Opportunistic Model Ensembling for Decentralized Learning over Intermittent Edge Networks

Brynx Junil Alegarbes [¶§], Victor Romero II [*§§], Tomokazu Matsui [*†],
Yuki Matsuda [‡*†], Hirohiko Suwa [*†], and Keiichi Yasumoto [*†]
[*] Nara Institute of Science and Technology, Ikoma, Nara, Japan
[†] RIKEN Center for Advanced Intelligence Project AIP, Chuo, Tokyo, Japan
[‡] Okayama University, Okayama, Okayama, Japan
[§] University of the Philippines Tacloban College, Tacloban City, Philippines
[¶] Ateneo de Manila University, Quezon City, Philippines
Emails: brynx.alegarbes@student.ateneo.edu, romero.victor_militante.rs1@is.naist.jp, m.tomokazu@is.naist.jp,
yukimat@okayama-u.ac.jp, h-suwa@is.naist.jp, yasumoto@is.naist.jp

***Abstract*—Reliable training and deployment of machine learning models in remote, resource-constrained environments is hindered by intermittent connectivity, limited computation, and decentralized data. These challenges limit the potential of applications like plant disease detection, where the ability of edge devices to collaboratively learn from local observations could greatly enhance agricultural resilience. We propose a simple yet efficient approach for decentralized model training that does not rely on continuous connectivity. Edge nodes train lightweight base models on local data and exchange them opportunistically during transient peer-to-peer encounters, enabling incremental refinement across diverse datasets. Rather than synchronized updates, the system ensembles these independently trained models to support robust predictions under sparse, asynchronous communication. Experiments show that our method achieves strong classification performance across varying levels of connectivity, heterogeneity, and node participation, offering a scalable solution for distributed learning in real-world agricultural and infrastructure-limited settings.**

***Index Terms*—opportunistic networks, ensemble techniques, federated learning**

I. Introduction

In many real-world scenarios—particularly in rural regions—deploying machine learning (ML) systems presents unique logistical and infrastructural challenges. These environments often lack reliable networks, centralized data infrastructure, and sufficient computational capacity to support conventional ML workflows. A representative example is plant disease detection, where locally observed crop conditions could be used to train diagnostic models that support smallholder farmers in monitoring plant health [1]. Standard ML pipelines are poorly suited for such scenarios, as they typically assume centralized data collection and reliable infrastructure [2].

Federated learning (FL) offers a promising alternative by enabling decentralized model training, where devices perform local computations without sharing raw data [3]. Recent works have explored the application of FL, demonstrating strong performance in controlled settings [4]–[7]. However, conventional FL frameworks often assume frequent and reliable communication rounds, as well as the presence of a centralized orchestrator. These assumptions are difficult to satisfy in rural areas due to intermittent connectivity and the impracticality of deploying and maintaining a central server. Such constraints call for decentralized approaches that can function under sporadic connectivity and do not rely on synchronized training or global coordination.

To address these challenges, we propose a framework that enables decentralized model refinement through the ensembling of sequentially trained lightweight base models. Rather than synchronizing full models across devices, nodes independently refine smaller classifiers and opportunistically exchange them during transient peer-to-peer encounters. This communication-efficient strategy sidesteps the need for continuous aggregation and allows learning to proceed asynchronously, even under sparse connectivity. By progressively integrating knowledge from diverse local datasets across nodes, the ensemble evolves to capture generalized patterns while preserving data locality and minimizing resource demands. Through extensive simulations, we demonstrate that our approach maintains high classification accuracy and exhibits strong resilience to data heterogeneity and network sparsity. Our findings highlight the feasibility of training effective ML models in infrastructure-limited environments, thereby promoting ubiquitous intelligence by enabling decentralized learning in marginalized regions.

Contributions. In this paper:

- We propose an opportunistic federated learning framework that constructs classifier heads via ensembling of sequentially trained base models exchanged over transient peer-to-peer links;
- We develop a lightweight training and communication protocol designed for deployment in highly sparse and dynamic topologies without centralized coordination;
- We evaluate the proposed method on real-world plant disease datasets, systematically comparing it with classical, local, and federated learning baselines, and demonstrate its robustness and scalability under varying levels of data heterogeneity and network sparsity.

The study was supported by JSPS KAKENHI JP25K03107.

II. Related Work

We organize the related work into two areas: recent efforts in opportunistic federated learning under limited connectivity and prior studies on federated learning for plant disease detection.

A. Opportunistic Federated Learning

The vision of a decentralized and democratic training process has spurred considerable research into federated learning methods that do not depend on conventional network infrastructure [8]–[12]. In under-resourced or remote settings, participants cannot rely on continuous connectivity; instead, they depend on sporadic, opportunistic interactions using any available communication interface to exchange model updates with nearby nodes. This scenario diverges from traditional decentralized federated learning, which typically relies on regular, orchestrated communication rounds to converge on a shared global model [13].

One early attempt to adapt FL to such opportunistic conditions is proposed by Lee et al. [8], who address the challenge of personalization by targeting subsets of class labels, where local updates are performed during brief encounters. Their method requires nodes to share label distribution information and predict encounter durations, enabling dynamic, encounter-aware collaboration. While the study frames personalization as user-specific adaptation, this shift can also be seen as a practical response to the difficulty of sustaining a consistent global model in opportunistic settings. To mitigate data skew, they adopt opportunistic momentum, stabilizing training at the cost of storing gradient history across encounters.

