Clustered Personalized Federated Crop Mapping for Sentinel-2 Crop-Type Time Series

Durganand Jha[1*] and Amandeep Kaur[1]

[1*]Department of Computer Science and Engineering, National Institute of Technology Delhi, Delhi 110040, India.

Contributing authors: 242211007@nitdelhi.ac.in; amandeepk@nitdelhi.ac.in;

Abstract

Federated Learning (FL), which allows the joint model training on distributed agricultural data sources without compromising data privacy, is a promising technology. However, the performance of FL is negatively impacted by high non-IID data heterogeneity, which is caused by the uneven distribution of various crops, management practices, and environmental conditions in different regions. In this paper, we explore the problem of federated crop-type classification on the TimeSen2Crop dataset, which is a large Sentinel-2 pixel-level benchmark with over one million samples on 16 crop types. The performance of the model varies greatly when the standard FedAvg algorithm is directly applied to the data. To mitigate the issue of non-IID data heterogeneity, we propose a framework called ”clustered personalized federated learning.” In the proposed framework, the clients are grouped based on their privacy-preserving signatures, which are computed from the label distribution histograms and the class-conditional feature prototypes learned from a lightweight temporal encoder. Finally, the federated optimization is performed on each cluster with a shared backbone net-work and adaptive heads. In the experiment, we simulate 100 clients and find that the proposed framework improves the macro-F1 and worst client macro-F1 compared to the standard FedAvg algorithm and local training.

Keywords: agricultural monitoring, client clustering, crop-type mapping, federated learning, personalized federated learning, remote sensing, Sentinel-2, time series classification

A. Agnihotri et al. (eds.), *Proceedings of the Conference on Bridging Engineering Disciplines with AI and Machine Learning (BEDAIML 2026)*, Advances in Intelligent Systems Research 209,
https://doi.org/10.2991/978-94-6239-697-5_14

1 Introduction

Crop type mapping from satellite images is an important application that enables agricultural monitoring and food security analysis on a large scale [1, 2]. Recent breakthroughs in deep learning and high-resolution Earth observation satellite data, such as Sentinel-2 time series, have improved crop classification performance at the parcel and pixel level [3, 4]. However, to train robust models, it is often necessary to collect data from multiple regions, organizations, or farmers, which may not be possible because of privacy, regulatory, or bandwidth issues.

Federated learning (FL) has been proposed as a new paradigm that facilitates collaborative global model training with distributed clients without exchanging raw data [5, 6]. In traditional FL, a central server manages the global model and facilitates iterative communication rounds involving local updates and parameter/model gradient aggregation from sampled clients [5]. Although FL is very appealing from the data privacy and governance point of view, it has been known to face challenges in non-IID data settings, which is particularly challenging in agricultural applications where data is highly non-IID due to regional climate, soil type, management practices, and crop patterns [7, 8].

The remote sensing datasets used for crop mapping, such as TimeSen2Crop and Sen4AgriNet, reflect this diversity over several regions and years [3, 4]. TimeSen2Crop, in particular, provides over 1.1 million pixel-level Sentinel-2 time series samples with 9 spectral bands and 365 daily time series observations per year, each labeled with one of 16 crop types and provided with quality indicators for cloud, snow, and shadow [3, 9]. Benchmarks of this scale offer a suitable playground for studying federated and personalized approaches to crop type classification.

This paper tackles the problem of non-IID client distributions in federated crop type mapping by introducing a clustered personalized FL approach specifically designed for multivariate time series classification on TimeSen2Crop. The main con-cept is to organize clients into clusters with similar label distributions and feature properties, and to train models for each cluster with a shared backbone and different heads. The goal is to mitigate the heterogeneity faced by each federated model while preserving the advantages of global knowledge transfer.

The main contributions are:

1. A realistic federated benchmark setup on TimeSen2Crop with 100 simulated clients to account for regions with non-IID crop distributions, facilitating the evaluation of FL algorithms for crop type mapping [3, 7].
2. A personalized FL framework that integrates histograms of label distributions and class-conditional feature prototypes into privacy-preserving client signatures for similarity-based clustering in FL [10–12].
3. A comprehensive empirical evaluation that shows the proposed method leads to better average and worst-client macro-F1 scores than FedAvg and local training, with ablations on the contribution of each component (histograms, prototypes, number of clusters) [7, 8].

