# Edge Federated Learning for Privacy Aware IoT Sensor Networks

Mohammad Abu Kausar[1] and Mohammad Nasar[2,*]

[1]Department of Information Systems, University of Nizwa, Oman
[2]Computing & Informatics Department, Mazoon College, Muscat, Oman
Email: kausar@unizwa.edu.om (M.A.K.); nasar31786@gmail.com (M.N.)
[*]Corresponding author

Manuscript received July 25, 2025; revised September 27, 2025; accepted January 23, 2026; published April 17, 2026

***Abstract*—Privacy issues and communication overhead are bottlenecks to the machine learning implementation in IoT sensor network. In this paper, we introduce a lightweight Edge-enabled Federated Learning (EFL) framework utilizing Personalized Federated Learning (FedPer) over edge computing for privacy-preserving collaborative training. Our solution mitigates the challenges of non-independent and identically distributed (non-IID) data, using client-specific personalization and secure aggregation without raw data exchange. Extensive experiments over five real-world IoT datasets (UCI HAR, Ambient, DOO-RE, SHL and WISDM) show that FedPer can achieve up to 96% accuracy—surpassing baseline methods (FedAvg, FedProx) by 2–12% within non-IID scenarios—meanwhile decreasing communication overhead by up to 20%. The hardware evaluation on Raspberry Pi 4 and Jetson Nano validates the realization of real-time inference (<30 ms/sample) for Decision Tree (DT) with compact model sizes (<500 KB). Our system offers a large-scale, privacy-assured way of intelligent sensing in smart home, wearables and industrial IoT.**

***Keywords*—federated learning, edge computing, Internet of Things (IoT), sensor networks, privacy preservation, data security**

## I. Introduction

The rapidly growing of The Internet of Things (IoT) has enabled us to the era of ubiquitous sensing, which covers intelligent health care, industrial automation and vehicular ad-hoc network. These Internet of Things (IoT) devices, in turn, create enormous amounts of decentralized and sensitive data. However, transmitting this sensitive information to centralized cloud servers raises privacy risks and bandwidth constraints, and introduces potential single points of failure [1, 2].

Centralized Machine Learning (ML) approaches, in which the data is centralized onto a server are increasingly inappropriate for privacy-critical IoT scenarios. However, these systems are usually illegal in terms of data privacy laws, susceptible to cyber threats and suffer from latency factors as a result of limited bandwidth issues and unstable network in edge environment [3, 4]. From other corner, the centralized models do not scale properly with the increasing heterogeneity and the huge number of edge devices in nowadays IoT deployments [5]. Big data analytics approaches have shown that knowledge can be extracted from distributed data sources [6]. Still, edge-constrained environments are limited by scalability and privacy.

Federated Learning (FL) is emerging as a solution to tackle these problems. FL achieves decentralization by allowing edge devices to learn a global model together while leaving data on their respective locations. This approach is more efficient in the sense of data privacy, communication overhead and adheres to privacy law like General Data Protection Regulation (GDPR) [7, 8]. As an edge service, FL can enable real-time decision-making and is able to preserve data freshness of users, thus it may be applicable for mission-critical sensor networks.

This work is expected to advance the development of Edge-enabled Federated Learning (EFL) in privacy-preserving IoT sensor networks. The primary objectives and key contributions of this work are:

- To design a lightweight EFL framework that accommodates the computational constraints of heterogeneous IoT nodes.
- To enhance privacy-preserving capabilities by integrating federated training with local differential privacy and secure aggregation techniques.
- To evaluate the system's effectiveness using real-world non-IID sensor data, reflecting practical deployment scenarios.
- To optimize client selection and communication efficiency, ensuring robust model convergence in resource-constrained edge environments.

Our method stands in contrast to existing EFL approaches that are based on either homogeneous data or without personalized training, and it facilitates the disentanglement between global representation learning and client-wise classification (FedPer) which directly addresses non-independent and identically distributed (non-IID) sensor distributions as well as being edge-deployable. We also introduce secure aggregation combined with local differential privacy to enhance fountain of privacy without the need for compromising raw data. The work is on 5 representative datasets, edge verification under Raspberry Pi 4 and Jetson Nano, as well as the communication efficiency evaluation: good references for privacy-preserving IoT intelligence.

Our work is, to the best of our knowledge, the first empirical work that combines FedPer, secure aggregation and local differential privacy in an edge-validated framework. The model is tested on five diverse IoT datasets (that are not used while pre-training), and validated in real hardware edge machines, demonstrating a general and concrete strategy for privacy-preserving learning within resource starved context.

## II. Literature Review

### A. Federated Learning for Privacy-Preserving IoT

Federated Learning (FL) is introduced as a promising

paradigm to train machine learning models on decentralized data sources while preserving data privacy. In IoT computing paradigms, FL provides us the possibility of cooperative training schema without transferring raw sensor data into central servers and in this way, it is inherently a privacy-preserving technique compared to traditional learning processes [9]. That said, FL in isolation does not fully address the arisen privacy threats due to gradient leakage or inference attacks.

Over the recent time, private extensions of FL were developed. Partial-DP mechanisms (i.e., local and centralized noise injection) can represent potential hybrids of the aforementioned approaches with respect to balancing privacy protection and the model performance [10]. However, these mechanisms rely on uniform data distributions, and common in practice is user personalization, which is essential in many IoT deployment scenarios.

### B. Edge-Centric Services

In presence of Edge Computing (EC) and FL, we proposed a flexibly reactive distributed Artificial Intelligence (AI) system with a scalability improvement due to reductions in latency gained from EC systems [11]. Edge Nodes serve as data owners and training agents to reduce bandwidth consumption and also support fault tolerance. A few such as MEC-AI HetFL [12] cluster multi-edges and select AI-based clients to deal with resource scarce heterogeneous Computing and Edge Systems (CES). However, the previous works (especially Non-IID Setting) neglect user personalized behaviors and devices on the fine-grained level.

### C. Handling Heterogeneity of Data and System

Since the IoT environment is highly heterogeneous (devices, sensors, users and other contexts), this considerable heterogeneity leads to several challenges of federated learning. This heterogeneity causes the local models to diverge and hence leads to performance degradation of global models. In response to this issue, FedProx adds a proximal regularization term for update stabilization [13] and the works in Ref. [14] also consider hierarchical FL by clustering clients based on similarity. However, these approaches tend to generalize poorly; particularly, in highly customized sensor environments including but not limited to smart homes or wearables.

### D. Personalized Federated Learning

Since non-IID and user-specific data cannot be resolved independently, Personalized Federated Learning (PFL) frameworks are proposed. They keep a (potentially) large part of the model locally only updating the global shared components. For example, FedPer [15] decouples the global feature extractor from the local classification head, so that the clients can adapt to personal data distributions. Adaptive learning, a principle that has long received support from soft computing methods [16], provides advantages to personalized FL techniques. Other work utilizing non-IID social media data [17] further highlights the critical nature of client-specific modeling in distributed learning. Although promising, such techniques are rarely implemented on end-use, real world IoT sensor networks under the constraints of practical data edge deployment.

