Serverless Federated Learning in Multi-Cell Scenarios Robust To Test-Time Distribution Shifts

Kwanyoung Cho†, Jongyul Park†, Atif Rizwan and Minseok Choi
Department of Electronic Engineering, Kyung Hee University, Yongin, South Korea
E-mails: chaile99@khu.ac.kr, tubduk@khu.ac.kr, atifrizwan@khu.ac.kr, choims@khu.ac.kr
† These authors contributed equally.

***Abstract*—This paper introduces a novel serverless Federated Learning (FL) framework designed for multi-cell environments, addressing the significant limitations of traditional centralized FL methods, especially in dynamic settings with frequent test-time distribution shifts. Unlike conventional FL approaches that rely on a central cloud server, our framework decentralizes model training and aggregation entirely to edge servers, thereby reducing communication latency, improving scalability, and enhancing data privacy. The proposed framework incorporates additional hyperparameters of α and β to control the tradeoff between personalization and generalization for the model aggregation process, while ensuring robust performances in out-of-distribution (OOD) tasks. Furthermore, the framework exhibits adaptability across varying numbers of clients in overlapping regions, making it a scalable and effective solution for real-world FL applications in environments with diverse and unreliable connectivity. Experimental results on the CIFAR-10 dataset demonstrate that our framework outperforms existing serverless and centralized FL methods, achieving superior personalization and accuracy even under challenging conditions.**

***Index Terms*—Federated learning, Split learning, Personal model training**

I. Introduction

Traditional deep learning (DL) algorithms have predominantly employed a centralized learning (CL) architecture, wherein a central server collects users' data for model training. While this methodology has been instrumental in developing high-performance models, it is accompanied by significant limitations, particularly in terms of communication cost and privacy concerns [1]. The substantial communication overhead and the risk of data breaches inherent in transferring vast amounts of personal data to a centralized server pose serious challenges for real-world applications [2]. To overcome these limitations, distributed learning approaches have been proposed, such as training models directly on user devices or utilizing edge servers to distribute the computational load [3]. With advancements in device capabilities, local training has emerged as a promising alternative, enabling on-device model training and thereby addressing the challenges of data transmission and privacy [4].

Federated Learning (FL) has emerged as a viable approach for enabling distributed model training by uploading only model updates, rather than raw data, to a central server, thereby protecting privacy and significantly reducing communication overhead [5]. Despite these advantages, the traditional FL framework, which often relies on a central server for aggregating model updates, introduces new challenges, such as communication bottlenecks and transmission delays caused by millions of devices connecting to a single server and unreliable communication conditions. To mitigate these issues, hierarchical federated learning (HierFAVG) has been proposed [7], leveraging multiple edge servers in conjunction with a cloud server to perform multi-level aggregation of models before generating a final global model.

Despite the advancements introduced by HierFAVG, the dependence on a central cloud server still creates communication bottlenecks on the core network when implementing FL in hierarchical wireless networks. This limitation has driven research toward serverless FL as a viable alternative [8], [9]. This serverless architecture reduces communication latency and costs, compared to HierFAVG, in environments with limited or unreliable cloud connectivity. However, the existing serverless FL approaches have focused on the global model training without adequately addressing the challenges posed by test-time distribution shifts, which are prevalent in real-world scenarios [6]. Additionally, although several studies proposed the novel FL algorithms which personalize the model while maintaining the generalization capability at a certain level [10], these methods frequently depend on cloud-based aggregation, thereby constraining their applicability in fully serverless environments [11].

This study proposes a serverless FL framework that introduces key enhancements for improved personalization and robustness to test-time distribution shifts. By introducing additional hyperparameters for the model aggregation process, our framework achieves high accuracy in personalized tasks while handling out-of-distribution (OOD) tasks at a certain level. Unlike traditional methods that rely on cloud-based aggregation, our approach fully leverages the serverless FL architecture for the multi-cell system, thereby enhancing the scalability and adaptability of edge models in diverse and dynamic environments. This research specifically addresses the challenges posed by dynamic, real-world settings where test-time distribution shifts are prevalent.