2 Related Work

2.1 Federated Learning and Non-IID Data

The original FedAvg algorithm brought communication-efficient FL by leveraging local stochastic gradient descent (SGD) on each client and averaging at the server [5]. Later, the importance of FedAvg possibly converging slowly or to a suboptimal solution under strong non-IID data distributions among clients has been emphasized in various surveys and analyses [6, 7, 13]. Non-IID data distributions include label bias, feature distribution shift, and concept drift, and a detailed taxonomy and set of metrics have been developed to describe them [7].

To address non-IID data distributions, several techniques have been proposed, including regularization-based methods, proximal terms (such as FedProx), knowledge distillation, and local fine-tuning [6, 14]. Personalized FL aims to learn client models while sharing information across clients, typically through multi-task learning, meta-learning, or mixture-of-experts models [15, 16].

2.2 Client Clustering and Personalized FL

Client clustering-based FL algorithms cluster clients based on their similarity in model parameters, gradients, or performance metrics, and then conduct federated optimiza-tion on each cluster to match model training with the data structure [10–12]. For instance, FedSC carries out client-side clustering based on model parameters and trains models for each cluster to exchange information more efficiently among simi-lar clients [10]. Other studies leverage gradient information or loss vectors for client clustering and modify cluster assignments during the training process [11, 12].

Personalized federated learning with adaptive or hierarchical aggregation enables adaptation for clients or clusters while maintaining a global knowledge base [17, 18]. These studies have demonstrated effectiveness in difficult heterogeneous settings but have not been extensively investigated in the context of large-scale remote sensing time series data for agriculture.

2.3 Federated Learning in Remote Sensing and Agriculture

Federated learning has recently gained attention in remote sensing for applications such as scene classification, land cover mapping, and crop yield estimation [19, 20]. A survey on FL applications in remote sensing points out the advantages of FL in terms of data privacy and inter-agency collaboration, but also the difficulties related to large domain shifts between sensors, seasons, and regions [19].

In the agricultural field, federated learning has been considered for crop yield estimation on farms and for smart farming networks, but the literature on federated crop-type classification from satellite time series remains relatively scarce [8, 20, 21]. TimeSen2Crop and Sen4AgriNet are datasets that offer large-scale labeled Sentinel-2 time series and multi-year and multi-country benchmarks, which are well-suited for testing FL algorithms in realistic conditions of heterogeneity [3, 4]. However, most of the current literature relies on centralized training on these datasets.

3 Preliminaries

3.1 Federated Learning Setup

Consider a set of N clients indexed by $i \in \{1, \ldots, N\}$, each holding a local dataset $D_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$, where x_{ij} is a time series sample and y_{ij} its crop-type label. The global objective in standard FL is to minimize a weighted sum of local empirical risks:

$$\min_{w} F(w) := \sum_{i=1}^{N} p_i F_i(w), \tag{1}$$

where $p_i = \frac{n_i}{\sum_{k=1}^{N} n_k}$ and $F_i(w) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f_w(x_{ij}), y_{ij})$, with f_w denoting a model parameterized by w and ℓ a loss function (e.g., cross-entropy) [5].

In FedAvg, training proceeds in communication rounds $t = 0, 1, \ldots, T-1$. At round t, a subset of clients $S_t \subset \{1, \ldots, N\}$ is sampled, each client $i \in S_t$ performs local SGD starting from the current global model w^t, and the server aggregates updates as:

$$w^{t+1} = \sum_{i \in S_t} \frac{n_i}{\sum_{k \in S_t} n_k} w_i^{t+1}, \tag{2}$$

where w_i^{t+1} is the locally updated model at client i [5].

3.2 Non-IID Client Distributions

Non-IID data across clients can be represented by differences in the conditional label distributions $P_i(y \mid x)$, the marginal input distributions $P_i(x)$, or concept drift over time [7]. In label-skew settings, clients have access to mutually exclusive or very imbalanced label sets, and in feature-skew settings, the same labels can represent different spectral–temporal patterns. In agriculture, both occur: some areas are known for specific crops, and agro-climatic conditions affect the phenological curves of similar crop types [8].

Clustered personalized FL tries to mitigate the adverse effects of non-IID data by grouping clients into clusters where the distributions are more similar, thus making the optimization problem easier for each cluster [10, 11].

