

# CE-FedAvg: A Communication-Efficient Federated Learning Framework for LoRaWAN-Based Edge AI

Syedmohammadamin Razaghi  
Faculty of Computer Science  
University of Vienna  
Vienna, Austria

Atakan Aral  
Faculty of Computer Science  
University of Vienna  
Vienna, Austria

**Abstract**—Federated Learning (FL) enables collaborative model training without centralizing data, yet deployment over low-power wide-area networks (LPWANs), such as LoRaWAN, remains challenging due to severe bandwidth, payload, and duty-cycle constraints. This work examines the feasibility of FL on LoRaWAN by designing and implementing a communication-efficient framework tailored to constrained Internet of Things (IoT) environments. CE-FedAvg is introduced as a communication-efficient variant of FedAvg that restructures client-server exchanges to comply with standard LoRaWAN operation. In addition to sparsification, quantization, and lightweight compression, CE-FedAvg incorporates *seed-based initialization* for deterministic synchronization and explores *model improvements* optimized for ultra-low-throughput communication environments. The framework is fully compatible with the LoRaWAN specification and is implemented across heterogeneous edge devices and a low-power gateway, demonstrating end-to-end feasibility under realistic IoT conditions.

**Index Terms**—Federated Learning, LoRaWAN, Edge AI, Communication Efficiency, Sparsification, Quantization

## I. INTRODUCTION

The rise of the Internet of Things (IoT) and edge computing has spurred interest in *Federated Learning* (FL) as a means to train models collaboratively on distributed devices without centralizing data [1], [2]. By keeping data on devices and exchanging only model updates, FL can preserve privacy and reduce backhaul traffic. In many IoT deployments, long-range, battery-powered sensing relies on *low-power wide-area networks* (LPWANs) such as LoRaWAN for smart metering, agriculture, and industrial condition monitoring [3]. In these settings, FL is attractive because it enables on-device adaptation to local drift while keeping raw data on premises, which reduces uplink volume and recurring connectivity cost, supports privacy/regulatory compliance (e.g., GDPR [4]), and works where only LPWAN coverage exists. LoRaWAN is widely deployed, energy-efficient, and uses unlicensed spectrum. Enabling FL over LoRaWAN would enable large fleets to learn in place without requiring broadband backhaul. However, deploying FL in LoRaWAN introduces strict bandwidth and duty-cycle limitations that can impede model convergence and overall training reliability, particularly when traditional FL protocols are directly applied over LoRaWAN links.

Most existing FL research assumes high-bandwidth and low-latency communication channels, where frequent model synchronization and full-precision parameter exchange are feasible. In contrast, LoRaWAN offers only a few hundred bits per second of uplink bandwidth, with strict 1% duty-cycle limitations and long inter-packet intervals. Moreover, the downlink channel is even more constrained, often operating under asynchronous ALOHA-based access with limited acknowledgments. Consequently, communication strategies in LoRaWAN-based FL must be carefully redesigned to ensure protocol compliance while minimizing resource consumption.

Recent studies have begun to explore FL over LPWAN scenarios. For instance, Singh and Borkotoky [5] develop a detailed FL-over-LoRa simulator, highlighting that LoRaWAN's low data rate and strict duty-cycling can drastically slow down FL training unless communication-efficient techniques are applied. Torres *et al.* [6] demonstrate a federated anomaly detection system for industrial IoT using LoRaWAN, achieving accuracy comparable to a centralized model but requiring careful scheduling of rounds and incurring substantial training-message airtime in their setting. Their work underscores the need to adapt the FL paradigm to LoRaWAN's limitations. Likewise, in a LoRa mesh network, Giménez *et al.* [7] implemented embedded FL on microcontroller-based sensors and confirmed the feasibility of distributed training at the *tiny edge*, although network throughput remained a bottleneck.

This work introduces **CE-FedAvg**, a communication-efficient approach that operates fully within the LoRaWAN standard without modifying MAC or PHY layers. In contrast to approaches that propose new network-layer adaptations or scheduling protocols, CE-FedAvg focuses on optimizing the FL communication pattern itself by combining *update sparsification*, *model quantization*, and *lightweight compression* to reduce uplink payload sizes by up to an order of magnitude while maintaining model convergence. CE-FedAvg further incorporates a *seed-based initialization* mechanism that ensures deterministic synchronization among participating clients, leading to improved reproducibility and stable convergence even under heterogeneous edge conditions. Complementary *model improvements*, such as lightweight convolutional architectures optimized for constrained devices, reduce computational demand and update size. These techniques collectively enable FL to operate efficiently on real LoRaWAN

infrastructure without requiring protocol-level modifications.

