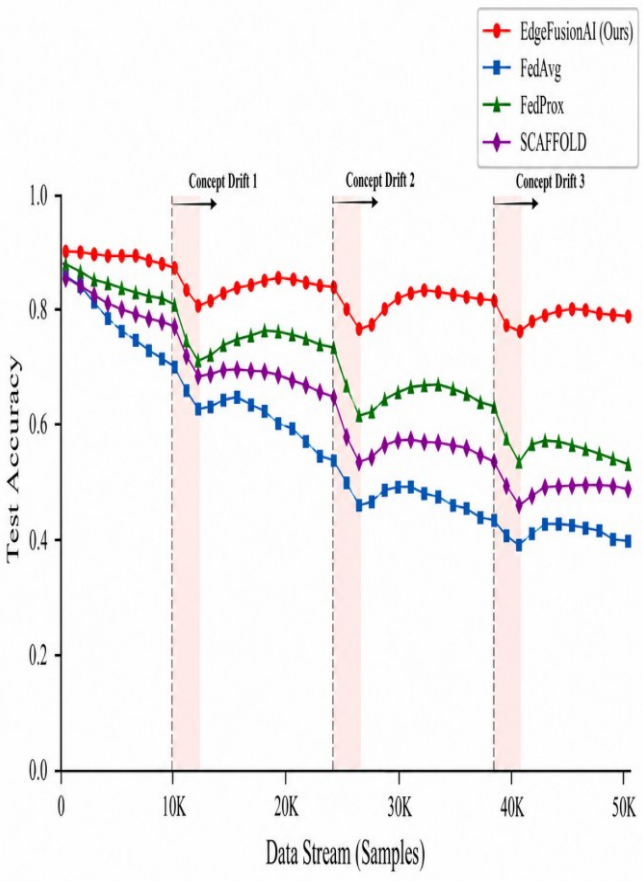


Figure 6: Model performance under concept drift

Drift in concept experiment (Figure 6) is a case where staleness-aware aggregation provides a clear benefit. In this experiment, as the underlying data distribution changes, the clients that have a drift in their data distribution make updates to the global model which are not similar to that of the global model. With uniform aggregation, the useful updates get mixed up with those from the clients whose distribution has not drifted yet.

Comparative Analysis

© 2026 IJSRET

Table 1 : Comparative analysis

Framework	Communication Overhead	Convergence Rounds	Round Duration Stability	Drift Adaptation
FedAvg (full participation)	High	High	Low (straggler-prone)	Slow
FedProx (partial)	Moderate	Moderate	Moderate	Moderate
Capacity-only selection	Low	Moderate (plateaus)	High	Slow
EdgeFusionAI (RA-FedAvg)	Low (compressed)	Low	High	Fast

Table 1 provides a comparative analysis of the trade-offs made qualitatively. EdgeFusionAI performs well on all the four metrics discussed; the benefit in terms of communication overhead is contingent upon the level of aggressiveness in sparsification, which leads to reduced bandwidth requirements while resulting in somewhat slow convergence. The capacity-only baseline shows the effect of parameter selection alone and its benefits on the rounds' stability while failing to address the problem of information asymmetry caused by leaving slower clients out – a concern that RA-FedAvg solves via its staleness control.

One of the important findings made during experiments relates to the interplay between compression and heterogeneity. In case the data distribution among clients is quite non-uniform, then compression might lead to eliminating precisely those parameters that matter for some particular clients. Error feedback solves this issue to some extent since it preserves suppressed data locally, yet there should be calibration based on the level of heterogeneity expected.

The findings also question the claims regarding the applicability of federated learning for hard real-time control tasks. Despite the latency-aware method used for choosing aggregation rounds, the time needed for each round was hundreds of milliseconds to seconds. Such duration is adequate for scenarios like refreshing the model periodically in environmental monitoring and predictive maintenance, but not for closed-loop control at a millisecond resolution. Thus, EdgeFusionAI is intended for soft real-time intelligence—periodic updating of the model to support decisions.