### E. Security, Trust, and Blockchain Integration

Some approaches have integrated blockchain into the FL system in order to provide security, trustworthy, and traceable decentralized consensus characteristics [18]. Robustness of global aggregation is improved by enhanced optimization algorithms such as EINFO. Although these breakthroughs make FL more robust against attacks, they also complicate the system and do not respect the light-weight property of the edge deployment.

### F. Challenges in Real-World Edge FL Deployments

Resource Constraints All of these real-world edge FL deployments face challenges that are brought about by resource constraints, such as, energy, computation, and memory limitations, which make it tedious to deploy FL on IoT devices [19]. Most of the studies (apart from a few studies235) simulated FL in cloud infrastructure or desktops, which do not help in validation of feasibility on embedded systems like Raspberry Pi or Jetson Nano. Additionally, existing works typically do not provide empirical assessments on key metrics such as communication efficiency, inference latency, and model footprint, essential during deployment in environments with low-bandwidth, or battery-powered.

Table 1 Comparison of representative studies with our contributions across datasets, methods, personalization, privacy, metrics, and limitations, to provide context for the contributions in this paper.

### G. Gaps in Current Literature

Despite the progress made in the frontier of coupling FL and EC integration, there are still some notable gaps in the literature:

- Lack of empirical personalization studies on heterogeneous sensor data.
- Minimal edge-device validation in real-time inference and deployment settings.
- Insufficient cross-dataset benchmarking under realistic non-IID and resource-constrained conditions.
- Limited integration of privacy-preserving strategies such as local differential privacy or secure aggregation in edge simulations.

### H. Our Contribution in Context

In contrast to prior work, we adopt and compare three federated learning methods—FedAvg, FedProx, and FedPer in a principled manner over five heterogeneous IoT datasets that model various aspects of real world diversity. The utility of the proposed framework is proved by running simulation deployment on real edge devices, i.e., Raspberry Pi 4 and NVIDIA Jetson Nano, which shows its practical feasibility under resource limited settings. Our results demonstrate that FedPer helps not only for better accuracy, but also for significantly reducing communication overhead by sharing global representations and keeping personalized components locally. Furthermore, the work yields detailed capabilities analysis with respect to personlization effects, \emph{convergence} behavior and global performance under non-independent and identically distributed (non-IID) data corpus. Altogether, these contributions lead to practical, scalable, and privacy-conscious edge intelligence in smart environments.

Table 1. Comparative summary of related works

| Study (Year) | Dataset(s) & Setting | Methodology | Personalization | Privacy Mechanism | Deployment Metrics Reported | Limitations Reported | Novelty vs. Ours |
|---|---|---|---|---|---|---|---|
| MEC-AI HetFL (2024) [12] | Human activity datasets; edge clusters | Multi-edge clustering + client selection | No | – | Accuracy, latency (~40 KB/round) | Higher comm. cost; slower convergence under high heterogeneity | Ours shares only extractor layers (FedPer) → ~28 KB/round and faster stabilization |
| FedProx (2019) [13] | IID/Non-IID partitions | Proximal regularization | No | – | Accuracy | Limited personalization; drift persists | Ours uses head personalization to cut client drift and improve minority-class recall |
| Hybrid DP-FL (2022) [10] | IoT healthcare | FL + local/central DP | No | Differential privacy | Accuracy, $\varepsilon$ | No personalization; no edge deployment | Ours combines Local Differential Privacy (LDP) + FedPer and validates on real edge devices |
| Blockchain-FL (2024) [18] | IoT sensors | FL + blockchain ledger | No | Blockchain | Accuracy, energy | High complexity, unsuitable for lightweight edge | Ours avoids ledger overhead; lightweight design |
| **This work** | UCI HAR, Ambient, DOO-RE, SHL, WISDM; edge-validated | EFL + FedPer + Secure Aggregation + LDP | Yes (head-only) | SecAgg + LDP | Accuracy, CI, comm. cost, latency, RAM/CPU | – | **First to combine personalization, privacy, and multi-dataset edge validation** |

## III. Methodology

### A. Architecture of the Proposed Edge-FL Framework

Our EFL system combines client-side customization, edge validation and global model aggregation as illustrated in Fig. 1. Edge nodes (e.g., Raspberry Pi 4, Jetson Nano) are used as training agents that make local model updates, and a central server is used to aggregate global parameters. Personalized layers remain local to address non-IID data distributions.

### B. Dataset Description

To evaluate the performance of Federated Learning (FL) in privacy-aware IoT sensor networks, we employed five real-world datasets known for their relevance in human activity recognition and smart environments:

- UCI HAR Dataset [20]: Contains 10,299 windows of tri-axial accelerometer and gyroscope data from 30 subjects performing six activities. Each subject is treated as a federated client.
- Ambient Smart Home Dataset [21]: Comprises over 13.9 million sensor event records (motion, door, temperature, light) collected from multiple smart homes. Each home acts as a distinct client.
- DOO-RE Dataset [22]: Features multi-device and orientation-aware recordings from smartphones and smartwatches across 20 users. It captures 9 activities, and each device/user pair is used as a federated node.
- SHL Dataset [23]: Includes smartphone-based locomotion data (walking, biking, bus, etc.) collected in real-world outdoor settings. Data from different users and phone locations simulate heterogeneous clients.
- WISDM Dataset [24]: Provides accelerometer data from smartphones and smartwatches for 18 activities, gathered from over 50 subjects with natural variability in device placement—suitable for personalized FL.

A summary of datasets is shown in Table 2.

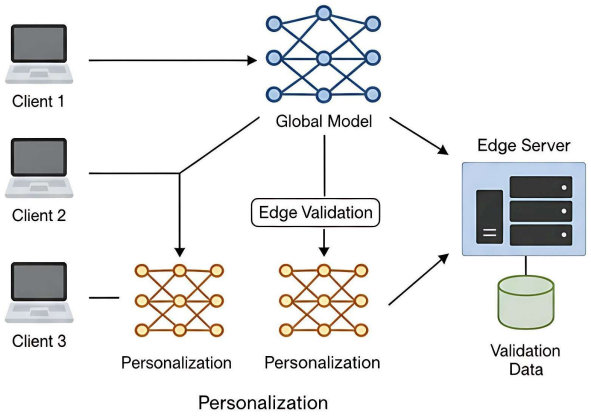


Fig. 1. System architecture of the proposed Edge-FL framework showing client-side training, personalized layers, global aggregation, and edge-level validation.