Meanwhile, in [10] the authors shift focus toward optimizing communication efficiency in pairwise collaborations within a tourism object detection domain. By employing a regressor that predicts the potential accuracy gain from aggregating two models, they ensure that model exchanges occur only when the anticipated improvement surpasses a predefined threshold, thereby reducing transmission costs.

On the other hand, Ochiai et al. [9] extend federated learning to mobile ad hoc networks through their wireless ad-hoc federated learning (WAFL) framework. Unlike traditional centralized approaches, WAFL attempts to construct a generalized model in a completely decentralized manner, operating on the assumption however that nodes can freely exchange model updates with all immediate neighbors.

Addressing the complexities of large-scale heterogeneous networks, Deng and Yan [11] propose a distributed grouping protocol that partitions the network into small, localized clusters—each containing no more than four nodes. Federated learning is carried out within these compact groups, enabling the system to manage node and data heterogeneity while maintaining communication efficiency. However, the proposed approach assumes a relatively stable topology within each cluster to ensure convergence.

A further direction explored in the literature is the decomposition of neural networks into independent, non-overlapping subnetworks [12], [14], rather than training the full model at each node. In this design, smaller subnetworks preserve the layer-wise structure but contain fewer parameters, allowing them to be individually assigned, trained by different nodes, and exchanged during encounters. This strategy enables parallel and conflict-free updates while reducing the computational burden on each node, thereby encouraging broader participation in training. However, its flexibility is limited since each subnetwork remains strictly tied to a predefined portion of the full model.

B. Federated Learning for Plant Disease Detection

Federated learning (FL) has been increasingly applied to agricultural settings as a means of preserving data privacy while enabling collaborative model training across distributed devices. Mehta et al. [4] introduced a federated convolutional neural network (CNN) model for tomato leaf disease severity classification, demonstrating accuracies between 96% and 98% across different severity levels. Their results highlight the viability of FL in constrained agricultural domains where centralized data aggregation is impractical. Similarly, Mamba et al. [5] leveraged both CNNs and vision transformers (ViTs) in a federated setup on the PlantVillage dataset [15], illustrating the potential of advanced deep architectures. However, their study also underscored the limitations of ViTs, particularly in terms of computational overhead, which can pose significant challenges for edge devices with limited resources.

Aggarwal et al. [6] focused on developing lightweight FL frameworks, proposing a method tailored for rice leaf disease classification that achieved strong performance under both IID and non-IID data distributions. Their work emphasizes the importance of accommodating heterogeneity in real-world deployments. Complementing these efforts, Caminha et al. [7] explored the use of mobile devices and drones to prototype federated model training in agricultural contexts, achieving high accuracy while maintaining data confidentiality. Collectively, these studies demonstrate the promise of FL in agricultural sensing, but they largely assume stable communication infrastructure and coordinated aggregation, assumptions that may not hold in rural or resource-constrained environments.

C. Positioning of this Study

Building on these foundations, our study explores opportunistic federated learning in infrastructure-limited environments, using plant disease detection as a motivating domain. In contrast to approaches that rely on predefined model partitioning or frequent aggregation opportunities, we propose an ensembling strategy in which lightweight base models are trained sequentially and exchanged during transient peer-to-peer encounters. To support this design, we incorporate feature-based transfer learning, enabling nodes to train compact classifier heads. This approach allows nodes to retain functional models throughout training, supports decentralized collaboration under high network sparsity, and minimizes both computational and communication overhead. While agricultural sensing provides a compelling real-world scenario, the proposed framework is broadly applicable to any mobile edge setting characterized by intermittent connectivity and decentralized data.

III. Methodology

A. Preliminaries

Let $C = \{c_0, c_1, ..., c_{N-1}\}$ denote the set of N participating nodes. The nodes are mobile, moving within a spatial domain according to individual movement patterns that determine their encounters over time. Let $\mathcal{G}_t = (C, E_t)$ represent the time-varying communication graph at time step t, where $E_t \subseteq C \times C$ is the set of active communication links. A connection between two nodes c_i and c_j is established, i.e., $(c_i, c_j) \in E_t$, when they come within communication range at time t. Since nodes move independently, $\mathcal{G}_t$ evolves dynamically, with connections forming and breaking over time. Each node c_i holds a private dataset $\mathcal{D}_i$, with the global dataset defined as $\mathcal{D} = \bigcup_{i=0}^{N-1} \mathcal{D}_i$. These local datasets are assumed to be disjoint, i.e., $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for $i \neq j$.

In opportunistic federated learning, where communication is sparse and device resources are limited, reducing overhead is essential. To this end, we adopt a feature-based transfer learning formulation in which the model parameters are conceptually divided into two main components: the feature extractor, denoted by θ_{feat}, and the classifier head, denoted by θ_{cls}. The feature extractor, frozen during training, transforms raw inputs into intermediate representations and is derived from a pre-trained backbone. The classifier head θ_{cls}, by contrast, is trainable and adapted to the target task using locally available data at each node.