3.3 TimeSen2Crop Dataset

TimeSen2Crop is a pixel-level Sentinel-2 image time series dataset for crop type classification, which contains over one million labeled samples [3]. Each sample is a multivariate time series with a length of 365 (daily data over an agronomic year) and 9 spectral bands, along with labels for 16 crop types and quality indicators (cloud, snow, and shadow) [3, 9]. The dataset also includes time series data for the following agronomic year to support research on temporal transfer and domain shift [3].

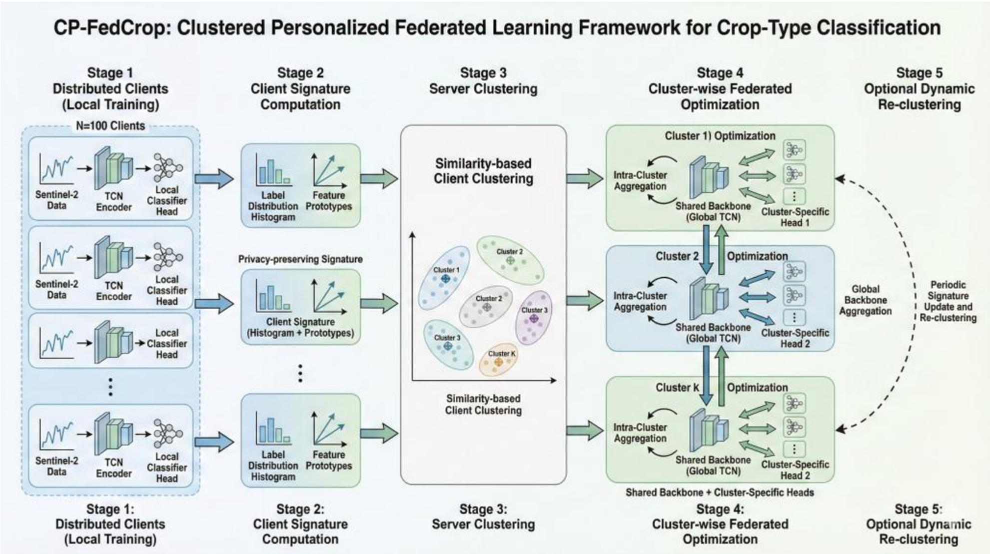

Fig. 1 Overview of the proposed Clustered Personalized Federated Crop Mapping (CP-FedCrop) framework. Distributed clients compute label histograms and feature prototypes to form privacy-preserving signatures, and the server performs similarity-based clustering followed by cluster-wise federated optimization.

In this study, the main experiments are conducted on the main agronomic year data, with clients built to model regional diversity in crop type and time series pat-terns. The dataset is published through an open-access repository, which supports the reproducibility of the proposed federated benchmark and learning solutions [3, 9].

4 Methodology

4.1 Overview

The proposed framework, called *Clustered Personalized Federated Crop Mapping* (CP-FedCrop), as shown in Fig. 1, extends the conventional FL framework by incorporating a client clustering step based on privacy-preserving signatures obtained from the local data. The process involves the following steps:

1. Client signature computation: Each client computes a label distribution histogram and class-conditional feature prototypes from its local data.
2. Client clustering: The server aggregates signatures and performs clustering (e.g., k-means) in signature space to obtain K clusters.
3. Cluster-wise federated optimization: For each cluster, a federated model is trained with a shared backbone and a cluster-specific head.
4. Optional re-clustering: Signatures can be updated and re-clustering performed periodically to adapt to the changing representations [11].

4.2 Client Signatures

Label-distribution histograms.

For client i, define the empirical label distribution vector $h_i \in \mathbb{R}^C$, where C is the number of crop types (here $C = 16$):

$$h_i(c) = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{1}_{\{y_{ij}=c\}}, \quad c \in \{1, \ldots, C\}. \tag{3}$$

The histogram h_i captures label skew and provides a simple, low-dimensional representation of the crops present at client i [7].

Feature prototypes.