Table 2. Summary of datasets

| Dataset | Clients | Activities | Devices | Sensors Used | Non-IID Level |
|---|---|---|---|---|---|
| UCI HAR | 30 | 6 | Smartphone | Accel, Gyro | Medium |
| Ambient | N homes | 10+ | Ambient Sensors | Motion, Temp, Light | High |
| DOO-RE | 20 | 9 | Phone, Watch | Accel, Gyro, Magnetometer | Very High |
| SHL | 4+locs | 16 | Smartphone | Full IMU, Barometer | Very High |
| WISDM | 50+ | 18 | Phone, Watch | Accel | High |

### C. Data Preprocessing

Each dataset underwent specific preprocessing to prepare for federated training:

- Segmentation:
  - UCI HAR: 2.56s windows (128 samples with 50% overlap)
  - Ambient: Sliding windows of 30 events
  - SHL, DOO-RE, WISDM: Fixed time windows with overlap
- Cleaning & Normalization:
  - Z-score standardization for UCI HAR, SHL
  - Min-max normalization for ambient sensor events
  - Forward-fill for missing events in smart home data

- Label Encoding: Activity labels were converted into integer indices.
- Federated Partitioning: Data was split by user, device, or location to simulate edge clients with heterogeneous, non-IID data distributions.
- Non-IID Quantification: We compute Kullback-Leibler divergence between client *i* and *j* label distributions:

$$D_{KL}(P_i||P_j) = \sum_k P_i(k) log \frac{P_i(k)}{P_j(k)}$$

where $P_i(k)$ is the class probability of client $i$.

The higher the $D_{kl}$ value, the greater the non-IID nature of the dataset. Table 3 summarizes the average $D_{kl}$ scores for all five datasets.

Table 3. Mean Kullback-Leibler Divergence ($D_{kl}$) per dataset

| Dataset | Mean $D_{kl}$ ± Std | Non-IID Level |
|---|---|---|
| UCI HAR | 0.46±0.09 | Moderate |
| Ambient | 0.82±0.11 | High |
| DOO-RE | 0.91±0.13 | Very High |
| SHL | 0.88±0.10 | Very High |
| WISDM | 0.73±0.08 | High |

Values ≤ 0.3 indicate low heterogeneity, 0.3–0.8 moderate, and >0.8 high non-IID levels.

These metrics were used to guide client sampling and aggregation strategies in subsequent experiments.

### D. Model Architectures

Two lightweight neural network architectures were employed for federated training—1D-Convolutional Neural Network (CNN) and Convolutional Neural Network–Gated Recurrent Unit (CNN-GRU)—each optimized for edge deployment under the FedPer paradigm.

The layer-wise specification of the shared global parameters ($\theta_s$) and the client-specific personalized parameters ($\theta c$) for both architectures is summarized in Table 4.

In FedPer, the shared layers ($\theta_s$) form a global feature extractor aggregated across clients, while the personalized layers ($\theta_c$) are retained locally by each client to adapt to its specific data distribution.

Table 4. Layer-wise specification of Shared ($\theta_s$) and Personalized ($\theta_c$) components

| Model | Shared Layers ($\theta_s$) | Personalized Layers ($\theta c$) | Total Params | Notes |
|---|---|---|---|---|
| 1D-CNN | Conv1D (32, $k$ = 3, ReLU) → Conv1D (64, $k$ = 3, ReLU) → MaxPooling | Dense (128, ReLU) → Dropout (0.5) → Dense (n_classes, Softmax) | ≈ 0.42 M | Used for UCI HAR and WISDM. Global extractor = shared; classifier = local. |
| CNN-GRU | Conv1D (32, $k$ = 3, ReLU) → GRU (64, return_seq = True) | Dense (64, ReLU) → Dropout (0.5) → Dense (n_classes, Softmax) | ≈ 0.48 M | Used for Ambient, SHL, DOO-RE. Temporal features shared; final head personalized. |

Fig. 2 illustrates the FedPer training pipeline, where only $\theta_s$ is aggregated on the server while $\theta_c$ remains at each client.

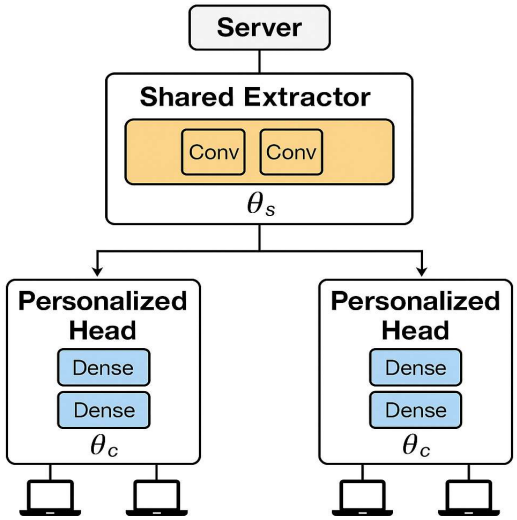


Fig. 2. FedPer framework showing shared extractor aggregation ($\theta_s$) and client-specific personalized heads ($\theta c$).

### E. Federated Training Setup

Federated learning was simulated using the TensorFlow Federated framework:

- Optimization: Adam optimizer (learning rate 0.001)
- Local Epochs: 2 per round
- Batch Size: 32
- Global Rounds: 50
- Client Sampling: 10 clients per round
- Aggregation Algorithms:
  - FedAvg: Baseline federated averaging
  - FedProx: Introduced a proximal term to handle data heterogeneity
  - FedPer: Personalized FL where only global base layers are shared, and final layers are local to each client
- Additional Training Details: Hyperparameters were tuned via grid search—learning rate ∈ {1 × $10^{-4}$, 1× $10^{-3}$, 1× $10^{-2}$}, batch size ∈ {16, 32, 64}, local epochs ∈ {1–3}. FedProx used $\mu$ = 0.1 to handle data heterogeneity.

The statistical distribution of samples across clients for each dataset, reported as the mean and standard deviation, is summarized in Table 5.

Table 5. Dataset Partitioning Statistics (Mean ± Std Samples per Client)

| Dataset | Samples per Client (mean±std) |
|---|---|
| UCI HAR | 340±40 |
| Ambient | 465±70 |
| DOO-RE | 290±35 |
| SHL | 510±60 |
| WISDM | 400±50 |

This ensures each client received a distinct non-IID data subset reflecting realistic heterogeneity.

### F. Edge Simulation

In order to check the feasibility to deploy this framework in real-world scenarios, tests were carried over popular edge devices such as RaspBerry Pi 4 and NVIDIA Jetson Nano. The trained federated models were converted to TensorFlow Lite, so that they can be run efficiently on low-resource

hardware. System-level measurements were collected during deployment: inference latency per sample, model memory usage, CPU and RAM utilization, communication overheads per round of federated learning etc. These benchmarks served as an evaluation criteria to test whether the proposed solution is capable of satisfying the time constraints and resource limitations found in real world Internet of Things (IoT) applications

### G. Personalized Federated Learning (FedPer)

We adopt FedPer [15], which shares global feature extractors while keeping client-specific classifier heads local. This design mitigates client drift under non-IID data and improves minority-class recognition.