II. System Model

A. Federated Learning in Multi-Cell Scenarios

HierFAVG [7] is an advanced extension of traditional FL, designed to address the communication and scalability challenges in large-scale distributed learning systems. In conven-

tional FL, a central cloud server aggregates models trained locally on clients' devices, however, this centralized approach introduces significant communication overheads and latency, particularly in geographically expansive environments. HierFAVG mitigates these challenges by introducing an intermediate layer of edge servers (ESs) between clients and the cloud server. This hierarchical structure reduces the communication load on the cloud server, speeding up the training particularly in environments with a large number of clients.

Despite these advantages, HierFAVG's reliance on a central cloud server still presents potential bottlenecks in communication and privacy risks. To overcome these limitations, we leverage a serverless FL framework in the multi-cell system with overlapping areas, FedMes [9]. This serverless architecture not only reduces communication latency but also enhances scalability by keeping all data and model updates localized at the edge. The elimination of the cloud server also simplifies the system architecture, making it more robust and adaptable to environments with limited or unreliable cloud connectivity. Also, FedMes leverages handover regions in the wireless cellular system to make cooperation among adjacent cells even without collaborations among ESs.

Suppose that there are M ESs and K clients, and each client k has a local dataset D_k. The ES and client sets are defined as $\mathcal{M} \triangleq \{1, \cdots, M\}$ and $\mathcal{K} \triangleq \{1, \cdots, K\}$, respectively. In the coverage region of each ES i, there are K_i clients and their set is $\mathcal{K}_i \triangleq \{1, \cdots, K_1\}$. The nonoverlapped area of ES i and its area overlapped with neighboring ESs are denoted by $\mathcal{N}_i$ and $\mathcal{O}_i$, respectively. Also, the set of ESs associated with the clients in the overlapped area, i.e., clients $k \in \cup_{i=1}^{M} \mathcal{O}_i$, is denoted by $\mathcal{S}_k$. The goal of FL is to train a global model $\mathbf{w}^*$ which minimizes the following global loss function:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \min_{\mathbf{w}} \sum_{k=1}^{K} \frac{|D_k|}{\sum_{j=1}^{K} |D_j|} F_k(\mathbf{w}), \tag{1}$$

where $F_k(\mathbf{w})$ is the local loss function of client k, defined as $F_k(\mathbf{w}) = \frac{1}{|D_k|} \sum_{(x,y) \in D_k} \ell(f_{\mathbf{w}}(x), y)$. Here, $\ell(\cdot, \cdot)$ represents the loss function (e.g., cross-entropy loss), and $f_{\mathbf{w}}(x)$ is the model's prediction with $\mathbf{w}$ for input x whose label is y.

B. Personalization and Test-Time Distribution Shifts

In this paper, each cell has a specific primary task that is closely aligned with the personalized needs of the clients within that cell. Our approach ensures that the edge models are highly optimized for the main task, delivering a tailored experience that effectively meets the specific requirements of the clients. However, this strong focus on personalization could introduce potential vulnerabilities in OOD tasks. In other words, the limited number of classes can be found in the coverage area of a single ES, and in particular, the clients in the overlapped area can have the training data with any classes among the primary classes of all of the neighboring ESs. For instance, consider a scenario where cell i is primarily responsible for classifying images labeled $\{0, 1, 2\}$, while an adjacent cell j specializes in labels $\{3, 4, 5\}$. In the overlapping region between cell i and cell j, a client's primary classes are originally from cell j could move into cell i, and then the tasks related to labels $\{3, 4, 5\}$ will appear in cell i. Consequently, the edge model of ES i, which has been primarily optimized for labels $\{0, 1, 2\}$, may struggle to provide accurate inferences for this OOD data.