The remainder of the paper reviews related work, defines the problem, introduces **CE-FedAvg**, outlines implementation and evaluation, and concludes with a discussion of future work.

## II. BACKGROUND AND RELATED WORK

### A. Federated Learning and Communication Efficiency

The classical FedAvg algorithm [1] and large-scale implementations [8] established the feasibility of training shared models without centralizing data. Communication quickly emerges as a dominant bottleneck in FL, particularly under limited client bandwidth. To address this, prior work has proposed gradient sparsification [9], quantization [10], and update compression [11] to reduce transmitted payloads with limited accuracy loss, providing a foundation for operation in low-capacity networks. However, most communication-efficient methods assume broadband or cellular links. LPWANs such as LoRaWAN differ substantially: they offer extremely low data rates, long inter-packet intervals, and strict duty-cycle limits. In addition, ALOHA-based random access introduces uncertainty due to collisions and constrained acknowledgment windows [12]. Consequently, FL algorithms designed for Wi-Fi or LTE cannot be directly applied to LoRaWAN without violating compliance or energy budgets.

### B. FL in LPWAN and LoRaWAN Contexts

Recent studies have explored federated learning in constrained IoT and LPWAN scenarios. Chen *et al.* [13] examine communication- and resource-optimized FL over wireless IoT networks, but their design assumes relatively high-throughput links compared with LPWANs. Nguyen *et al.* [14] survey FL techniques for IoT and emphasize communication efficiency and device heterogeneity as central challenges in low-power environments. Kaur and Jadhav [15] review FL approaches under resource constraints and identify the lack of practical implementations targeting ultra-narrowband systems such as LoRaWAN. LoRaWAN behavior has been extensively characterized in terms of airtime, capacity, and collision dynamics [16]. Simulation tools, such as LoRaSim [17], enable analysis of large-scale deployments, modeling capture effects and channel diversity. These analyses underline the importance of minimizing transmission airtime, which is the motivation behind CE-FedAvg’s compressed communication design.

### C. Seed-Based Initialization and Model Adaptation

Reproducibility and synchronization are key challenges in the initial round of federated settings with intermittent connectivity. Our framework introduces *seed-based initialization*, where clients share a global pseudorandom seed to initialize model weights deterministically. This approach improves cross-client consistency without increasing communication load. Similar ideas have been explored in adaptive FL literature [18], [19], [20], but CE-FedAvg applies them specifically to LoRaWAN’s constrained communication regime. Architectural adaptation complements communication-side optimizations. Lightweight convolutional designs with reduced channel

widths, batch normalization, and optional depthwise separable layers can reduce training latency and update size with limited accuracy impact on tasks such as FEMNIST [21]. Such model-level choices provide gains comparable to protocol-level compression and highlight the value of jointly considering architecture and communication budgets.

### D. Summary and Research Gap

Existing FL frameworks frequently rely on cross-layer coordination or dynamic MAC control that is incompatible with standard LoRaWAN operation. For example, Yang *et al.* [18] optimize communication–computation trade-offs in wireless networks, while Jin *et al.* [19] balance update frequency and bandwidth to reduce power consumption; da Silva *et al.* [20] study energy-aware user sampling under multichannel ALOHA. Although these approaches effectively reduce energy consumption in broadband or cellular environments, they assume adaptive link-layer control and abundant downlink bandwidth. Such assumptions are not valid for LPWANs like LoRaWAN, where payload size, transmission interval, and duty-cycle are strictly limited by regulation. In contrast, our focus in this work is purely on the application layer, where communication efficiency and synchronization are achieved through sparsification, quantization, and seed-based initialization, without modifying the LoRaWAN MAC. This compliance-oriented design addresses a gap between theoretical proposals and deployable systems by enabling immediate operation over existing LoRaWAN infrastructure.