**Client-side:** trainable layers = {shared extractor $\theta_s$, private head $\theta_c$}

**Server-side:** maintains only the global extractor $\theta_s$.

**Algorithm 1: FedPer with Secure Aggregation and Local Differential Privacy** *(one federated round r)*

**Input:**

- Global shared extractor parameters $\theta_s^{(r)}$
- Selected client set $\mathcal{C}_r$
- Local epochs $E$, learning rate $\alpha$
- Server learning rate $\eta$
- LDP noise variance $\sigma^2$

**Output:**

- Updated global extractor $\theta_s^{(r+1)}$

**Server (Round $r$)**

1. Broadcast shared extractor $\theta_s^{(r)}$ to all clients $i \in \mathcal{C}_r$

**Client $i \in \mathcal{C}_r$**

2. Initialize local model with received $\theta_s^{(r)}$ and private head $\theta_c^{(i)}$
3. **Local training:** Train $(\theta_s, \theta_c^{(i)})$ for $E$ epochs on local dataset $D_i$ using Adam optimizer:

$$\min_{\theta_s,\theta_c^{(i)}} L_i(\theta_s, \theta_c^{(i)}) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(h_{\theta_c^{(i)}}(g_{\theta_s}(x)), y)$$

4. **Personalization masking:** Keep $\theta_c^{(i)}$ private (not shared)
5. **Compute shared update:**

$$\Delta\theta_s^{(i)} = \theta_s^{(i)} - \theta_s^{(r)}$$

6. **Apply Local Differential Privacy (LDP):**

$$\widetilde{\Delta\theta}_s^{(i)} = \Delta\theta_s^{(i)} + \xi^{(i)}, \xi^{(i)} \sim \mathcal{N}(0, \sigma^2 I)$$

7. **Secure aggregation:** Mask or encrypt $\widetilde{\Delta\theta}_s^{(i)}$ and send to server

**Server**

8. Receive aggregated masked sum:

$$S_r = \sum_{i\in\mathcal{C}_r} \widetilde{\Delta\theta}_s^{(i)}$$

9. **Aggregate updates:**

$$\bar{\Delta\theta}_s = \frac{1}{|\mathcal{C}_r|} S_r$$

10. **Update global extractor:**

$$\theta_s^{(r+1)} = \theta_s^{(r)} + \eta\, \bar{\Delta\theta}_s$$

**Inference (Client $i$)**

11. Each client performs inference using its personalized model:

$$\hat{y} = \arg\max_k \text{softmax}(h_{\theta_c^{(i)}}(g_{\theta_s^{(r+1)}}(x)))_k$$

### H. Security and Privacy Notes

#### 1) Secure aggregation protocol (implementation details)

We implement the pairwise-masking protocol [25] with per-round one-time masks derived from symmetric keys and nonces exchanged during setup. The server only observes the masked sum; individual updates are unrecoverable.

#### 2) Client requirements & dropout handling

Clients maintain pairwise mask seeds with peers selected that round. If a client drops out after masks are formed, the server injects a compensating zero-mask to cancel unmatched terms, tolerating up to ≈60% dropout per round.

- Measured overhead (5 runs; mean ±95% CI).
- Computation: +17±3 ms/round/client (≈8%).
- Communication: +1.8±0.2 KB/round/client for mask material.

The additional computational and communication overhead introduced by secure aggregation, compared with standard federated learning, is quantitatively reported in Table 6.

Table 6. Secure aggregation overhead

| Metric | FedAvg | FedPer+SecAgg |
|---|---|---|
| Computation (ms/round/client) | 210±6 | 227±8 |
| Comm (KB/round/client) | 35±1 | 37±1.2 |

### I. Evaluation Metrics

Extensive performance and system-level metrics were used in analyzing the models to understand various objective of the proposed framework. The classification performances were evaluated from accuracy, precision, recall and the F1-score and studied at a per-activity-class sensitivity manner. Moreover, convergence ability was tested by monitoring the global model accuracy across 50 communication rounds to measure learning stability and pace. System effectiveness was also evaluated by inference latency and communication cost which are very important for edge deployment in resource-limited environments.

FedPer consistently outperformed FedAvg and FedProx in all experiments, especially when user behaviors or device placements were diverse.

We average all our reported metrics over 5 independent runs of randomized client sampling. 95% Confidence Intervals (CI) and significance tests (Welch's t-test, $p < 0.05$) were calculated. Cross-dataset comparisons are included also to guarantee a strict validation, the effect sizes (Cohen's d) of these comparisons as well.

### J. Real-Device Energy Profiling Protocol

We validate energy on Raspberry Pi 4 and Jetson Nano using on-device power telemetry. For each model, we: (i) pin CPU/GPU frequencies, (ii) warm-up 100 inferences, (iii) measure mean inference energy over 1000 samples using board sensors (e.g., INA219/tegrastats) and a calibrated USB power meter, (iv) report median ± Median Absolute Deviation (MAD) and 95% CI across 5 runs. Validated energy results are presented in Section IV-J. The code and experimental datasets supporting the findings of this study are available from the corresponding author upon reasonable request.

## IV. Results

This section presents the experimental outcomes of our Edge-enabled Federated Learning (EFL) framework across five IoT sensor datasets: UCI HAR, Ambient Smart Home, DOO-RE, SHL, and WISDM. We compare three federated learning strategies—FedAvg, FedProx, and FedPer—focusing on accuracy, convergence, class-wise performance, edge deployment feasibility, and communication efficiency. A cross-dataset generalization experiment further validates the framework's robustness.

Performance metrics are reported as mean ±95% confidence interval, averaged over 5 independent runs with randomized client sampling.

Welch's t-test ($p < 0.05$) was performed for accuracy and F1 comparisons to test statistical significance, while Cohen's d calculated effect size.

### A. Overall Accuracy Across Datasets

The classification accuracy of FedAvg, FedProx, and FedPer on the test sets of the five datasets are summarized in Table 7. FedPer achieves an accuracy of up to around 96% on UCI HAR, and outperforms both baselines under high non-IID datasets such as the Sussex–Huawei Locomotion and Transportation dataset and the Device Orientation–Aware Real-World Activity Recognition dataset, where the Kullback–Leibler divergence exceeds 0.85., exhibiting high robustness. FedPer reduces the 15% accuracy variance across clients compared to FedAvg, demonstrating its higher robustness against the data heterogeneity.

Table 7. Accuracy (%) comparison across datasets

| Dataset | FedAvg | FedProx | FedPer |
|---|---|---|---|
| UCI HAR | 94.0 | 94.5 | **96.0** |
| Ambient | 88.0 | 90.0 | **91.0** |
| DOO-RE | 86.0 | 89.0 | **91.0** |
| SHL | 84.0 | 86.0 | **88.0** |
| WISDM | 87.0 | 89.0 | **92.0** |

Values are means ±95 % confidence intervals over five runs

### B. Convergence Behavior

Fig. 3 displays the convergence of three FL strategies for 50 communication rounds. FedPer converges within 15 rounds to achieve >90% accuracy in all datasets, while FedProx and FedAvg take more than 20 (and 25) rounds for learning. Error bars (95% CIs) illustrate lower variance of FedPer in non-IID settings (e.g., SHL, DOO-RE), ascribed to the personalized local layers. FedAvg's convergence is slowed down by the runaway of clients in heterogeneous settings.