We model the classifier head θ_{cls} as an ensemble of K base models, each parameterized by θ_{cls}^k for $k \in \{0, 1, \dots, K-1\}$. By default, we assume $K = N$ to maximize participation, assigning one base model per node so that all nodes actively contribute to training. This is not a strict requirement—when $K < N$, only a subset of nodes participate at each time step, while others remain inactive until assigned a model. While base models could, in principle, differ in architecture or size, we focus on configurations where the combined capacity of all base models matches that of a single classifier head in conventional federated learning. This design aligns with the constraints of opportunistic settings, where communication is intermittent and devices are resource-constrained. By training smaller, modular models, we further reduce both computational and communication overhead, enabling broader participation while preserving model expressiveness.

Given an input feature representation $z = f_{\text{feat}}(x; \theta_{\text{feat}})$, where x is the raw input, each base model produces an output $\hat{y}^k$ by applying a shared classifier function f_{cls}, parameterized by its respective weights θ_{cls}^k:

$$\hat{y}^k = f_{\text{cls}}(z; \theta_{\text{cls}}^k), \quad k \in \{0, 1, \dots, K-1\}.$$

The final prediction $\hat{y}$ is computed by applying an aggregation function $\mathcal{A}(\cdot)$ over all base model outputs:

$$\hat{y} = \mathcal{A}(\hat{y}^0, \hat{y}^1, \dots, \hat{y}^{K-1}).$$

The choice of $\mathcal{A}(\cdot)$ defines how individual predictions are combined. A common approach is *max voting*, where each base model selects a predicted class label $c^k = \arg\max_j \hat{y}_j^k$, and the final output is the most frequently predicted label:

$$\hat{y} = \arg\max_c \sum_{k=0}^{K-1} \mathbb{I}(c^k = c),$$

where $\mathbb{I}(\cdot)$ is the indicator function, which evaluates to 1 when a base model votes for class c, and 0 otherwise. The overall objective is to minimize the ensemble's classification error by refining the constituent base models.

In standard ensembling, base models are trained independently on centrally available data, assuming unrestricted access and concurrent optimization. In contrast, our setting involves data segregation across nodes and opportunistic communication, preventing direct coordination. To address these challenges, we adopt a sequential training strategy, where base model parameters are refined progressively through intermittent node encounters.

B. Sequential Training of Base Models

Before training begins, base models are uniquely assigned to participating nodes. Given the decentralized nature of training and the opportunistic encounters between nodes, these models are progressively refined as they are exposed to different datasets over time. To structure this process, training is divided into T discrete time steps, indexed by $t \in \{0, 1, \dots, T-1\}$. At each step, nodes perform local updates on their assigned base models using only their respective datasets.

Let $\hat{\theta}_{\text{cls},i}^t$ denote the parameters of the base model assigned to node c_i at time step t, where $\hat{\theta}_{\text{cls},i}^t \in \{\theta_{\text{cls}}^0, \theta_{\text{cls}}^1, \dots, \theta_{\text{cls}}^{K-1}\}$. Specifically, at time step t, c_i minimizes the local objective:

$$\mathcal{L}_i(\hat{\theta}_{\text{cls},i}^t, \mathcal{D}_i) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y) \in \mathcal{D}_i} \ell(f_{\text{cls}}(z; \hat{\theta}_{\text{cls},i}^t), y),$$

where $z = f_{\text{feat}}(x; \theta_{\text{feat}})$ is the extracted feature representation, and $\ell(\hat{y}, y) = -\sum_j y_j \log \hat{y}_j$ is the *categorical cross-entropy loss*.

Each node updates its assigned base model for a single epoch using stochastic gradient descent (SGD) or an equivalent optimization method, following the update rule:

$$\hat{\theta}_{\text{cls},i}^{t'} = \hat{\theta}_{\text{cls},i}^t - \eta \nabla_{\hat{\theta}_{\text{cls},i}^t} \mathcal{L}_i(\hat{\theta}_{\text{cls},i}^t, \mathcal{D}_i), \tag{1}$$

where $\hat{\theta}_{\text{cls},i}^{t'}$ represents the *updated version* of the model parameters after training at time t. The term $\nabla_{\hat{\theta}_{\text{cls},i}^t} \mathcal{L}_i(\hat{\theta}_{\text{cls},i}^t, \mathcal{D}_i)$ denotes the gradient of the loss function evaluated on the local dataset $\mathcal{D}_i$, and η represents the learning rate.