## III. PROBLEM FORMULATION AND RESOURCE MODEL

### A. FL Objective

Consider clients  $i \in \{1, \dots, N\}$  with local data distributions  $\mathcal{D}_i$  and objectives  $F_i(w) = \mathbb{E}_{(x,y) \sim \mathcal{D}_i} \ell(f_w(x), y)$ . The federated objective is

$$\min_w F(w) = \sum_{i=1}^N p_i F_i(w), \quad \sum_{i=1}^N p_i = 1. \quad (1)$$

FedAvg performs  $E$  local epochs per round  $t$  to produce  $w_i^{t+1}$ , followed by aggregation

$$w^{t+1} = \sum_{i=1}^N p_i w_i^{t+1}. \quad (2)$$

### B. LoRaWAN Constraints

A LoRaWAN transmission with payload  $B$  bytes, spreading factor  $s$ , bandwidth  $W$ , and coding rate  $r$  has airtime  $\text{ToA}(B, s, W, r)$  (e.g., [16]). In EU868, each sub-band enforces a 1% duty cycle: after a packet with airtime  $\tau$ , the device must idle for at least  $99\tau$ . With a per-frame maximum MAC payload (e.g.,  $B_{\max}=222$  bytes for EU868 DR5) [22], federated exchanges must satisfy

$$\text{Packetization: } B_{\text{uplink}} \leq B_{\max}, \quad (3)$$

$$\text{Duty cycle: } \Delta t_{\text{round}} \geq \sum_{k \in \text{pkts}} (1 + 99) \text{ToA}_k, \quad (4)$$

where  $\text{ToA}_k$  denotes airtime of fragment  $k$  and  $\Delta t_{\text{round}}$  is the minimum wall-clock time imposed by regulation per update.

### C. Design Goal

The goal is to minimize communication overhead while preserving convergence, subject to LoRaWAN compliance:

$$\min_{\Theta} \sum_t B_{\text{uplink}}^{(t)} \quad \text{s.t.} \quad \text{LoRaWAN}, \Delta\text{Acc} \leq \epsilon, \quad (5)$$

where  $\Theta$  collects protocol knobs such as sparsity  $K$ , quantization level  $q$ , and scheduling parameters;  $\Delta\text{Acc}$  denotes the accuracy drop relative to a dense baseline, bounded by  $\epsilon$ .

## IV. CE-FEDAVG FRAMEWORK

### A. Overview

CE-FedAvg preserves FedAvg’s round semantics but modifies messages:

- 1) **Local training:** Each client performs  $E$  epochs on data.
- 2) **Sparse delta:** Compute update  $\Delta w_i$ , list Top- $K$  entries  $(\mathcal{I}_i)$ , quantize to  $q$ -bit (float16/int8).
- 3) **Packetization:** Encode indices +  $q$ -quantized values into one or more LoRaWAN-compliant packets.
- 4) **Aggregation:** Server reconstructs sparse updates, aggregates, and issues compact refreshes.

The client-side procedure is summarized in Alg. 1, and the server aggregation is given in Alg. 2.

### B. Seed-Based Initialization

To eliminate the need for transmitting full model weights in the initial round, CE-FedAvg employs a *seed-based initialization* mechanism. The server broadcasts only a small pseudorandom seed  $s_0$ , a CRC checksum, and the initializer type (e.g., Kaiming or Xavier), allowing each client to deterministically reconstruct the same model parameters  $w^0 = \text{Init}(s_0)$  locally. This avoids dense model broadcast while ensuring identical initial states.

### C. Model Improvements (Lightweight CNN)

Communication-side compression is complemented with architecture-level reductions. A compact model with reduced channel widths, batch normalization, and optional depthwise separable layers lowers parameter count while maintaining accuracy with substantially fewer bytes per round. While not universal, lightweight architectures can be tailored for low-bandwidth settings (e.g., pruning or depthwise factorization), and can be as impactful as communication-side compression.

### D. Update Compression and Packetization

Let  $\Delta w \in \mathbb{R}^d$  denote the flattened client update. CE-FedAvg retains indices  $\mathcal{I}$  for the Top- $K$  magnitudes and quantizes their values  $v$  using scale  $s$  and zero-point  $z$ :

$$\hat{v} = \text{clip}(\text{round}(\frac{v}{s}) + z), \quad v \approx s(\hat{v} - z). \quad (6)$$

Indices and quantized values are entropy-coded into a payload  $\leq$  *maximum MAC payload*; if the payload exceeds this limit, the modem fragments it into frames and appends CRC.