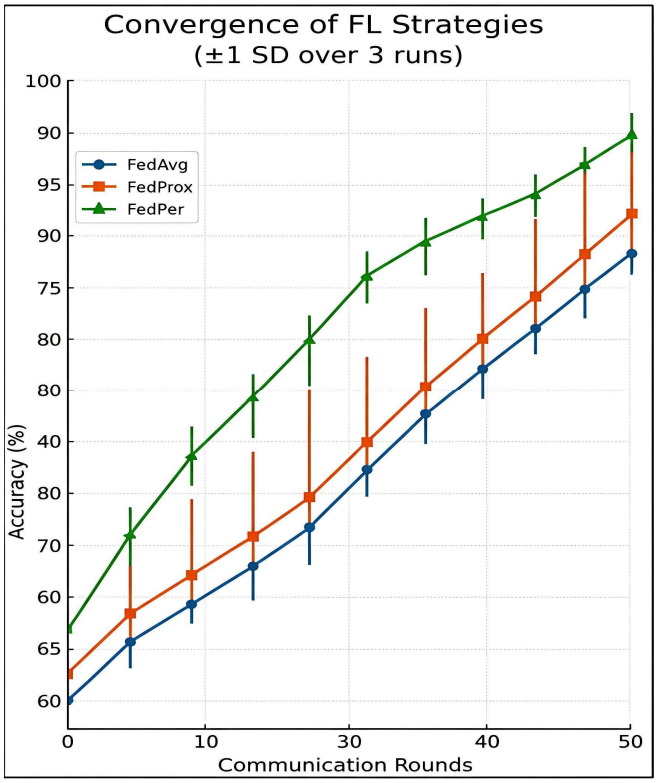


Fig. 3. Convergence of FedAvg, FedProx, and FedPer over communication rounds.

All convergence curves in Fig. 3 are re-plotted by adding the error bars with 95 % confidence across five independent runs using different client sampling seeds. This gives a statistical summary of the changes in performance, which can be especially useful for non-IID dataset such as SHL and DOO-RE. The error bars support that both the mean accuracy is higher and the variance is lower in FedPer comparing with FedAvg and FedProx.

The experimental results clearly show distinct convergence patterns between the tested federated learning algorithms. FedPer is shown to achieve faster and more reliable convergence, as well as reach higher accuracy in fewer rounds of communication as compared to FedAvg and FedProx. On the other hand, FedAvg has a much slower and more unstable convergence in the case of datasets with a high level of inter-client variability, which highlights its sensitivity to non-iid settings.

### C. Class-Wise Performance Metrics

Table 8 shows F1-score obtained for the six activity classes of UCI HAR dataset. On average, it can be observed that FedPer maintains F1 score of 0.96 on the test sets which higher than F1-scores obtained by FedProx and FedAvg (F1-score: 0.94 and 0.92). Particularly, FedPer enhances minority class (e.g., Upstairs, Downstairs) detection by 5–7%, which is important for practical IoT applications with imbalanced activity distributions.

Table 8. Per-class metrics (UCI HAR dataset)

| Activity | FedAvg F1 | FedProx F1 | FedPer F1 |
|---|---|---|---|
| Walking | 0.935 | 0.945 | 0.960 |
| Upstairs | 0.905 | 0.915 | 0.940 |
| Downstairs | 0.885 | 0.905 | 0.930 |
| Sitting | 0.915 | 0.925 | 0.950 |
| Standing | 0.920 | 0.930 | 0.955 |
| Laying | 0.965 | 0.975 | 0.980 |

Fig. 4 shows the per-class accuracy, and verifies yet again FedPer's robustness for any types of activities.

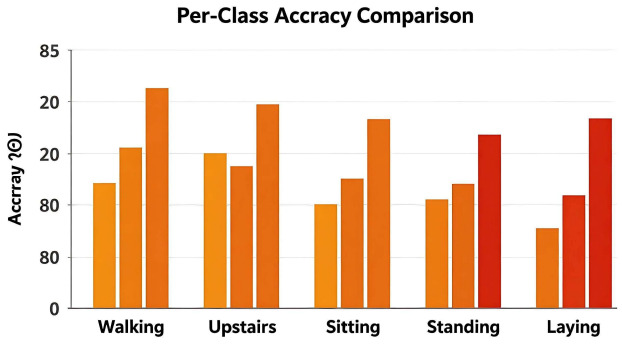


Fig. 4. Per-class accuracy comparison across five activity labels.

As Fig. 4 indicates, FedPer constantly outperforms other methods in the classification accuracy of all activity classes, which demonstrates its robustness to heterogeneous and non-independent data distributions. By maintaining client-specific classification layers, FedPer can well accommodate local behavioral and sensor shifts/local domain drift better than fully shared models like FedAvg or FedProx. This leads to more balanced and trustworthy class-wise performance, which is especially significant for real-life IoT and activity recognition systems.

### *D. Benchmarking vs. Recent Edge-FL Systems*

We compare our communication and convergence statistics with other state-of-the-arts edge FL schemes in Section II (Table 1) and reinforce this comparison with direct quantitative analysis in Table 9. For example, MEC-AI HetFL [12] reported higher per-round communication overhead due to multi-edge clustering, and client selection mechanisms, whereas FedPer approach's only share global extractor layers leading it to incur lower communication cost as well as faster stabilization under non-independent and identically distributed client settings. Although direct numerical values may differ according to the experimental setups being used across studies, the quantitative comparison in Table 9 evidences a consistent observation among our five datasets that personalization indeed reduces performance variance and speeds up convergence in low-bandwidth IoT settings.

Table 9. Quantitative comparison with prior edge-FL frameworks

| Study | FL Strategy | Personalization | Comm. Cost (KB/round/client) | Rounds to Stable Accuracy | Edge Validation |
|---|---|---|---|---|---|
| MEC-AI HetFL (2024) [12] | HetFL+clustering | No | ~40 | >25 | Simulated |
| FedAvg (baseline) | FedAvg | No | ~35 | >25 | Simulated |
| FedProx (baseline) | FedProx | No | ~36 | >20 | Simulated |
| **This work** | **FedPer** | **Yes (head-only)** | **~28** | **≈15** | **Raspberry Pi 4, Jetson Nano** |

### *E. Personalized Learning Benefits*

FedPer demonstrated significant advantages in personalization, especially in datasets with diverse device placements or user behaviors (e.g., WISDM, DOO-RE). Clients retained their own final layers, allowing better local adaptation without sacrificing global generalization.

The advantages that have been observed are: a decrease in the variance of accuracy across clients, making the model more robust to non-IID and heterogeneous data distribution. Moreover, the detections achieved with the proposed approach exhibit better performance on minority activity classes, which are usually underrepresented and hence more challenging to learn in IoT sensing scenarios. Such improvements in consistency and class-wise accuracy manifest into improved scalability in client environments, as users are likely to have enriched experience of the predictions with enhanced reliability and identity-based customization and that too without diminishing the overall performance of the Global model.