This situation is further complicated in dynamic environments where user mobility is common, and clients frequently move between cells with different main tasks. Without appropriate adjustments, the edge model could fail to generalize well to these unexpected tasks, leading to a degradation in performance. In this regard, we consider the test scenario in which the majority of test samples are from the primary classes and the remaining samples have the OOD classes. We denote the ratio of the number of test samples with primary tasks to the total size of the test dataset by ρ.

III. Serverless Federated Learning in Multi-Cell Scenarios

In this section, we describe the detailed descriptions of our proposed serverless FL algorithm in multi-cell scenarios. This framework extends the FedMes [9] algorithm by introducing additional hyperparameters to improve the personalization capability and robustness to test-time distribution shifts.

A. Local Training

At the beginning of the FL algorithm, we suppose that the local models of all clients are identically initialized, i.e., $\mathbf{w}_k(0) = \mathbf{w}^0$ for all $k \in \mathcal{K}_i$ and $i \in \mathcal{M}$, where $\mathbf{w}_k^{(i)}(t)$ is the local model of client k in the region of ES i at local epoch t. Then, each client performs the local training process for E epochs which minimizes the local loss function, as follows:

$$\mathbf{w}_k^{(i)}(t+1) \leftarrow \mathbf{w}_k^{(i)}(t) - \eta_t \nabla F_k(\mathbf{w}_k^{(i)}(t)), \tag{2}$$

where η_t is the learning rate at epoch t.

B. Model Aggregation

After completing E local updates, each client k sends its locally updated model, i.e., $\mathbf{w}_k^{(i)}(E)$, back to the corresponding ES. Once the updated models are received from all clients, the ES performs the model aggregation process with the additional hyperparameter α, as given by

$$\begin{aligned} \bar{\mathbf{w}}^{(i)}(t) \leftarrow & \frac{1}{(1+\alpha)} \sum_{k \in \mathcal{N}_i} \frac{|\mathcal{N}_i|}{|\mathcal{N}_i| + |\mathcal{O}_i|} \mathbf{w}_k^{(i)}(t) \\ & + \frac{\alpha}{(1+\alpha)} \sum_{k \in \mathcal{O}_i} \frac{|\mathcal{O}_i|}{|\mathcal{N}_i| + |\mathcal{O}_i|} \mathbf{w}_k^{(i)}(t), \end{aligned} \tag{3}$$

for every E local epochs, i.e., $t \mid E = 0$, where $\bar{\mathbf{w}}^{(i)}(t)$ is the aggregated model of ES i after t local epochs. For clients in non-overlapping regions, the ES assigns a weight of $\frac{1}{1+\alpha}$ to the model updates, where α is a parameter that reflects the relative importance of models from overlapping regions. Conversely, for clients in overlapping regions, the ES assigns a weight of $\frac{\alpha}{1+\alpha}$ to the received model updates. This approach ensures that the non-overlapping regions, which are more representative

of the cell's primary task, exert a stronger influence on the aggregated edge model, while still incorporating essential information from the overlapping regions to maintain robustness against OOD tasks.

C. Initial Model Setting

For the next communication round, the parameter server of the traditional FL broadcasts the aggregated model to all clients; however, in our algorithm, the clients in the nonoverlapped area only set their local modes by the aggregated model, i.e., $\mathbf{w}_k^{(i)}(t) \leftarrow \bar{\mathbf{w}}^{(i)}(t)$ for all $k \in \mathcal{N}_i$ and $i \in \mathcal{M}$. On the other hand, client k in the overlapped area can receive multiple models from all neighboring ESs $i \in \mathcal{S}_k$. Clients in the overlapped area can be the bridge for sharing information between adjacent ESs, but each ES still wants to personalize its own model to the main tasks rather than OOD tasks; therefore, each client k in the overlapped area trains multiple models, each tailored to the main tasks of one of the neighboring ES. In other words, client k first receives the aggregated models $\bar{\mathbf{w}}^{(i)}(t)$ from ESs $i \in \mathcal{S}_k$, generates multiple initial models $\mathbf{w}_k^{(i)}(t)$ for different ESs $i \in \mathcal{S}_k$, and trains multiple local models independently. Here, the initial model of client k for ES i is obtained by