Nodes leverage opportunistic encounters to exchange base models, thereby exposing them to a broader range of local datasets. Let $\mathcal{N}_i^t = \{c_j \in C \mid (c_i, c_j) \in E_t\}$ denote the set of neighbors of node c_i at time t. We assume symmetric connectivity, meaning $c_j \in \mathcal{N}_i^t$ if and only if $c_i \in \mathcal{N}_j^t$. Suppose that at $t = 1$, node c_0 has only c_1 as a neighbor. After exchanging base models, the assignments at $t = 2$ are updated such that $\hat{\theta}_{\text{cls},0}^2 \leftarrow \hat{\theta}_{\text{cls},1}^{1'}$ and $\hat{\theta}_{\text{cls},1}^2 \leftarrow \hat{\theta}_{\text{cls},0}^{1'}$, where $\hat{\theta}_{\text{cls},0}^{1'}$ and

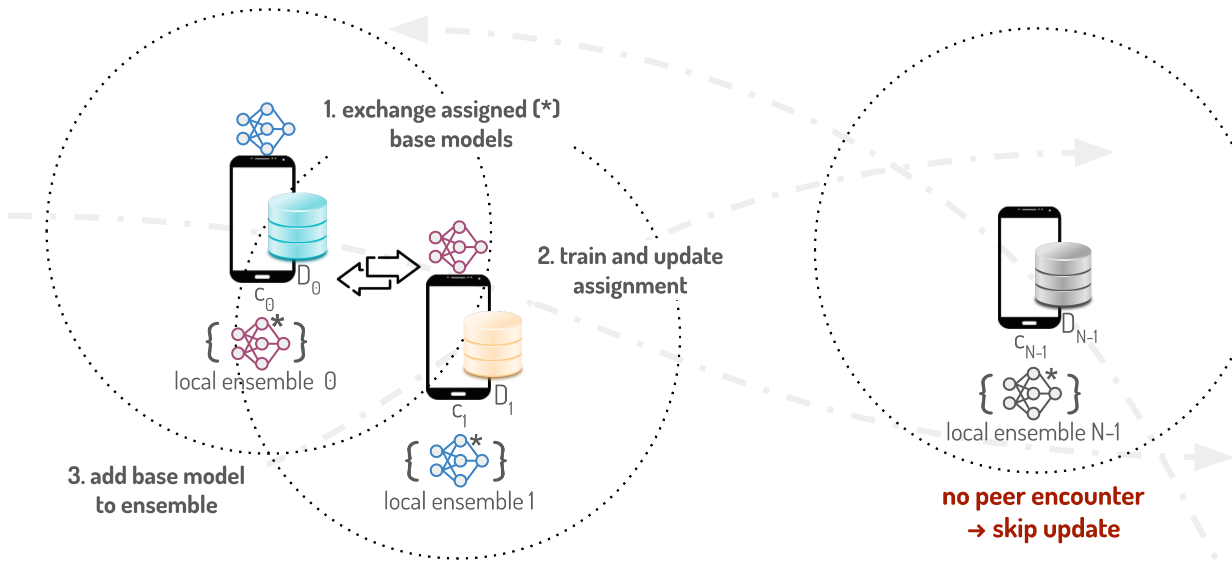

Fig. 1: Schematic of the proposed decentralized training process. Each node holds a local dataset D_i and one assigned base model θ_k (marked with *). Upon encountering a peer, (1) nodes exchange their assigned base models, (2) locally train the received model, and (3) add it to their local ensemble. Nodes without a peer in range skip updates for that round.

$\hat{\theta}^{1'}_{\text{cls},1}$ are the locally trained versions prior to exchange. Both nodes then proceed to refine their newly assigned models using their respective datasets. If a node has no neighbors at a given time step, it abstains from updating to avoid overfitting.

C. Ensembling Strategy in Opportunistic Setting

Each node c_i maintains a *local ensemble* $\mathcal{E}^t_{\text{local},i}$, which consists of base models that have been trained on its local dataset up to time t. Consider nodes c_0 and c_1 as neighbors at $t = 0$, with initial assignments $\hat{\theta}^0_{\text{cls},0} = \theta^0_{\text{cls}}$ and $\hat{\theta}^0_{\text{cls},1} = \theta^1_{\text{cls}}$. After local training, their ensembles are:

$$\mathcal{E}^0_{\text{local},0} = \{\theta^{(0,0')}_{\text{cls}}\}, \quad \mathcal{E}^0_{\text{local},1} = \{\theta^{(1,0')}_{\text{cls}}\},$$

where $\theta^{(k,t')}_{\text{cls}}$ denotes the locally trained version of the k^{th} base model at time t.

At $t = 1$, each node trains the base model received during the exchange at $t = 0$. Their updated ensembles become:

$$\mathcal{E}^1_{\text{local},0} = \{\theta^{(0,0')}_{\text{cls}}, \theta^{(1,1')}_{\text{cls}}\}, \quad \mathcal{E}^1_{\text{local},1} = \{\theta^{(1,0')}_{\text{cls}}, \theta^{(0,1')}_{\text{cls}}\}.$$

If another exchange and retraining occur at $t = 2$, the updated base models $\theta^{(0,2')}_{\text{cls}}$ and $\theta^{(1,2')}_{\text{cls}}$ are added to the respective local ensembles of each node and their previously stored instances are evicted.

IV. Experiments

We evaluate our method in a hypothetical plant disease detection scenario, where mobile devices operate in remote agricultural areas under limited connectivity and computation.

A. Dataset and Distributions

We use the Kaggle plant disease dataset recreated from PlantVillage [15], consisting of $87,867$ images across 38 classes. It includes 14 plant types, 13 diseases (some plant-specific), and one healthy class. Example images for 3 of the 14 plants are shown in Fig. 2. This dataset captures many real-world classification challenges, such as visually subtle distinctions between disease stages, varying lighting and background conditions, and class imbalance—factors that make accurate learning from limited, localized data difficult in practical deployments.