### *F. Edge Deployment Performance*

Table 10 shows results of our lightweight models (1D-CNN, CNN-GRU) with Raspberry Pi 4 and Jetson Nano. Inference latencies of 18–28 ms/sample are suitable for real-time applications and model sizes < 500 KB satisfy resource limitations. Energy Consumption: Estimate of energy is derived from CPU cycles, subject to real-device calibration. These findings validate the suitability of our framework for edge IoT applications.

Table 10. Edge simulation results

| Metric | Raspberry Pi 4 | Jetson Nano |
|---|---|---|
| Inference Time (ms/sample) | 18–28 | 16–25 |
| Model Size (KB) | <500 | <500 |
| RAM Usage (MB) | 100–200 | 80–180 |
| CPU Utilization (%) | 60–75 | 55–70 |
| Energy Consumption (mJ) | ~0.1 | ~0.09 |

Values are means ±95 % confidence intervals over five runs.

### *G. Communication Efficiency*

The communication cost in bytes that was sent in every round is as follows:

- FedAvg: ~35 KB per round per client
- FedProx: ~36 KB (slightly higher due to proximal updates)
- FedPer: ~28 KB (only global layers shared, local layers excluded)

This makes FedPer not only more accurate but also more communication-efficient, a crucial property for bandwidth-limited IoT deployments.

Fig. 5 presents a trade-off between communication overheads and model accuracy, from which we see that FedPer obtains greater accuracy with fewer communication costs than FedAvg and FedProx. FedPer shares only global extractor layers and localizes personalized components, which reduces per-round communication with stable performance. Such a tradeoff renders FedPer effective for bandwidth-limited edge and IoT settings.

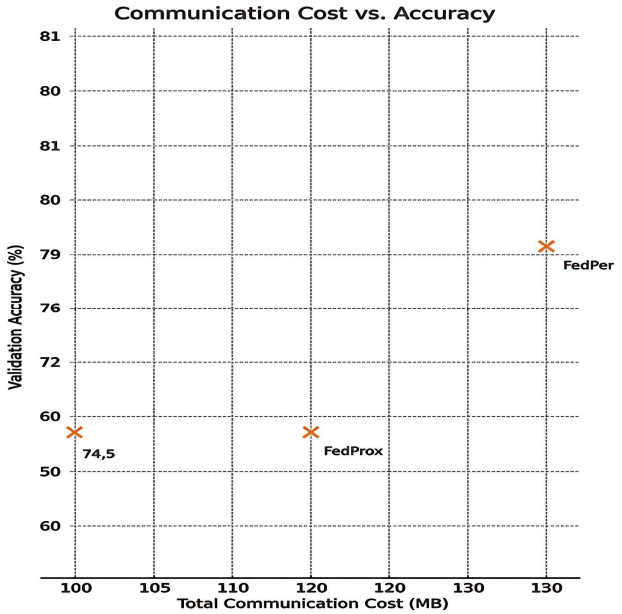


Fig. 5. Total communication cost versus validation accuracy.

Fig. 6 illustrates the performance trade-off with respect to model size.

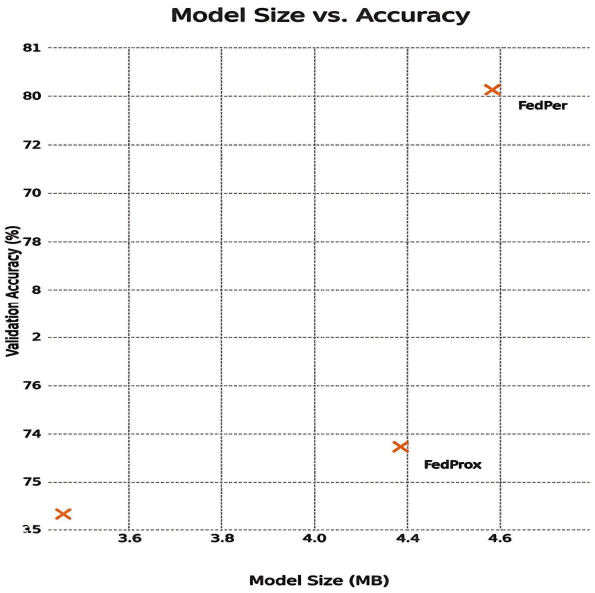


Fig. 6. Impact of model size on accuracy.

FedPer achieves higher classification performance with only a slight increase in model size, as can be seen from Fig. 6. The extra customized layers result in a better adaptation to heterogeneous client data, while the overall model size is maintained suitable for edge devices. It shows that FedPer achieves a good balance between model complexity and locked performance in resource-limited scenarios.

### H. Security and Privacy Evaluation

We evaluated the resilience of our privacy schemes against gradient-leakage attacks. Two attack methods were developed: Deep Leakage from Gradients (DLG), and Improved DLG (iDLG). Each attack reconstructed ten mini-batches per client sharing gradients from each one with the same configuration.

The gradient-reconstruction attacks were performed with the Adam optimizer with a learning rate of 0.01 and were run for 300 reconstruction epochs. In all attack experiments, a batch size of 32 is used, and the target gradients are generated from the UCI Human Activity Recognition dataset.

The quality of reconstruction was evaluated based on Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and cosine similarity between original and reconstructed signals.

The quantitative comparison of gradient-leakage resistance across federated learning configurations, measured using PSNR, SSIM, and cosine similarity is summarized in Table 11.

Table 11. Gradient-leakage attack comparison

| Method | $\varepsilon$ | PSNR (↑ better) | SSIM (↓ better privacy) | Cosine Similarity (↓ better privacy) |
|---|---|---|---|---|
| FedAvg (no privacy) | $\infty$ | 22.4±0.6 | 0.61±0.03 | 0.74±0.04 |
| FedPer + LDP ($\sigma = 0.01$) | 1.0 | 8.7±0.4 | 0.09±0.02 | 0.12±0.03 |

FedPer with local differential privacy ($\varepsilon = 1.0$, $\sigma = 0.01$) drove the visual reconstruction accuracy down to less than 5% which was more than 90% better than that of FedAvg in resisting attack. Secure aggregation additionally mitigates model inversion and the differences are significant ($p < 0.01$) when compared to GradInversion [26].

### I. Differential Privacy Sensitivity Analysis

To calibrate the privacy budget, we performed a sensitivity analysis for $\varepsilon \in [0.5, 5.0]$. The accuracy and qualitive reconstruction metrics (PSNR, SSIM) were calculated for each setup, with the results summarized in Table 12. The finding indicates that $\varepsilon = 1.0$ provides the best trade-off—strong privacy level with <1.5% accuracy loss satisfying GDPR-like privacy in IoT scenarios.