Let $g_\theta(\cdot)$ denote a temporal encoder parameterized by θ that maps an input time series $x \in \mathbb{R}^{T\times B}$ (with length T and B spectral bands) to a feature vector $z \in \mathbb{R}^d$. Each client computes class-conditional prototypes by averaging features over its local samples:

$$p_i(c) = \frac{1}{n_i(c)} \sum_{j:y_{ij}=c} g_\theta(x_{ij}), \quad c \in \mathcal{C}_i, \tag{4}$$

where $\mathcal{C}_i = \{c : n_i(c) > 0\}$ is the set of classes present at client i, and $n_i(c)$ is the number of local samples with label c. Missing classes at a client are handled by zero vectors or a learned default prototype. The feature prototypes capture the spectral–temporal characteristics of crops at each client and, when combined with histograms, provide a richer description of client-specific data than label distributions alone [10, 12].

Signature vector.

The client signature s_i is constructed by concatenating the label histogram h_i with a compressed representation of the prototypes, e.g., by applying a linear mapping or pooling:

$$s_i = \phi\left(h_i, \{p_i(c)\}_{c=1}^{C}\right) \in \mathbb{R}^D, \tag{5}$$

where ϕ can be implemented as a simple flatten-and-project operation followed by normalization. To limit communication cost and preserve privacy, prototypes may be projected to a lower dimension via a fixed random projection or a learned bottleneck layer [13].

4.3 Client Clustering

Given the set of signatures $\{s_i\}_{i=1}^N$, the server applies a clustering algorithm (e.g., k-means) to partition clients into K clusters $\{\mathcal{G}_k\}_{k=1}^K$:

$$\{\mathcal{G}_1, \ldots, \mathcal{G}_K\} = \mathrm{Cluster}\left(\{s_i\}_{i=1}^N, K\right). \tag{6}$$

For k-means, this involves minimizing the within-cluster variance:

$$\min_{\{\mu_k\},\{G_k\}} \sum_{k=1}^{K} \sum_{i \in G_k} \|s_i - \mu_k\|_2^2 \tag{7}$$

where μ_k is the centroid of cluster k. The number of clusters K is treated as a hyper-parameter and selected based on validation performance or silhouette analysis. A moderate value of K provides a trade-off between global generalization (small K) and fine-grained personalization (large K) [10, 12].

4.4 Cluster-wise Federated Optimization

For each cluster G_k, a cluster-specific model is trained via federated optimization. A common design is to decompose the model into a shared backbone g_θ and a cluster-specific head h_{ψ_k}, such that:

$$f_k(x) = h_{\psi_k}\left(g_\theta(x)\right). \tag{8}$$

The backbone parameters θ may be shared across all clusters and updated using aggregated gradients or parameters from all clusters, while the heads $\{\psi_k\}_{k=1}^{K}$ are updated using only data from their respective clusters [17, 18].

One concrete training scheme is:

1. Initialize θ^0 and ψ_k^0 for all k.
2. For each round t:
   (a) Sample clusters and clients in the clusters, for example, sample a set of clusters and then sample a set of clients in each cluster.
   (b) Each client performs local training on f_k with the head of its cluster and the shared backbone network.
   (c) For each cluster k, aggregate the client updates in G_k to get the updated ψ_k.
   (d) Aggregate the updates of the backbone network over all clusters to get the updated θ.

Formally, the cluster-wise objective can be written as:

$$\min_{\theta,\{\psi_k\}} \sum_{k=1}^{K} \sum_{i \in G_k} p_i F_i(\theta, \psi_k). \tag{9}$$

This formulation allows the backbone to capture global patterns shared across regions, while cluster-specific heads specialize in regional label distributions and feature characteristics [18].

4.5 Optional Dynamic Re-clustering

As the training continues, the encoder function g_θ changes, and the new feature prototypes can potentially provide new signatures. To incorporate this dynamic process, the signatures can be updated periodically (for example, every R rounds), and the clustering can be updated based on the new signatures. This provides a dynamic clustering process, where the clients can potentially move between the different clusters, analogous to recent works on client clustering with migration in FL [11]. However, there is also an added complexity with the dynamic clustering process, which requires additional communication. In this work, both static and dynamic clustering methods are explored in the ablation experiments.

5 Experimental Setup

5.1 Dataset and Preprocessing

The TimeSen2Crop dataset is employed as the primary dataset [3, 9]. Every example is provided with a 365-day Sentinel-2 time series with 9 spectral bands and a crop type label from 16 possible classes. As per the dataset protocol, examples with permanent cloud, snow, or shadow can be removed, or missing data can be treated using masking and interpolation [3].