To simulate data heterogeneity, we sample training distributions for $N = 10$ nodes using a Dirichlet distribution parameterized by $\alpha \in \{0.1, 0.5, 100\}$. Smaller values of α yield partitions where each node observes only a few dominant classes, mimicking the effect of geographical or crop-specific locality. Conversely, larger α values approach an IID distribution, offering a basis for comparison under more favorable statistical conditions. This partitioning method is widely adopted as a principled way to explore the effects of controlled data skew across clients.

Fig. 3 visualizes the resulting class histograms across local datasets at each α setting. As heterogeneity increases (i.e., lower α), the distribution becomes sharply peaked, increasing the risk of local overfitting. This visualization highlights how much of the dataset's global diversity is lost at the node level—especially under $\alpha = 0.1$.

During training, images are resized to 256×256, then randomly cropped to 224×224 and horizontally flipped for data augmentation. They are then converted to tensors and normalized using ImageNet [16] statistics (mean: $[0.485, 0.456, 0.406]$, std: $[0.229, 0.224, 0.225]$). The test

Fig. 2: Sample images from the dataset, showing diseased (left) and healthy (right) leaves for apple, corn, and grape crops (top to bottom). The dataset exhibits natural variations in color, texture, and background, which can complicate visual diagnosis and highlight the need for robust learning approaches.

pipeline applies the same resizing and normalization but uses a center crop instead. This ensures consistent input dimensions while introducing variability during training to improve generalization.

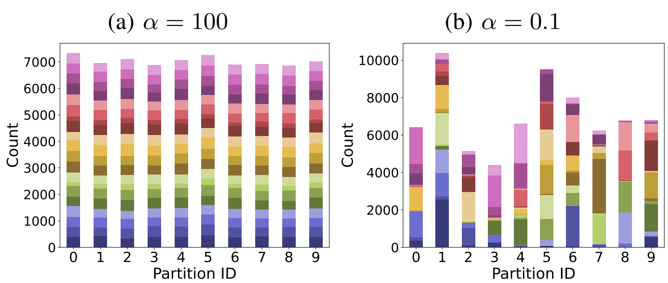

Fig. 3: Representative class distributions across nodes under Dirichlet sampling for two of three settings: $\alpha = 100$ (nearly IID) and $\alpha = 0.1$ (highly non-IID). Lower α values produce sharp class concentration at individual nodes, reflecting strong data heterogeneity.

B. Model Architecture

We use EfficientNetB0, pre-trained on ImageNet, as the feature extractor. It is a compact convolutional network that scales depth, width, and resolution in a balanced way, achieving strong accuracy-efficiency trade-offs [17]. Its final classification layer is replaced with an identity mapping, and its weights are frozen to produce fixed 1,280-dimensional feature vectors. This design is well-suited for edge deployment and allows us to focus on collaborative classifier refinement without confounding effects from feature learning. All nodes use the same frozen backbone to ensure architectural consistency across methods.

These feature vectors are used to train the classifier head collaboratively. In baseline methods, the head is a fully connected network with one hidden layer of 500 units. In contrast, our ensembling-based approach uses smaller classifier heads with 50-unit hidden layers per node, such that their combined capacity matches that of the larger baseline model. This configuration reflects practical constraints of mobile deployments while enabling a fair comparison between unified and modular strategies.

C. Modeling Dynamic Topology

To emulate opportunistic settings, we simulate a dynamic communication topology where node connectivity varies across time steps. This is controlled by a sparsity parameter ζ, with $\zeta = 1$ representing a fully connected graph and $\zeta = 0.1$ indicating that each node connects to only 10% of others at a given step. The topology is re-sampled each round, reflecting fluctuating peer availability over time. While this abstraction does not fully capture the structured nature of real-world mobility, it offers a principled way to probe robustness under varying levels of connectivity. Incorporating trace-driven topologies to reflect realistic spatial and temporal patterns remains an important direction for future work.

D. Evaluation

We evaluate the effectiveness of ensemble-based training across different data distributions, model configurations, and network conditions, as summarized in Table I. Each experiment runs for $T = 10$ time steps and is repeated five times with different random seeds.

TABLE I: Summary of Simulation Parameters

Parameter	Description	Value
$Dir(\alpha)$	concentration parameter	$\{100, 0.5, 0.1\}$
N	# of participating nodes	10
K	# of base-models	$\{1, 2, 5, 10\}$
T	time steps	10
η	learning rate (Adam)	0.001
ζ	network sparsity	$\{0.5, 0.25, 0.1\}$
n_h	hidden units	$\{500, 50\}$

We compare our approach against four baselines:

1) *Classical*: A single classifier is trained on the full dataset with global access to all samples. This serves as an upper performance bound.
2) *Local Learning*: Each node independently trains a full classifier on its private dataset, without any collaboration. This baseline reflects the most resource-efficient but isolated approach and is expected to struggle under non-IID data distributions.
3) *Centralized FL*: Nodes train local models on their own data and periodically send model updates to a central server for aggregation. This simulates conventional FL in ideal conditions with stable global coordination.
4) *Naive Opportunistic FL*: Nodes exchange full model parameters with neighbors during peer-to-peer encounters, but without any synchronization or aggregation protocol. This baseline simulates a direct adaptation of FL to opportunistic networks.