---

### Algorithm 1 CE-FedAvg (client $i$ )

---

- 1: **Input:** seed  $s_0$ , model  $w^t$ , epochs  $E$ , sparsity  $K$ , quantizer  $Q_q$
  - 2: **if**  $t = 0$  **then**
  - 3:  $w^0 \leftarrow \text{Init}(s_0)$
  - 4: **end if**
  - 5:  $w_i^{t+1} \leftarrow \text{LocalTrain}(w^t, E)$
  - 6:  $\Delta w_i \leftarrow w_i^{t+1} - w^t$
  - 7:  $(\mathcal{I}_i, v_i) \leftarrow \text{TopK}(\Delta w_i, K)$
  - 8:  $\hat{v}_i \leftarrow Q_q(v_i)$
  - 9:  $\text{pkt} \leftarrow \text{Pack}(\mathcal{I}_i, \hat{v}_i)$
  - 10: **Transmit**  $\text{pkt}$  over LoRaWAN (duty-cycle compliant)
- 

---

### Algorithm 2 Server aggregation

---

- 1: **Input:** packets  $\{\text{pkt}_i\}$ , current model  $w^t$
  - 2: **for each**  $i$  **do**
  - 3:  $(\mathcal{I}_i, \hat{v}_i) \leftarrow \text{Unpack}(\text{pkt}_i)$
  - 4:  $v_i \leftarrow Q_q^{-1}(\hat{v}_i)$ ; reconstruct sparse  $\Delta w_i$
  - 5: **end for**
  - 6:  $w^{t+1} \leftarrow \text{Aggregate}(\{w^t + \Delta w_i\})$
  - 7: **Broadcast** compact refresh if needed
- 

## V. IMPLEMENTATION

### A. Hardware Topology and Runtime Orchestration

**Edge clients.** Each physical client is an NVIDIA Jetson Orin Nano that performs local training and inference, following typical edge-deployment architectures for distributed intelligence [23]. To comply with LoRaWAN constraints while keeping the Linux training stack unchanged, each Jetson attaches to an *external* LoRaWAN modem: an ESP32 driving a Semtech SX1262 transceiver. The Jetson and ESP32 communicate over UART using a compact binary framing. The assembled modem prototype is shown in Fig. 1.

**Gateway and core network.** Uplinks and downlinks traverse a Raspberry Pi 5 equipped with a Semtech SX1303 LoRaWAN Gateway HAT. The gateway forwards frames to a LoRaWAN Network Server (LNS), where ChirpStack is deployed. The FL server (aggregator) connects to the LNS via MQTT, mapping application topics to client identifiers. This setup follows LoRaWAN architecture principles [22], [17].

### B. Software Stack

Training employs PyTorch under Flower 1.19.0. The server implements the custom FedAvg-compatible strategy that (i) accepts sparse, quantized uplinks; (ii) performs weighted aggregation over reconstructed sparse deltas; and (iii) emits *sparse download refreshes* during evaluation rounds. This strategy extends the classical FedAvg algorithm [1], [8]. Table I summarizes the configuration and default settings. The source code of CE-FedAvg is available as an open repository<sup>1</sup>.

<sup>1</sup><https://github.com/aminrazaghi/CE-FedAvg>

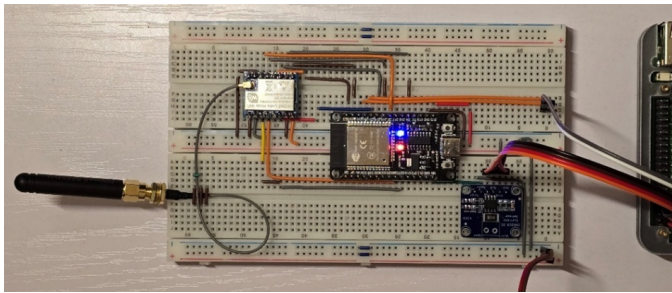


Fig. 1. Prototype LoRaWAN modem used, consisting of an ESP32 microcontroller, SX1262 transceiver, and external antenna.