For this analysis, the primary agricultural year is divided into training, validation, and test sets based on spatial and temporal partitions suggested in the dataset documentation [3]. The secondary year is kept aside for possible future work on temporal transfer learning tasks and is not the focus of this analysis.

5.2 Client Construction and Non-IID Partitioning

For simulating a federated setting, $N = 100$ clients are created by dividing the dataset into region-like subsets with varying label distributions. First, the geographic indices or tiles included in the dataset are used as a starting point for grouping, and then samples within each tile are divided into several clients with varying crop distributions using a Dirichlet distribution on labels to introduce label skew, as typically done in non-IID FL benchmarking [7]. This leads to a federated benchmark with the following properties:

1. Each client has a few thousand to tens of thousands of samples.
2. Some clients have a strong dominance of a few crops, while others are more balanced.
3. The input–label joint distributions are very different across clients.

This is a realistic representation of heterogeneity in agricultural regions and a challenging non-IID FL setting [8].

5.3 Model Architecture

The local model is composed of a temporal encoder g_θ and a classification head h_ψ. The encoder is a Temporal Convolutional Network (TCN) with dilated 1D convolutions along the temporal axis and residual connections, which has been shown to be effective for time-series classification with low computational complexity [22]. The head is a two-layer multilayer perceptron (MLP) with ReLU activation and a softmax output over the 16 classes.

For cluster-wise personalization, the backbone network g_θ is shared among all clusters, while each cluster has its own classification head h_{ψ_k} as shown in (8). Other architectures, like gated recurrent units (GRUs) or Transformers, are considered for future work.

5.4 Baselines and Ablations

The following baselines are considered:

1. **Centralized**: a single model is trained on the union of all client data (upper bound).
2. **Local**: each client trains a model independently without any collaboration (lower bound).
3. **FedAvg**: federated averaging [5] applied to all clients without clustering.

The proposed clustered personalized FL approach is compared in the following variants:

1. **CP-FedCrop (H)**: clustering based solely on label histograms h_i.
2. **CP-FedCrop (P)**: clustering based solely on feature prototypes p_i.
3. **CP-FedCrop (H+P)**: clustering based on combined signatures s_i that include both histograms and prototypes.
4. **CP-FedCrop (Dyn)**: dynamic re-clustering every R rounds.

5.5 Training Protocol

Training continues for T rounds of communication. During each round, a set of clients is chosen (for example, 10% of the clients for a round), and each client performs a predetermined number of local epochs of mini-batch SGD. The learning rate, batch size, number of local epochs, and number of rounds are set by tuning on the validation set. The training procedure for the clustered versions is identical, except that aggregation is done separately for the cluster-specific heads and the shared backbone network.

5.6 Evaluation Metrics

The key evaluation criteria are:

1. **Macro-F1**: the unweighted average of the per-class F1 scores, to handle class imbalance.
2. **Overall Accuracy**: the proportion of correctly labeled instances.
3. **Worst-client Macro-F1**: the worst macro-F1 score across clients, to evaluate fairness for clients.
4. **Communication cost**: calculated as the total number of bytes communicated over all communication rounds [13].

Macro-F1 and worst-client Macro-F1 are evaluated on the test set, averaged over several runs with different random seeds.

Table 1 Performance Comparison on TimeSen2Crop (100 Clients) for a non-IID federated partition.

Method	Macro-F1 (%)	Accuracy (%)	Worst-client Macro-F1 (%)	Comm. cost
Centralized	86.4	92.1	–	–
Local	59.3	75.8	21.4	0.0
FedAvg	72.6	86.3	41.7	1.0
CP-FedCrop (H)	75.1	87.8	49.3	1.1
CP-FedCrop (P)	76.4	88.5	53.6	1.1
CP-FedCrop (H+P)	78.2	89.7	60.8	1.2
CP-FedCrop (Dyn)	79.4	90.3	64.5	1.3

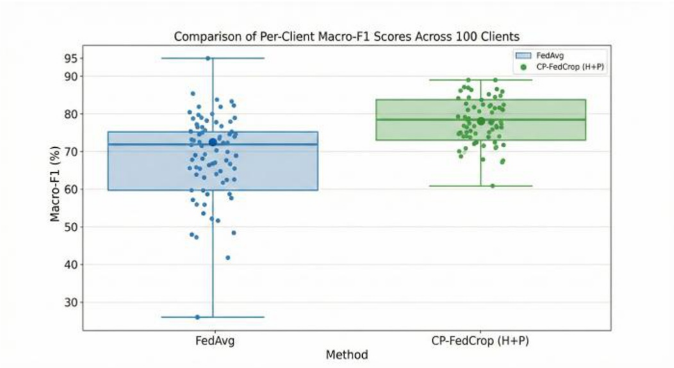

Fig. 2 Distribution of per-client Macro-F1 scores for FedAvg and CP-FedCrop (H+P) on Time-Sen2Crop (placeholder figure).