We do not include opportunistic momentum [8] in our baseline comparisons, as it primarily targets personalized model training, where each node optimizes for its own local objective. In contrast, our aim is to collaboratively construct a generalized model by aggregating diverse local knowledge. In all baselines, nodes train a full-sized classifier head with 500 hidden units, unlike the smaller heads used in our ensembling approach. All models are evaluated on a held-out global test set. We report standard classification accuracy averaged across five runs. Additionally, we report the accuracy of an oracle ensemble for our approach, constructed by aggregating the most recent versions of all trained base models. While not directly realizable during training—since nodes only maintain partial, asynchronously updated ensembles—the *oracle ensemble* serves as an upper bound on attainable performance. After training, delay-tolerant mechanisms can support dissemination of all base models, enabling nodes to progressively approximate this bound in practice.

V. Results and Discussions

This section presents the experimental results, starting with baseline performance and followed by an evaluation of our proposed approach under varying conditions.

A. Centralized and Local Baselines

Table II presents results for classical, local, and federated training. When all data is centrally available (Table II – Full Dataset), the model achieves 98.52% accuracy, highlighting the advantage of full accessibility. In contrast, local training—where each of the 10 nodes trains independently without collaboration—yields lower performance, with accuracy depending on data heterogeneity. For $\alpha = 100$, where partitions approximate IID, accuracy remains relatively high. However, as data heterogeneity increases, performance drops significantly. These results emphasize the impact of data skew on learning quality. While nodes can model their local data well, the lack of collaboration limits generalization.

TABLE II: Summary of Centralized and Baseline Results

Heterogeneity (α)	Classical / Local	Federated
Full Dataset	98.52%	–
100	95.59%	97.40%
0.5	74.83%	91.63%
0.1	39.09%	81.65%

Meanwhile, even under the most challenging conditions ($\alpha = 0.1$), federated learning achieves 81.65%, far outperforming the 39.09% of local training. This highlights the advantage of sharing model updates, enabling each node to learn from patterns beyond its own data. Across all heterogeneity levels, federated learning consistently outperforms local training, with the gap widening as α decreases. However, its strong performance relies on stable connectivity and centralized aggregation. As such, these results represent an idealized scenario and serve as a baseline for the subsequent experiments.

B. Opportunistic Federated Learning

We evaluate our ensembling approach alongside a naive opportunistic variant of federated learning (Naive-OppFL), which trains the full model and assumes that updates from all encountered neighbors can be integrated at each time step. This comparison highlights how each method adapts to decentralized learning under intermittent communication and varying local data distributions.

Testing accuracies are summarized in Table III. In high-connectivity ($\zeta = 0.5$) and near-IID conditions ($\alpha = 100$), our proposed method performs comparably to both Federated Learning and Naive-OppFL, despite relying on significantly smaller model exchanges. This suggests that when class distributions are balanced and peer availability is high, opportunistic ensembling of sequentially trained base models is sufficient to match the performance of traditional schemes, even without full-model synchronization or global aggregation.

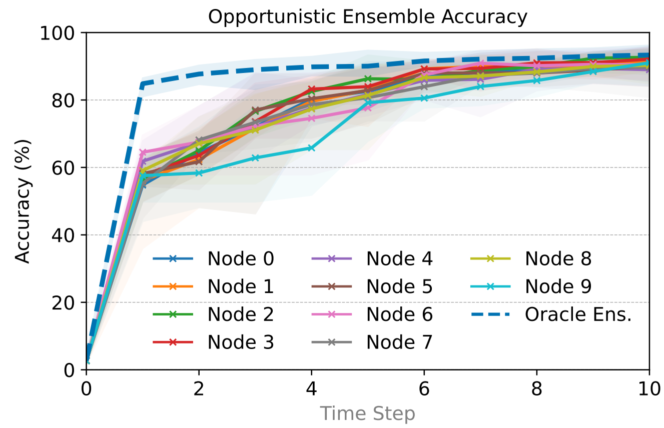

Fig. 4: Ensemble accuracy of the proposed method ($\alpha = 0.5$, $\zeta = 0.25$). Each colored curve with "×" markers shows the per-node mean accuracy of the local ensemble over time, with the shaded band indicating each node's min–max range. Local ensembles converge to oracle bounds, achieving near-optimal performance under heterogeneous, sporadic encounters.

TABLE III: Summary of Results for OppFL

	Naive-OppFL			Ensemble-OppFL		
ζ \ α	0.1	0.5	100	0.1	0.5	100
0.5	57.55%	85.75%	96.84%	81.33%	94.28%	97.29%
0.25	54.19%	85.21%	96.85%	81.93%	93.47%	97.10%
0.1	48.43%	83.45%	96.61%	77.54%	92.76%	96.42%

As heterogeneity increases, the performance gap between the two approaches becomes more pronounced. Under highly skewed data ($\alpha = 0.1$), our approach achieves up to 29% higher accuracy, consistently outperforming Naive-OppFL across all sparsity settings. This advantage stems from the robustness of the ensembling process, which mitigates the limitations of sequential training by integrating diverse, and sometimes conflicting, local perspectives. While base models may degrade as they move through nodes with drastically different distributions, their consolidation recovers lost representation and stabilizes overall performance. In contrast,