### C. Algorithm Overview

**Seed-Based Initialization.** At round  $t=0$ , the server transmits a *seed packet* containing a 32-bit PRNG seed  $s_0$ , the initializer type, and a CRC32 over the deterministically generated  $w^0$ . Each client reconstructs  $w^0$  locally and validates the checksum (Alg. 1, l. 2–3). This procedure avoids dense model broadcast and guarantees that heterogeneous devices (CPU/GPU, library versions) start from identical weights.

**Client-side update and uplink.** Each client trains for  $E$  local epochs (Alg. 1, l. 4), computes the model delta  $\Delta w$  (l. 5), selects Top- $K$  indices by magnitude and quantizes the corresponding values (l. 6–7). Indices (relative-offset encoded) and quantized values are packed into LoRaWAN-compliant frames and transmitted subject to duty-cycle limits (l. 8–9).

**Server-side reconstruction and aggregation.** The server receives fragments, unpacks indices/values, dequantizes, and reconstructs each sparse  $\Delta w_i$  (Alg. 2, l. 2–4). It then performs weighted FedAvg over the union of indices to obtain  $w^{t+1}$  (l. 6). During evaluation or when drift is detected, the server optionally sends a compact *refresh delta* (sparsified snapshot difference) to clients (l. 7).

**Timing and compliance.** For each fragment of size  $B \leq \text{maximum MAC payload}$ , the modem enforces standard-compliant pacing using the LoRa time-on-air formula [16] and the EU868 1% duty-cycle off-time. Thus, a  $J$ -fragment update incurs a wall-clock lower bound of  $\sum_{j=1}^J (\text{ToA}_j + 99 \text{ ToA}_j)$ . This schedule is enforced by the ESP32, ensuring transmissions remain standard-compliant without server-side pacing.

**Reliability and Integrity.** All fragments include CRC16; seed-init messages include CRC32 over  $w^0$ . The modem may retry uplinks with randomized backoff if MAC-layer confirms are requested (disabled by default to conserve downlink). On the server, missing fragments trigger a per-update timeout; incomplete updates are discarded for that round to preserve aggregation correctness. Similar reliability mechanisms are reported for FL over LoRaWAN systems [6], [24].

## VI. EVALUATION

### A. Experimental Setup

Evaluation adopts a hybrid *testbed-simulation* methodology. The physical testbed comprises three NVIDIA Jetson Orin Nano clients, each connected over UART to an

TABLE I  
LoRaWAN CONFIGURATION AND DEFAULT LEARNING/PROTOCOL SETTINGS

LoRaWAN Region, Classes, and Rate Control		
Parameter	Setting	Notes
Region	EU868	LoRaWAN 1.0.3
Join	OTAA	
Device class	C	reduces downlink latency
Data rate	DR5	SF7/125 kHz
Max MAC payload	222 B	EU868, DR5
Duty cycle	1% per sub-band	enforced per fragment
Uplink/Downlink	ALOHA	confirmed msg disabled
Default Hyperparameters and Protocol Knobs		
Component	Setting	Notes
Local epochs ( $E$ )	3	1–5 used in ablations
Batch size	32	fixed across runs
Optimizer	SGD (mom. 0.9)	LR 0.01, decay on plateau
Sparsity ( $K$ )	0.10 (Top-10%)	down to 0.01 when needed
Quantization (uplink)	float16	int8 optional

ESP32+sx1262 LoRaWAN modem. Communication proceeds via a Raspberry Pi 5 gateway (SX1303) to a ChirpStack network server. This setup validates LoRaWAN compliance, packetization, and per-fragment duty-cycle timing. To assess scalability, trace-driven simulations extend the system to 30 logical clients by replaying airtime traces measured on the physical deployment.

Experiments use FEMNIST (LEAF) with writer-based non-IID partitions (62 classes,  $28 \times 28$  grayscale) [21]. A common CNN baseline (two convolutional layers plus a fully connected head) is compared against a lightweight CNN with three convolutional blocks (Conv→BN→ReLU→Pool), an adaptive global pooling layer, and a final fully connected classifier. The lightweight model reduces parameters from  $\sim 214.6k$  to  $\sim 27.6k$  (–87%), lowering both local compute and nominal update size. Activations and weights remain in float32 during training; only transmitted deltas are quantized.