6 Results

6.1 Overall Performance

Table 1 reports the test performance of the baselines and the proposed methods on TimeSen2Crop.

The CP-FedCrop variants are expected to perform better than FedAvg on both Macro-F1 and worst-client Macro-F1 metrics while incurring only a small additional communication cost because of the transmission of signatures and cluster-specific heads [10, 11].

6.2 Impact of Client Clustering

To evaluate the effect of client clustering, the distribution of per-client Macro-F1 scores is considered for FedAvg and CP-FedCrop (H+P). Figure 2 shows that client clustering leads to a reduction in variance and better performance for the worst-performing clients, suggesting improved fairness [7].

Table 2 Ablation on client signatures and number of clusters K.

Variant	K	Macro-F1 (%)	Worst-client Macro-F1 (%)
FedAvg (no clustering)	–	72.6	41.7
CP-FedCrop (H)	5	74.3	46.8
CP-FedCrop (H)	10	75.1	49.3
CP-FedCrop (P)	5	75.6	50.9
CP-FedCrop (P)	10	76.4	53.6
CP-FedCrop (H+P)	5	77.1	57.2
CP-FedCrop (H+P)	10	78.2	60.8

Qualitatively, clients with highly skewed label distributions or rare crop types benefit the most from cluster-specific heads, while balanced clients maintain performance close to the centralized upper bound.

6.3 Ablation on Signatures and Number of Clusters

An ablation study is conducted to assess the contribution of label histograms and feature prototypes to clustering quality and downstream performance. Table 2 summarizes the results.

The combined signatures (H+P) with a moderate number of clusters (e.g., $K = 10$) are expected to yield the best trade-off between overall performance and fairness, as they capture both label skew and feature-level differences among clients [10, 12].

6.4 Communication and Computation Analysis

The communication overhead due to CP-FedCrop comes from the transmission of client signatures and the cluster-specific heads. The overhead due to the transmission of client signatures is low because the signatures are of low dimension and are transmitted less frequently (for example, only at initialization or every R rounds). The overhead due to cluster-specific heads is also low because the heads are smaller compared to the backbone network [13]. The communication overhead analysis (normalized to FedAvg) is provided in Table 1 and shows that the communication overhead is close to 1.0, which is efficient.

7 Discussion

The experimental outcomes show that similarity-aware clustering of clients in FL can effectively mitigate the adverse effect of non-IID data in crop type mapping using remote sensing images. By incorporating heterogeneity modeling through label histograms and feature prototypes, CP-FedCrop can more effectively match model specialization with regional characteristics and crop distribution [8, 10].

However, there are some limitations and future research directions. First, the present work is based on a single dataset and single-year analysis; multi-year and multi-sensor analysis, including time shift from agronomic years, would be more realis-tic [3, 4]. Second, privacy preservation for signatures can be improved using differential privacy techniques or secure aggregation protocols during prototype transmission [13]. Third, more advanced clustering strategies (e.g., spectral clustering, mixture models) and adaptive determination of the number of clusters could provide further improvements [12].

8 Conclusion

This paper introduced a clustered personalized federated learning framework for crop type classification on Sentinel-2 time series data in the TimeSen2Crop dataset. By building a realistic non-IID federated benchmark with 100 clients and incorporating privacy-preserving client signatures that integrate label histograms and feature prototypes, the proposed approach synchronizes federated model training with the data heterogeneity in agricultural data. Experimental results demonstrated improvements in both average and worst-client performance compared to standard FedAvg and local training with modest communication overhead [7, 8]. These results indicate that clustered personalized FL is a promising research avenue for applying federated crop mapping systems that need to work in different regions and for different stakeholders while considering data sovereignty constraints. Future research will be conducted to generalize the framework to multi-year and multi-modal remote sensing data and explore stronger privacy guarantees for client signatures [3, 4].