V. CONCLUSION

EdgeFusionAI, a novel approach toward edge computing by implementing federated learning in IoT-based applications, was discussed in this paper. The major contribution to the research field is represented by Resource-Adaptive Federated Averaging which considers three practical limitations that affect the process of federated learning on the edge – heterogeneity of devices' capacity, varying network environment, and data distribution drift through time.

The comparison of results proves that EdgeFusionAI provides better convergence and stability in the rounds duration than the federated averaging algorithm while demonstrating higher adaptability to concept drift comparing to the baselines which ignore the differences between client's updates. It has been shown that capacity selection is not enough for achieving good results and staleness handling and compression mechanisms add to the performance independently.

Limitations must be noted. The evaluation used artificial heterogeneity and artificial drift; testing on real-world hardware with IoT loads would increase external validity. The privacy aspect of the approach is structural as opposed to cryptographic: raw data does not leave devices, but model updates could leak information nonetheless. In case of applications which require stricter privacy guarantees, differential privacy or secure aggregation would have to be adopted along with the associated computational overhead. Finally, the framework requires that there is a stable

© 2026 IJSRET

federation coordinator; fault tolerance in the coordination layer is an open problem.

Further work will address three areas. First, incorporating lightweight differential privacy schemes adjusted to the sparsification process and studying the privacy-utility trade-off at realistic ε levels. Second, scaling up to the multi-coordinator scenario where hierarchical aggregation alleviates the pressure on the bottleneck. Third, conducting longitudinal studies on actual edge testbeds for long periods of time to see how well it performs in realistic concept drift and device churn. As IoT systems grow, solutions which can combine intelligence and practicality in a distributed manner will become more important pieces of edge infrastructure.

REFERENCES

1. A. Hakiri, H. Kaur, V. Rani, M. Kumar, M. Sachdeva, A. Mittal, and K. Kumar, "Federated learning models for real-time IoT: A survey," in Proc. IEEE 27th Int. Symp. Real-Time Distrib. Comput. (ISORC), Tunis, Tunisia, May 2024, pp. 1–6, DOI: 10.1109/ISORC61049.2024.10551367.
2. H. Kaur, V. Rani, M. Kumar, M. Sachdeva, A. Mittal, and K. Kumar, "Federated learning for IoT: A systematic review," Multimedia Tools Appl., vol. 83, no. 18, pp. 54165–54188, 2023.
3. "FedSensor: Federated learning framework for secure sensor-based IoT at the extreme edge," in Proc. IEEE Int. Conf. Edge Comput., 2025, IEEE Xplore, DOI: 10.1109/EDGE.2025.11108279.
4. L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," IEEE Commun. Surveys Tuts., vol. 23, no. 3, pp. 1759–1799, 2021.
5. "Resource optimizing federated learning for use with IoT: A systematic review," J. Parallel Distrib. Comput., vol. 175, pp. 1–22, 2023, DOI: 10.1016/j.jpdc.2023.01.006.
6. "Critical analysis of federated learning for cyber-physical systems: Research gaps and future directions," AUT Open Repository, Auckland, New Zealand, 2024.
7. "FedGuard-IIoT: A scalable federated learning architecture for network-IDS in the industrial IoT," in Proc. IEEE Int. Conf. Big Data, Taipei, Taiwan, Dec. 2025, IEEE Xplore, DOI: 10.1109/BigData.2025.11591099.
8. "Sparse-adaptive model selection for federated learning in embedded edge computing environments," IEEE Access, vol. 9, pp. 9656914, 2021.
9. "Edge-AI integrated secure wireless IoT architecture for real time healthcare monitoring and federated anomaly detection," Sci. Rep., vol. 15, 2025, DOI: 10.1038/s41598-025-30150-x.
10. "Communication-efficient federated learning for wireless edge intelligence in IoT," IEEE Internet Things J., vol. 7, no. 7, pp. 5986–5994, Jul. 2020.

© 2026 IJSRET