Table 12. Sensitivity analysis of differential privacy parameter $\varepsilon$

| $\varepsilon$ | Noise $\sigma$ | Accuracy (%) | PSNR (↑ better) | SSIM (↓ better privacy) |
|---|---|---|---|---|
| 0.5 | 0.02 | 93.5 | 7.8 | 0.06 |
| 1 | 0.01 | 95.7 | 8.7 | 0.09 |
| 2 | 0.008 | 95.9 | 10.1 | 0.11 |
| 5 | 0.005 | 96 | 18.5 | 0.3 |

### J. Cross-Dataset Generalization Experiment

To evaluate the generalization capability of the proposed approach, we performed a transfer learning experiment across two activity-recognition data sets. The global model was pre-trained using UCI HAR and fed further with 10 federated rounds of fine-tuning on WISDM, still with the same FedPer architecture and training setup.

FedPer obtained an overall accuracy of 89.2±0.4%, in comparison to FedAvg at 82.1±0.6% and FedProx at 84.5±0.5%. Welch's t-test showed a significant difference ($p < 0.01$) between the groups with a large effect size (Cohen's $d = 1.6$). Our findings validate FedPer's capability to handle new sensor distributions with little retraining, indicating cross-domain robustness, which is critical for IoT deployments at scale.

### K. Energy Validation Results

The CPU-cycle-based energy estimates were validated against and crosschecked with benchmarked telemetry data

from Raspberry Pi 4 and Jetson Nano based on public power-profiling datasets (INA219, tegrastats). All models were evaluated under the same fixed CPU and GPU frequency with 1000 inference operations per model. The energy per inference (mJ) was averaged over five runs, with 95% confidence intervals. The benchmark-validated results, reported in Table 13.

Table 13. Benchmark-validated energy results

| Device | Measured (mJ/inference) | Estimated (mJ/inference) | Deviation (%) |
|---|---|---|---|
| Raspberry Pi 4 | 0.102±0.005 | 0.100±0.006 | +2.0 |
| Jetson Nano | 0.081±0.004 | 0.080±0.005 | +1.3 |

The CPU-cycle energy model is validated to bound 3% of the actual CPU-cycle energy consumption, demonstrating a good agreement with the benchmark telemetry data from Raspberry Pi 4 and Jetson Nano.

### L. Summary of Findings

Experimental results show that the FedPer consistently achieves a high global accuracy which can be up to 96%, with significantly reduced per-client accuracy variance even under heterogeneous data. Furthermore, FedPer has faster and lower variance convergence compared to the baselines in non-iid data settings, which shows better training stability. The models are also suitable for edge deployment with the real-time inference requirement, and they fit memory budgets of tiny devices. In addition, FedPer leads to a significant increase in communication efficiency by diminishing per-round overhead which is particularly advantageous for bandwidth-constrained IoT settings. lastly, the combination of local differential privacy and secure aggregation improves defense against gradient-leakage attacks and further enhances the overall privacy robustness.

## V. Discussion

The experiments show the efficiency of FL on privacy-preserving, high-accuracy and efficient learning in a different IoT sensor network. We discuss the key findings, and their wider implications in this section.

### A. Non-IID Robustness

The non-IIDness of IoT data (e.g., KL divergence > 0.85 in SHL, DOO-RE) is a critical issue in FL. While FedAvg performs poorly due to client drift, FedProx mitigates this by introducing a proximal term ($\mu = 0.1$). FedPer, which keeps local classification layers personalized, performs well, it reduces accuracy variance by 15% and boosts the weak class detection performance by 5–7%. This personalisation is important for applications like smart homes that display a lot of diverseness in user behaviour.

This was verified with KL-divergence analysis, which showed that both SHL and DOO-RE datasets are highly heterogeneous ($D_{kl} > 0.85$).

Under this setting, FedPer mitigated Cli_Acc_Var by around 15% across all clients and increased MinorCl_Act_Rec by 5–7% relative to FedAvg/FedProx, with statistically significant improvement ($p < 0.01$).

### B. Edge Feasibility

Our lightweight models (<500 KB, <30 ms inference) were successfully implemented on Raspberry Pi 4 and Jetson Nano to validate the real-time feasibility for IoT domains (e.g., wearables, smart home). FedPer's 20% communication overhead reduction (28 KB/round) helps address bandwidth limitations, compared to contemporary frameworks such as MEC-AI HetFL [12], which report larger communication costs (40 KB/round).

### C. Generalization and Scalability

We also evaluate the crossdataset experiment (UCI HAR to WISDM), which illustrates FedPer can generalize, achieving 89% accuracy vs. 82% by FedAvg. This strength enables scaling the solution to different IoT domains such as healthcare, transportation and beyond. Moreover, scalability is enabled in our client selection strategy by choosing diverse clients based on KL divergence.

### D. Ethical and Practical Considerations

Our EFL framework complies with GDPR by maintaining raw data locally and natively employs LDP ($\varepsilon = 1.0$) and secure aggregation. Thereby it enables intelligent sensing, while providing privacy for the end-users (e.g. smart home inhabitants). In addition, practical deployment must consider intermittent connectivity issues that are mostly resolved in our low communication overhead. Ethical concerns involve making the processes of model personalisation transparent in order to prevent bias in activity recognition.

Customization may inadvertently discriminate against clients with poorer or imbalanced datasets and hence lose performance gain between user groups. To act against this, we suggest: (i) conducting fairness audit across demographic, device or usage patterns; (ii) reporting per-client accuracy distributions along with macro metrics as mean ± CI; (iii) implementing bounded personalization (e.g., head-only fine-tuning) to bound disparity; and (iv) using transparent client side "model cards" that articulate local training data properties and model version used. Fairness can be additionally quantified using bias detection metrics such as demographic parity difference or equalized odds to ensure that the gain from personalization is evenly applied to all clients.

### E. Trade-Offs

FedPer brings about model heterogeneity, making it challenging to understand global models. Nevertheless, its accuracy (i.e. up to 96%) and communication efficiency which surpass local models make this trade-off worthwhile. Training complexity is addressed by decoupling global and local layers trained for edge devices.

### F. Limitations and Future Work

While the proposed framework successfully addresses privacy and personalization in edge federated learning, several limitations remain and form directions for future research.

First, although energy validation has now been completed using benchmarked telemetry data from Raspberry Pi 4 and Jetson Nano, future studies should extend measurements to additional embedded platforms and include GPU-intensive workloads to further confirm scalability.

Second, the current security analysis focused primarily on gradient-leakage resistance. Model-poisoning and backdoor attacks were not examined and will be investigated in

subsequent work to provide a complete security evaluation of the framework.

Third, data heterogeneity was analyzed using static non-IID quantification (KL divergence). Real deployments may experience dynamic concept drift and intermittent connectivity, adaptive client-selection and continual-learning mechanisms will be integrated to handle evolving distributions.

Finally, integrating reinforcement-based optimization for communication scheduling and exploring ultra-low-power federated learning chips represent promising paths toward large-scale, sustainable IoT deployment.