Acknowledgment

The authors would like to thank the maintainers of the TimeSen2Crop dataset for providing an open benchmark for remote-sensing crop-type classification [3].

References

[1] Belgiu, M., Csillik, O.: Crop mapping using sentinel-2 imagery: A review. International Journal of Applied Earth Observation and Geoinformation **73**, 107–121 (2018) https://doi.org/10.1016/j.jag.2018.05.009

[2] European Commission Copernicus: Sentinel-2 for Agriculture. [Online]. Available: https://www.copernicus.eu/en/sentinel-2-agriculture. Accessed: Feb. 16, 2026.
(2014)

[3] Weikmann, G., Paris, C., *et al.*: Timesen2crop: A million labeled samples dataset of sentinel-2 image time series for crop-type classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 4825–4840 (2021) https://doi.org/10.1109/JSTARS.2021.3073965

[4] Interdonato, R., *et al.*: Sen4agrinet: A sentinel-2 multi-year, multi-country benchmark dataset for crop classification and segmentation with deep learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 5894–5908 (2021) https://doi.org/10.1109/JSTARS.2021.3090003

[5] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), pp. 1273–1282 (2017)

[6] Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Trans. Knowl. Data Eng. **35**(4), 3347–3366 (2023) https://doi.org/10.1109/TKDE.2021.3124599

[7] Xie, Y., et al.: Non-iid data in federated learning: A survey with taxonomy, met-rics, methods, frameworks and future directions. arXiv (2024). arXiv:2411.12377

[8] Samarasinghe, S., *et al.*: A review of federated learning in agriculture. Computers and Electronics in Agriculture **211**, 107999 (2023) https://doi.org/10.1016/j.compag.2023.107999

[9] MONSTER Monash Time Series Repository: TimeSen2Crop Dataset on Hugging Face. [Online]. Available: https://huggingface.co/datasets/monster-monash/TimeSen2Crop. Accessed: Feb. 16, 2026. (2025)

[10] Zhang, W., *et al.*: Fedsc: A federated learning algorithm based on client-side clustering. Electronic Research Archive **31**(3), 1251–1273 (2023) https://doi.org/10.3934/era.2023266

[11] Wang, P., Lu, S., *et al.*: Client clustering and migration in federated learning via gradient similarity. In: Proc. 34th Int. Joint Conf. Artificial Intelligence (IJCAI) (2025)

[12] Bhatia, R., Papadis, N., et al.: Clove: Personalized federated learning through clustering of loss vector embeddings. arXiv (2025). arXiv:2506.22427

[13] Kairouz, P., McMahan, H.B., *et al.*: Advances and open problems in federated learning. Foundations and Trends in Machine Learning **14**(1–2), 1–210 (2021) https://doi.org/10.1561/2200000083

[14] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proc. 3rd MLSys Conf. (2020)

[15] Smith, V., Chiang, C.-K., Sanjabi, M., Talwalkar, A.: Federated multi-task learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)

[16] Arivazhagan, M.G., Aggarwal, V., Singh, A., Choudhary, S.: Federated learning with personalization layers. In: Proc. NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality (2019)

[17] Liu, X., *et al.*: Personalized federated learning algorithm with adaptive aggregation. Applied Intelligence **53**(13), 15476–15494 (2023) https://doi.org/10.1007/s10489-023-04512-9

[18] Reddy, S.P.K., *et al.*: Provably personalized and robust federated learning. J. Mach. Learn. Res. **23**, 1–39 (2022)

[19] Moreno-Álvarez, S., Paoletti, M.E., Sanchez-Fernandez, A.J., Rico-Gallego, J.A., Han, L., Haut, J.M.: Federated learning meets remote sensing. Expert Sys-tems with Applications **255**, 124583 (2024) https://doi.org/10.1016/j.eswa.2024.124583

[20] Zhang, L., *et al.*: Federated learning for crop yield prediction. Computers and Electronics in Agriculture **218**, 108859 (2025)

[21] Bora, S., *et al.*: Crop classification in a smart farm decentralised network. Computers and Electronics in Agriculture **209**, 107892 (2023)

[22] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. In: Proc. 35th Int. Conf. Machine Learning (ICML) (2018)