Unless otherwise noted, experiments run 30 rounds with the default settings in Table I. Reported metrics include global test accuracy (%), per-round communication volume (KB) in each direction, and LoRaWAN feasibility via time-on-air (ToA) and duty-cycle-imposed wall time for EU868. Each configuration is repeated five times, and reported values represent the mean over these runs. Error bars represent  $\pm 1$  standard deviation.

### B. Numerical Results

*a) Lightweight architecture maintains accuracy at lower cost.* Key quantitative results are reported in Table II. The baseline CNN attains 85.0% accuracy with a model update size of 858 KB per round. The lightweight CNN reaches 84.5% with float16 while reducing payload to 50.3 KB (17 $\times$  smaller). Applying moderate sparsification trades bytes for minor accuracy loss: Top-10% achieves 83.4% (6.21 KB), whereas Top-1% yields 63.6% (0.75 KB). See the accuracy–sparsification trend in Fig. 2.

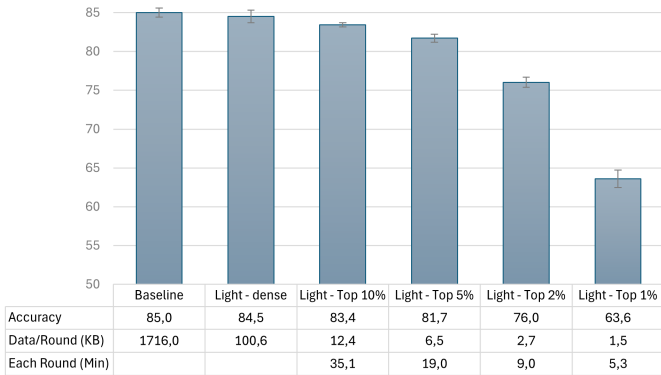


Fig. 2. Accuracy vs. Symmetric uplink/downlink Top- $K$  sparsification. (Mean over 5 runs;  $\pm 1$  SD error bars)

TABLE II  
KEY RESULTS ON FEMNIST UNDER LORAWAN-AWARE CE-FEDAVG

Setting	Acc. (%)	Uplink/Downlink (KB/rd)
Baseline CNN, dense	85.0	858
Lightweight, dense (fp16)	84.5	50.3
Lightweight, Top-10%	83.4	6.21
Lightweight, Top-5%	81.7	3.26
Lightweight, Top-2%	76.0	1.37
Lightweight, Top-1%	63.6	0.75

*b) Rounds vs. local epochs (fixed 90 total epochs):*

With a fixed total training budget and Top-10% sparsification, 30 rounds  $\times$  3 epochs attains 83.4% accuracy, outperforming 45 $\times$ 2 (82.9%), 18 $\times$ 5 (81.4%), and 90 $\times$ 1 (81.8%). The trade-off is illustrated in Fig. 3.

*c) LoRaWAN Feasibility: Airtime and Duty-Cycle:*

For EU868 at DR5, a 222 B frame has ToA  $\approx$  0.369 s; the 1% duty cycle imposes  $\approx$  36.9 s off-time per frame, permitting roughly 97 frames per hour. For example, with Top-1% sparsification ( $\sim$  0.75 KB), the update payload is transmitted in 4 fragments, totaling  $\approx$  1.47 s airtime and a duty-cycle-compliant wall-clock duration of  $\approx$  1.52 min per update. Larger payloads scale proportionally. These measurements confirm that CE-FedAvg’s lightweight and sparse configurations remain standard-compliant on LoRaWAN.

*d) Convergence time vs. sparsity ( $K$ ):*

Fig. 4 reports accuracy over wall-clock time (minutes) under different Top- $K$  sparsification levels. The Top-5% configuration ( $K=0.05$ ) converges the fastest to 80% test accuracy, reaching this threshold slightly earlier than Top-10% ( $K=0.10$ ). In contrast, Top-2% ( $K=0.02$ ) does not attain 80%, and Top-1% ( $K=0.01$ ) saturates around mid-60%. This result complements the byte-accuracy trade-off in Fig. 2: moderate sparsity (5–10%) yields both competitive final accuracy and faster time-to-target accuracy, while aggressive sparsification slows learning and does not reach the same quality.