$$\mathbf{w}_k^{(i)}(t) = \begin{cases} \bar{\mathbf{w}}^{(i)}(t) & \text{for } k \in \mathcal{N}_i \\ \frac{1}{(1+\beta)}\bar{\mathbf{w}}^{(i)}(t) + \frac{\beta}{(1+\beta)}\sum_{j\in\mathcal{S}_k\setminus\{i\}} \frac{\bar{\mathbf{w}}^{(j)}(t)}{|\mathcal{S}_k|-1} & \text{for } k \in \mathcal{O}_i \end{cases}. \tag{4}$$

This mechanism enables each client to integrate information from the adjacent ES, thereby enhancing the resilience of the edge model to OOD tasks. Afterwards, client k performs the local training process by (2). In particular, client k in the overlapped area separately trains local models for all neighboring ESs $i \in \mathcal{S}_k$. After T local epochs, each ES i has the finally aggregated model $\bar{\mathbf{w}}_f^{(i)} = \bar{\mathbf{w}}^{(i)}(T)$. The details of the proposed algorithm is described in Algorithm 1.

IV. Experimental Results

This section presents the experimental results of our model evaluated on the CIFAR-10 dataset [12], which consists of ten classes but we only utilized nine classes for the experiments. The dataset was partitioned into 50,000 training samples and 10,000 test samples. We employed a convolutional neural network (CNN) architecture consisting of 2 convolutional layers followed by 2 fully connected layers, resulting in a total of 2,155,977 trainable parameters. All client models were initialized with the same random seed to ensure consistency across experiments. For local model updates, we utilized mini-batch Stochastic Gradient Descent (SGD) with a batch size of 10, applying a weight decay of 10^{-4} and a momentum of 0.9. The learning rate was initialized at 0.001 and decayed by a factor of 0.995 after each communication round. Each communication round comprised $E = 5$ local epochs, with the entire set of experiments conducted over 600 communication rounds.

Algorithm 1 Multi-cell FL

Input: Initial model $\mathbf{w}^0$
Output: Final global model $\mathbf{w}_f(T)$ and edge model $\mathbf{w}_f^{(i)}(T)$
Set $\mathbf{w}_k^{(i)}(0) = \mathbf{w}^0$ for all ESs $i \in \mathcal{M}$ and $k \in \mathcal{K}_i$
for each local epoch $t \in \{0, 1, \ldots, T-1\}$ **do**
  for each client $k \in \{1, 2, \ldots, K\}$ **in parallel do**
    for $i \in \mathcal{S}_k$ **do**
      $\mathbf{w}_k^{(i)}(t+1) \leftarrow \mathbf{w}_k^{(i)}(t) - \eta_t \nabla F_k(\mathbf{w}_k^{(i)}(t))$
    end for
  end for
  if $t \mid E = 0$ **then**
    for each ES $i \in \mathcal{M}$ **in parallel do**
      $\bar{\mathbf{w}}^{(i)}(t) \leftarrow \frac{1}{(1+\alpha)}\sum_{k\in\mathcal{N}_i} \frac{|\mathcal{N}_i|}{|\mathcal{N}_i|+|\mathcal{O}_i|}\mathbf{w}_k^{(i)}(t)$
        $+\frac{\alpha}{(1+\alpha)}\sum_{k\in\mathcal{O}_i} \frac{|\mathcal{O}_i|}{|\mathcal{N}_i|+|\mathcal{O}_i|}\mathbf{w}_k^{(i)}(t)$
      ES i broadcasts $\bar{\mathbf{w}}^{(i)}(t)$ to clients $k \in \mathcal{K}_i$
    end for
    for each client $k \in \{1, 2, \ldots, K\}$ **in parallel do**
      if $k \in \cup_{i=1}^{M}\mathcal{N}_i$ **then**
        $\mathbf{w}_k^{(i)}(t) \leftarrow \bar{\mathbf{w}}^{(i)}(t)$
      else
        $\mathbf{w}_k^{(i)}(t) \leftarrow \frac{1}{(1+\beta)}\bar{\mathbf{w}}^{(i)}(t)$
          $+\frac{\beta}{(1+\beta)}\sum_{j\in\mathcal{S}_k\setminus\{i\}} \frac{\bar{\mathbf{w}}^{(j)}(t)}{|\mathcal{S}_k|-1}$
      end if
    end for
  end if
end for