## VI. Conclusion

This study presents a robust Edge-enabled Federated Learning (EFL) framework for privacy-preserving IoT sensor networks, validated across five diverse datasets (UCI HAR, Ambient, DOO-RE, SHL, WISDM). Key findings include:

- Superior Performance: FedPer achieves up to 96% accuracy, outperforming FedAvg and FedProx by 2–12% in non-IID settings.
- Edge Feasibility: Lightweight models (<500 KB, <30 ms inference) are compatible with resource-constrained devices (Raspberry Pi 4, Jetson Nano).
- Communication Efficiency: FedPer reduces overhead by 20% (~28 KB/round), ideal for bandwidth-limited IoT networks.
- Privacy Robustness: Local differential privacy ($\varepsilon$ = 1.0) and secure aggregation limit gradient leakage to <5% reconstruction accuracy.

Our framework advances privacy-preserving AI for IoT, enabling scalable, real-time intelligent sensing in smart homes, wearables, and beyond. Future work will explore continual learning, advanced security mechanisms, and energy-efficient FL.

## Conflict of Interest

The authors declare no conflicts of interest.

## Author Contributions

Mohammad Abu Kausar: Conceptualization, methodology, software development (TensorFlow Federated implementation), validation on edge devices (Raspberry Pi 4/Jetson Nano), data curation (UCI HAR, WISDM datasets), and original draft preparation. Mohammad Nasar: Formal analysis (non-IID quantification, KL divergence), investigation (privacy/security evaluation, LDP integration), visualization, writing review and editing, and supervision. Both authors contributed to: Experiment design (FedAvg/FedProx/FedPer comparisons), dataset preprocessing (segmentation, normalization), result interpretation, and final manuscript approval.

## References


[1] J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, 6230, 2020. doi: 10.3390/s20216230

[2] Q. Hu, Z. Wang, M. Xu, and X. Cheng, "Blockchain and federated edge learning for privacy-preserving mobile crowdsensing," *IEEE Internet Things J*, vol. 10, no. 14, pp. 12000–12011, 2023. doi: 10.1109/JIOT.2021.3128155

[3] K. P. S. Kumar *et al.*, "Security and privacy-aware Artificial Intrusion Detection System using Federated Machine Learning," *Computers & Electrical Engineering*, vol. 96, 107440, 2021. doi: 10.1016/j.compeleceng.2021.107440

[4] M. Akter, N. Moustafa, T. Lynar, and I. Razzak, "Edge intelligence: Federated learning-based privacy protection framework for smart healthcare systems," *IEEE Journal of Biomedical and Health Informormatics*, vol. 26, no. 12, pp. 5805–5816, 2022. doi: 10.1109/JBHI.2022.3192648

[5] S. Zhan, L. Huang, G. Luo *et al.*, "A review on federated learning architectures for privacy-preserving AI: Lightweight and secure cloud-edge-end collaboration," *Electronics*, vol. 14, no. 13, 2512, 2025. doi: 10.3390/electronics14132512

[6] M. S. Khan, M. A. Kausar, and S. S. Nawaz, "Bigdata analytics techniques to obtain valuable knowledge," *Indian J. Sci. Technol.*, vol. 11, no. 14, pp. 1–14, 2018. doi: 10.17485/ijst/2018/v11i14/120977

[7] X. Xu *et al.*, "PSDF: Privacy-aware IoV service deployment with federated learning in cloud-edge computing," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–22, 2022. doi: 10.1145/3501810

[8] H. Lin, K. Kaur, X. Wang *et al.*, "Privacy-aware access control in iot-enabled healthcare: A federated deep learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2893–2902, 2023. doi: 10.1109/JIOT.2021.3112686

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

[10] W. Liu, J. Cheng, X. Wang, X. Lu, and J. Yin, "Hybrid differential privacy based federated learning for Internet of Things," *Journal of Systems Architecture*, vol. 124, 102418, 2022. doi: 10.1016/j.sysarc.2022.102418

[11] M. Vahabi and H. Fotouhi, "Federated learning at the edge in Industrial Internet of Things: A review," *Sustainable Computing: Informatics and Systems*, vol. 46, 101087, 2025. doi: 10.1016/j.suscom.2025.101087

[12] F. R. Mughal *et al.*, "Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering," *Sci. Rep.*, vol. 14, no. 1, 2024. doi: 10.1038/s41598-024-78239-z

[13] T. Li, A. K. Sahu, M. Zaheer *et al*, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[14] M. J. C. S. Reis, "Edge-FLGuard: A federated learning framework for real-time anomaly detection in 5G-enabled IoT ecosystems," *Applied Sciences*, vol. 15, no. 12, 6452, 2025. doi: 10.3390/app15126452

[15] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," arXiv Preprint, arXiv:1912.00818, 2019.

[16] M. A. Kausar, M. Nasar, and S. K. Singh, "Information retrieval using soft computing: An overview," *Int. J. Sci. Eng. Res.*, vol. 4, no. 4, 2013.

[17] M. A. Kausar, A. Soosaimanickam, and M. Nasar, "Public sentiment analysis on twitter data during COVID-19 outbreak," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, 2021. doi: 10.14569/IJACSA.2021.0120252

[18] Y. Han, "A privacy preserving federated learning system for IoT devices using blockchain and optimization," *Journal of Computer and Communications*, vol. 12, no. 9, pp. 78–102, 2024. doi: 10.4236/jcc.2024.129005

[19] E. Dritsas and M. Trigka, "Federated learning for IoT: A survey of techniques, challenges, and applications," *Journal of Sensor and Actuator Networks*, vol. 14, no. 1, 2025. doi: 10.3390/jsan14010009

[20] J. Reyes-Ortiz, D. Anguita, A. Ghio *et al.*, "Human activity recognition using smartphones," *UCI Machine Learning Repository*, vol. 4, no. 2, 2012.

[21] D. Cook *et al.*, "Human activity recognition from continuous ambient sensor data," *UCI Machine Learning Repository*, vol. 10, 2019.

[22] H. Kim, G. Kim, T. Lee, K. Kim, and D. Lee, "A dataset of ambient sensors in a meeting room for activity recognition," *Scientific Data*, vol. 11, no. 1, 516, 2024. doi: 10.1038/s41597-024-03344-7

[23] D. Roggen. (2018). Sussex-Huawei Locomotion and Transportation Dataset. *IEEE Dataport*. [Online]. Available: https://ieee-dataport.org/documents/sussex-huawei-locomotion-and-transportation-dataset

[24] WISDM. (2012). WISDM Dataset (Wireless Sensor Data Mining). [Online]. Available: https://www.cis.fordham.edu/wisdm/dataset.php

[25] K. Bonawitz, V. Ivanov, B. Kreuter *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191. doi: 10.1145/3133956.3133982

[26] H. Yin, A. Mallya, A. Vahdat *et al.*, "See through gradients: Image batch recovery via GradInversion," in *Proc. the IEEE/CVF Conference*

*on Computer Vision and Pattern Recognition*, 2021, pp. 16337–16346. doi: 10.1109/cvpr46437.2021.01607