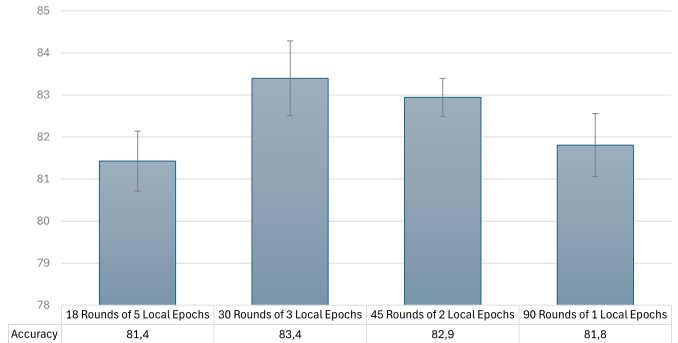


Fig. 3. Fixed 90 total epochs: 30 rounds  $\times$  3 epochs strikes the best balance. (Mean over 5 runs;  $\pm 1$  SD error bars)

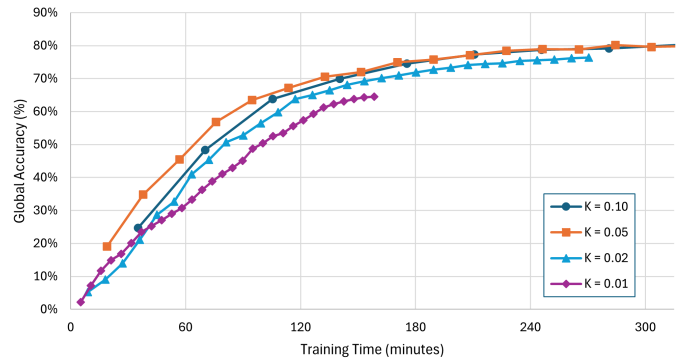


Fig. 4. Convergence speed under different Top- $K$  levels. Curves show global accuracy vs. wall-clock time (minutes).

## VII. DISCUSSION

### A. Feasibility of FL over LoRaWAN

The results indicate that communication-aware model design and protocol choices can bring FL within LoRaWAN’s tight constraints. Three factors are decisive: (i) a lightweight model that trims parameters without sacrificing accuracy. (ii) sparse, quantized updates that reduce communication overhead. (iii) seed-based initialization that removes dense model broadcast at  $t=0$ .

### B. Rounds vs. local computation

Under a fixed total budget of 90 local epochs, 30 $\times$ 3 rounds $\times$ epochs offers the best balance (83.4%), with 45 $\times$ 2 close behind (82.9%). This suggests moderate local work per round keeps the global model “fresh” without incurring excessive communication or local overfitting. In duty-cycle-limited settings, these schedules are preferable to very frequent short rounds or very long local training.

### C. Reliability, timing, and scheduling

Fragment-level CRC and per-fragment pacing on the modem enforce standard-compliant transmissions without server pacing. In practice, missed fragments or sporadic downlink availability can be handled by discarding incomplete updates for that round or by issuing periodic sparse refreshes. Because

off-time dominates round time, *when* to schedule clients can matter as much as *how much* they transmit.

#### D. Limitations

The physical testbed includes three clients; scalability to larger cohorts is evaluated via trace-driven simulation rather than a large over-the-air deployment. Experiments focus on FEMNIST; broader workloads (e.g., metering, industrial, environmental/agriculture) remain to be explored. Energy per update is inferred from airtime rather than direct power measurements. Secure aggregation and differential privacy are not enabled here, and would add bytes and latency that must be budgeted. Finally, the design maintains LoRaWAN MAC/PHY, which preserves compatibility but leaves admission control, collisions, and ADR behavior to the underlying network.

#### E. Paths forward

Two directions appear immediately promising: (i) extending CE-FedAvg to additional IoT workloads and larger models representative of LoRaWAN deployments, while jointly adapting sparsity and quantization to optimize *time-to-target* accuracy under airtime and duty-cycle constraints. (ii) incorporating system and model level enhancements, such as structured sparsification, error-feedback mechanisms, or lightweight architectural adaptations, to improve convergence without increasing payload size. System-level strategies, including ADR-aware or staggered client selection and loss-resilient update handling, may further mitigate collisions and raise feasible round rates at scale. Security and privacy add-ons (e.g., secure aggregation and differential privacy) should be integrated with explicit airtime budgets and validated under realistic deployment conditions.