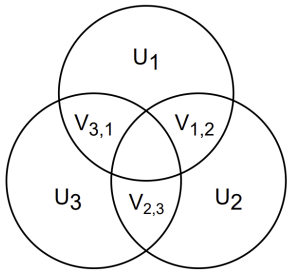


Fig. 1: The cell topology configured for the experiments.

A. Experimental Environment

We consider a cellular architecture consisting of $M = 3$ ESs and a total of $K = 144$ clients. For simplicity, we assume that the overlapped areas of only two adjacent ESs are considered, as shown in Fig. 1, $|\mathcal{N}_1| = |\mathcal{N}_2| = |\mathcal{N}_3| = u$ clients in non-overlapping regions, and $|\mathcal{O}_{1,2}| = |\mathcal{O}_{2,3}| = |\mathcal{O}_{3,1}| = v$ clients in overlapping regions, where $\mathcal{O}_{i,j}$ is the overlapping area of cells i and j. To differentiate the main and OOD tasks for each cell, classes $\{0, 1, 2\}$ are assigned to cell 1 as main tasks, and cell 2 and cell 3 have the main classes of $\{3, 4, 5\}$ and $\{6, 7, 8\}$, respectively. Each client randomly chooses two classes from their respective cell's main tasks for its training

dataset. For instance, clients in cell 1 can have data distributed across three combinations: $\{0,1\}$, $\{1,2\}$, and $\{2,0\}$. On the other hand, half of the clients within the overlapping area of two ESs choose their main classes from the main classes of one ES, and the other half choose their main classes from the main classes of the other ES. For example, in the overlapping region $\mathcal{O}_{1,2}$, a half of clients have the main tasks among $\{0,1\}$, $\{1,2\}$ and $\{2,0\}$ from the main classes of cell 1, and the remaining ones have the main classes among $\{3,4\}$, $\{4,5\}$, $\{5,3\}$ from the main classes of cell 2. Here, $u = 36$ and $v = 12$ are used unless otherwise noted.

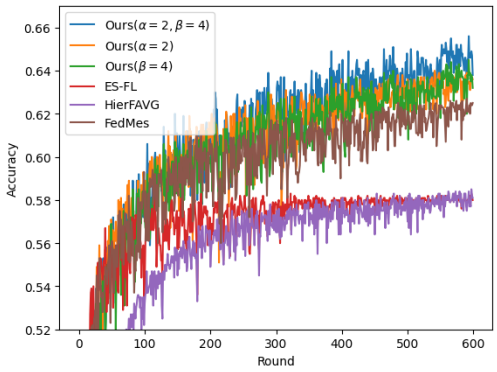
To evaluate the performance of the proposed method, we compared it with several baselines: HierFAVG [7], ES-FL (ES-based FL), and FedMes [9]. For the HierFAVG, edge aggregation was performed every 5 local epochs, and cloud aggregation was carried out after every 5 edge aggregations. ES-FL refers to a scenario where each ES independently conducts FedAvg [1] without intercommunication between ESs. FedMes, originally designed to mitigate the propagation delays caused by communications with a central cloud server, has primarily focused on training a global model to pursue the generalization capability. Since HierFAVG and ES-FL do not utilize overlapping regions, experiments for these methods were conducted with $u = 42$ and $v = 0$, ensuring that the main tasks across cells did not overlap. The proposed algorithm, FedMes, ES-FL do not generate a global model; therefore, we evaluate the final models of ESs individually and calculate the average performance. Although FedMes does not utilize the central server during the training phase, the authors of [9] allow the aggregation of the final models of all ESs at the end of the training. Accordingly, we also compare our method with the global models of FedMes as well as HierFAVG.