TABLE IV: Summary of Oracle Ensemble Accuracies for K-Ablation

ζ \ α	$K=1$ 0.1	$K=1$ 0.5	$K=1$ 100	$K=2$ 0.1	$K=2$ 0.5	$K=2$ 100	$K=5$ 0.1	$K=5$ 0.5	$K=5$ 100
0.5	55.14%	85.01%	94.96%	59.61%	84.27%	95.60%	71.19%	93.50%	97.40%
0.25	55.63%	81.98%	95.28%	58.52%	87.45%	95.40%	73.81%	91.27%	97.20%
0.1	50.93%	82.60%	94.36%	50.68%	80.90%	94.67%	66.93%	90.97%	96.80%

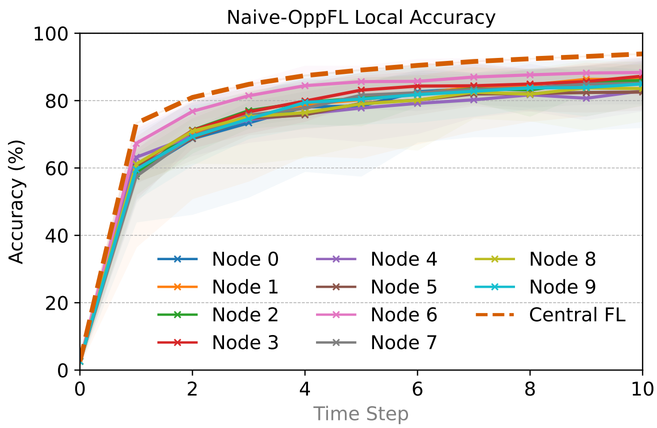

Fig. 5: Local-model accuracy under NaiveOppFL ($\alpha = 0.5$, $\zeta = 0.25$). Most nodes' standalone models remain below the centralized FL baseline (dashed line) and exhibit greater variability, highlighting how heterogeneous encounters and non-IID data degrade per-node model reliability.

Naive-OppFL relies on aggregating full-model updates from recent encounters without coordination or memory. As a result, it is highly sensitive to the local data distribution and the immediate neighborhood of each node. Fig. 4 and Fig. 5 offer a closer look at this behavior under moderate heterogeneity ($\alpha = 0.5$) and sparsity ($\zeta = 0.25$). In Ensemble-OppFL, local ensembles show a steady trajectory toward oracle accuracy as base models circulate. By contrast, Naive-OppFL exhibits more erratic behavior: some nodes improve if they encounter diverse peers, while others stagnate or regress due to redundant or unbalanced updates. These fluctuations reflect how peer diversity and exchange patterns affect local learning in the absence of coordination.

One trade-off of our approach is slower short-term performance. Since only one base model is exchanged at each step, local ensembles improve incrementally. As shown in Fig. 6, our method performs substantially fewer exchanges than Naive-OppFL across all sparsity levels. However, we find that the initial lag in accuracy is offset by more stable convergence and consistently reliable final performance. Allowing multiple exchanges per step could speed up ensemble saturation, but even with single-exchange rounds, the method consistently outperforms Naive-OppFL by the end of training. Interestingly, Fig. 4 also shows that if nodes were granted access to a wider set of base models early on, accuracy could improve more rapidly—suggesting that, under strict training budgets, prioritizing access to recent updates could offer an effective early stopping strategy. Finally, the combination of less frequent exchanges and smaller model size translates to a significantly lower communication burden.

These results underscore the limitations of naive FL adaptations in opportunistic settings. When connectivity is sparse and local data is highly skewed, mechanisms must go beyond standard aggregation to mitigate the risk of catastrophic model drift. Unfortunately, model alignment strategies such as [18]–[20] incur substantial computational overhead and often rely on stronger infrastructure assumptions. In addition, model exchange protocols should be lightweight and structured to minimize communication costs. By combining modular models with asynchronous, peer-driven refinement, our approach avoids these pitfalls and enables resilient decentralized learning in bandwidth-constrained environments.

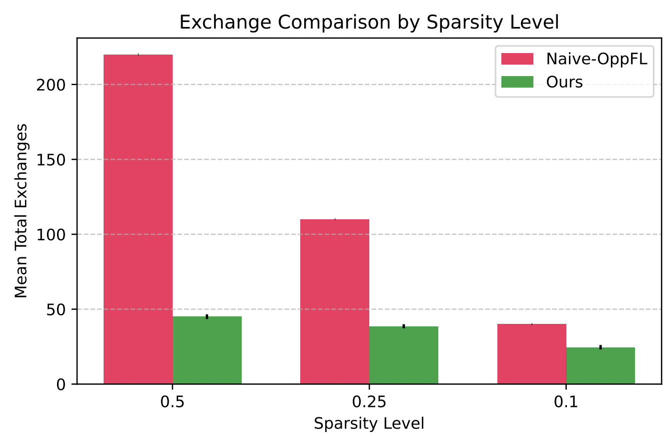

Fig. 6: Mean total model exchanges for Naive-OppFL vs. our ensemble method at different sparsity levels. By transmitting only compact base classifiers instead of full models, our approach slashes communication overhead—an effect that grows even more pronounced in denser topologies.