### VIII. CONCLUSION

This work presented **CE-FedAvg**, a communication-efficient FL framework that operates fully within the LoRaWAN standard. Seed-based initialization, a lightweight CNN, and sparse/quantized updates jointly reduce update payload from hundreds of kilobytes to only a few kilobytes while maintaining competitive accuracy on non-IID FEMNIST. On a LoRaWAN testbed with Jetson Orin Nano clients and an SX1303 gateway, Top-1% updates complete in  $\approx 5.3$  min at EU868 DR5, enabling  $\sim 12$  rounds per hour; Top-5% offers a favorable accuracy-bytes trade-off at 3.26 KB per update. These findings indicate that *federated learning over LoRaWAN is feasible in practice* without MAC/PHY modifications. Future work will integrate security primitives and validate broader tasks at larger scales.

### REFERENCES

- [1] B. McMahan *et al.*, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. of AISTATS*, 2017, pp. 1273–1282.
- [2] P. Kairouz *et al.*, “Advances and open problems in federated learning,” *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [3] S. Ahmad, *et al.*, “Sustainable environmental monitoring via energy and information efficient multinode placement,” in *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22065–22079, 2023.
- [4] European Union, “General Data Protection Regulation (GDPR),” Regulation (EU) 2016/679, Apr. 27, 2016.
- [5] A. Singh and S. S. Borkotoky, “Federated learning over LoRa networks: Simulator design and performance evaluation,” arXiv:2508.10574, 2025.
- [6] O. Torres-Sanchez *et al.*, “Federated learning framework for LoRaWAN-enabled IIoT communication: A case study,” *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 24944–24957, 2025.
- [7] N. Llisterra Giménez, J. M. Solé, and F. Freitag, “Embedded federated learning over a LoRa mesh network,” *Pervasive and Mobile Computing*, vol. 93, p. 101819, 2023.
- [8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan *et al.*, “Towards federated learning at scale: System design,” in *Proc. 2nd SysML Conf.*, 2019.
- [9] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent,” in *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 440–445.
- [10] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via gradient quantization and encoding,” in *Proc. 31st Int. Conf. Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1709–1720.
- [11] F. Sattler, S. Wiedemann, K. R. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-i.i.d. data,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2020.
- [12] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, “An efficient framework for clustered federated learning,” in *Proc. 34th Conf. Neural Information Processing Systems (NeurIPS)*, 2020.
- [13] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, “Federated Learning Over Wireless IoT Networks With Optimized Communication and Resources,” *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16592–16605, Sep. 2022.
- [14] 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.
- [15] I. Kaur and A. J. Jadhav, “Federated Learning in IoT: A Survey from a Resource-Constrained Perspective,” in *Proc. 2023 Int. Conf. on Artificial Intelligence Robotics, Signal and Image Processing (AIRoSIP)*, pp. 376–381, 2023.
- [16] B. Reynders, W. Meert, and S. Pollin, “Power and spreading factor control in LoRa networks,” in *Proc. IEEE Int. Conf. Communications (ICC)*, pp. 1–6, 2017.
- [17] M. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “Do LoRa low-power wide-area networks scale?” in *Proc. ACM Int. Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2016, pp. 59–67.
- [18] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy-efficient federated learning over wireless communication networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2021.
- [19] R. Jin, X. He, and H. Dai, “Communication efficient federated learning with energy awareness over wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 5204–5219, 2022.
- [20] R. V. da Silva, O. L. A. Lopez, and R. D. Souza, “Energy-aware federated learning with distributed user sampling and multichannel ALOHA,” *IEEE Communications Letters*, vol. 27, no. 10, pp. 2867–2871, 2023.
- [21] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, “LEAF: A benchmark for federated settings,” arXiv preprint arXiv:1812.01097v3, 2019.
- [22] LoRa Alliance, “LoRaWAN 1.0.3 specification,” 2018. [Online]. Available: <https://loro-alliance.org/>
- [23] A. Aral and I. Brandić, “Learning spatiotemporal failure dependencies for resilient edge computing services,” *IEEE Trans. Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1578–1590, 2021.
- [24] F. Busacca, *et al.*, “FedLoRa: IoT Spectrum Sensing Through Fast and Energy-Efficient Federated Learning in LoRa Networks,” in *Proc. 2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, 2024, pp. 47–54.