B. Test Accuracy Versus Global Round

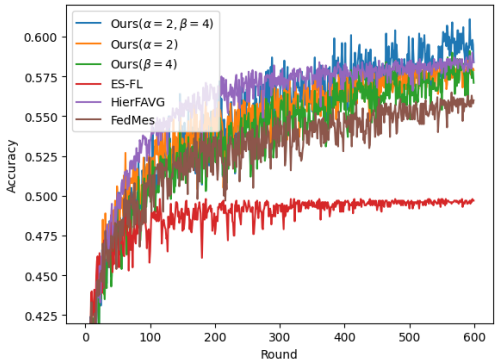
Fig. 2 presents the test accuracy performances over communication rounds with different ρ values. Recall that ρ is the ratio of the number of test samples with main classes to the total size of the test dataset. We observe that the proposed scheme with appropriate values of α and β outperforms all the baselines. Although the proposed scheme with α or β only also shows better performances than FedMes and ES-FL, we note that utilization of both α and β improves the accuracy more. Here, the performance of FedMes is obtained by taking averages of the accuracy of every ES; therefore, the proposed scheme, FedMes, and ES-FL show the improved accuracy as ρ grows. On the other hand, since HierFAVG trains a global model, its accuracy does not change with ρ. Accordingly, ↗he performance improvement of the proposed technique compared to HierFAVG increases as the ρ value grows.

C. Test Accuracy Versus Test-set Distribution

In Fig. 3, the test accuracy of the baselines for different ρ values is shown. Here, we note that all the schemes that train a single global model for the generalization capability, i.e., the global model of the proposed scheme, HierFAVG, and the



(a) Main/Total (ρ) = 0.7



(b) Main/Total (ρ) = 0.6

Fig. 2: The accuracy of the model per round when the proportion of Main tasks in the overall test set is (a) 60% and (b) 70%.

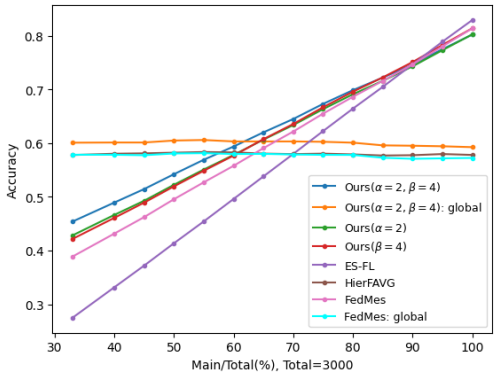
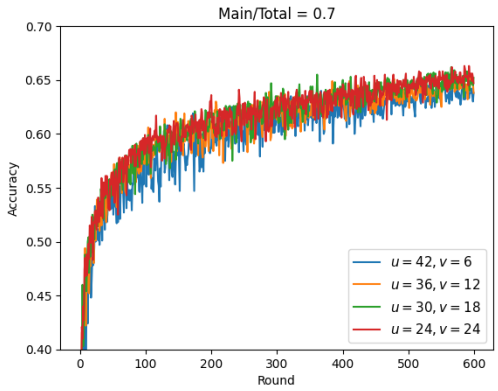
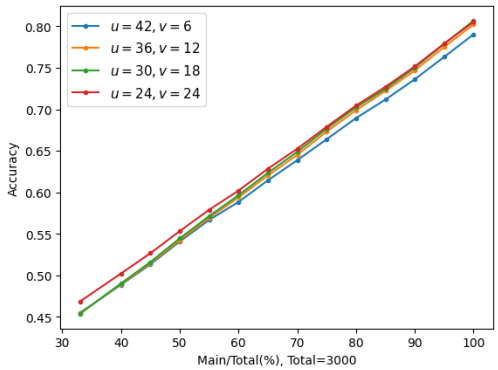