C. Impact of Varying Base Model Count and Capacity

Finally, we examine the impact of varying the number of base models K, while keeping the total model capacity constant across all configurations. When $K = 1$, the system relies on a single 500-unit classifier head, trained and passed sequentially between nodes. Increasing K divides this capacity across more lightweight base models—e.g., $K = 5$ corresponds to five classifiers with 100 units each—thus promoting modularity and allowing more nodes to participate concurrently.

As summarized in Table IV, larger K consistently yields better accuracy under higher data heterogeneity ($\alpha = 0.1$), reflecting the advantage of fine-grained ensembling when node-local datasets differ substantially. More base models allow diverse perspectives to be captured, helping the ensemble mitigate local bias. For instance, at $\zeta = 0.25$, accuracy improves from 55.63% ($K = 1$) to 73.81% ($K = 5$) under high heterogeneity. In contrast, under near-IID conditions ($\alpha = 100$), performance is already high even with a single model, and the gains from increasing K are minimal (e.g., from 94.96% to 97.40% at $\zeta = 0.5$). This suggests that when data is well-distributed across nodes, the expressivity of a monolithic model may be sufficient, and the benefit of ensembling diminishes.

These results suggest that the number of base models K can be used to tune multiple system-level trade-offs. Higher values of K enable greater node participation per round and allow better adaptation to data heterogeneity. Conversely, smaller values of K reduce memory and energy requirements and may accelerate convergence under near-IID conditions. The choice of K thus offers a practical mechanism for balancing efficiency, scalability, and generalization under varying resource and data distribution scenarios.

VI. Conclusion

This study investigated decentralized model training under highly constrained communication and data-sharing conditions. By introducing an opportunistic ensembling framework, we enabled mobile nodes to incrementally refine lightweight classifiers through transient peer-to-peer exchanges, without requiring synchronization or centralized orchestration. Our approach preserves model modularity and reduces communication overhead, while still supporting progressive integration of diverse local knowledge. Experimental results on a plant disease detection task confirm that the proposed method achieves high classification accuracy and demonstrates strong robustness to data heterogeneity and sparse connectivity, outperforming both local and naive federated baselines.

These findings suggest that ensembling offers a viable pathway for reliable machine learning in infrastructure-limited environments. The ability to train and share modular classifiers opportunistically can broaden the reach of collaborative intelligence, especially in settings where persistent connectivity or global coordination cannot be assumed. Future work will explore adaptive ensemble strategies, more dynamic exchange policies, and extensions to non-vision tasks and heterogeneous model architectures. Another direction is the evaluation of real-world deployments in mobile and rural networks, where the assumptions and trade-offs surfaced in this study may yield practical design insights.

References

[1] T. A. Shaikh, T. Rasool, and F. R. Lone, "Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming," *Computers and Electronics in Agriculture*, vol. 198, 2022.

[2] H. Chanyal and R. K. Yadav, "Machine learning-based plant disease detection for agricultural applications: A review," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2023.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

[4] S. Mehta, V. Kukreja, and R. Yadav, "A federated learning CNN approach for tomato leaf disease with severity analysis," in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2023.

[5] D. Mamba Kabala, A. Hafiane, and L. Bobelin, "Image-based crop disease detection with federated learning," *Sci Rep*, vol. 13, no. 19220, 2023.

[6] M. Aggarwal, V. Khullar, N. Goyal, A. Alammari, M. A. Albahar, and A. Singh, "Lightweight federated learning for rice leaf disease classification using non independent and identically distributed images," *Sustainability*, vol. 15, no. 16, 2023.

[7] P. V. Caminha and H. M. N. d. S. Oliveira, "Cloud-driven federated learning for plant disease detection in agriculture," in *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*. IEEE, 2024.

[8] S. Lee, X. Zheng, J. Hua, H. Vikalo, and C. Julien, "Opportunistic federated learning: An exploration of egocentric collaboration for pervasive computing applications," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021.

[9] H. Ochiai, Y. Sun, Q. Jin, N. Wongwiwatchai, and H. Esaki, "Wireless ad hoc federated learning: A fully distributed cooperative machine learning," *arXiv preprint arXiv:2205.11779*, 2022.

[10] "Fedtour: Participatory federated learning of tourism object recognition models with minimal parameter exchanges between user devices," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2022.

[11] Y. Deng and X. Yan, "Federated learning on heterogeneous opportunistic networks," in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. IEEE, 2024.

[12] "Towards opportunistic federated learning using independent subnetwork training," in *2024 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2024.

[13] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[14] "OPTIMIST: Opportunistic federated transfer learning for mobile devices using sequential independent subnetwork training," in *2025 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2025.

[15] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, Sep 2016.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[18] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[19] F. Yu, W. Zhang, Z. Qin, Z. Xu, D. Wang, C. Liu, Z. Tian, and X. Chen, "Fed2: Feature-aligned federated learning," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2066–2074.

[20] C. Liu, C. Lou, R. Wang, A. Y. Xi, L. Shen, and J. Yan, "Deep neural network fusion via graph matching with applications to model ensemble and federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 13 857–13 869.