Fig. 3: Accuracy of the models with respect to changes in the distribution of the test set.

global model of FedMes, maintain the accuracy at a similar level. On the other hand, the baselines that allow each ES to train its personalized model, i.e., the proposed scheme, ES-FL, and FedMes, show an increasing trend of the test accuracy with ρ values. In any cases, we can realize that the proposed scheme using both α and β outperform the baselines for most of ρ values, except for the case of $\rho \approx 1.0$, because the proposed scheme is designed for handling the OOD tasks. However, we



(a) Round vs. Accuracy



(b) Main/Total vs. Accuracy

Fig. 4: Accuracy results of the edge model with respect to changes in the number of clients in the overlapped region.

can argue that in the potential test scenario where the majority of the test samples are from the main classes, the proposed scheme always shows the best performance and the decrease of the accuracy compared to ES-FL is not significant even when $\rho = 1.0$ indicating no OOD tasks in the test time.

D. Impact of the number of clients in overlapped regions

We then investigated the impact of the number of clients in overlapped areas on the test accuracy of the proposed scheme in Fig. 4. In particular, $\rho = 0.7$ is assumed in Fig. 4a. As described in Fig. 1, the numbers of clients in the nonoverlapped and overlapped areas of a single cell are denoted by u and v, respectively. In both Figs. 4a and 4b, we observe that the test accuracy is not significantly degraded as v decreases, achieving the applicability of the proposed scheme to the real-world scenario. In Fig. 4b, a slight decrease of the accuracy with v is observed when ρ is high; however, this is not significant, and the test accuracy is not degraded at all with 25% of clients in the overlapped area only.

V. Conclusion

This paper proposes a serverless FL algorithm with hyperparameters that can control the tradeoff between personalization and generalization, specifically designed for multi-cell environments. In addition, we design the model aggregation process for achieving the robustness to the test-time distribution shifts. Experimental results on the CIFAR-10 dataset demonstrate that our proposed serverless FL framework outperforms the existing centralized and distributed FL approaches. Furthermore, the proposed scheme personalizes the model of the ES to its main classes while providing the generalization capability at a certain level to handle the OOD tasks. Although the proposed scheme leverages the handover region of the wireless cellular network to allow cooperations among neighboring ESs even without a central server, our approach is not sensitive to the number of the overlapped region.

Acknowledgment

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02201, Federated Learning for Privacy-Preserving Video Caching Networks), in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2022R1C1C1010766), and in part by NRF grant funded by MSIT (No. 2022R1A4A3033401).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, et al., "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1-19, Jan. 2019.

[4] K. Bonawitz, V. Ivanov, B. Kreuter, et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175-1191.

[5] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[6] H. Wang, Z. Kaplan, D. Niu, et al., "Optimizing federated learning on non-IID data with reinforcement learning," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 987-997.

[7] L. Liu, J. Zhang, S. H. Song and K. B. Letaief, "Client-Edge-Cloud Hierarchical Federated Learning," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-6.

[8] B. Ganguly et al., "Multi-Edge Server-Assisted Dynamic Federated Learning With an Optimized Floating Aggregation Point," in IEEE/ACM Transactions on Networking, vol. 31, no. 6, pp. 2682-2697, Dec. 2023.

[9] D.-J. Han, M. Choi, J. Park, and J. Moon, "FedMes: Speeding up federated learning with multiple edge servers," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3870-3883, Dec. 2021.

[10] D. -J. Han, D. -Y. Kim, M. Choi, C. G. Brinton and J. Moon, "SplitGP: Achieving Both Generalization and Personalization in Federated Learning," IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, New York City, NY, USA, 2023, pp. 1-10.

[11] Y. Zhao, M. Li, J. Liang, et al., "Federated learning with non-IID data," *arXiv preprint arXiv:1806.00582*, 2018.

[12] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," 2009.