Multi-center federated learning: clients clustering for better personalization

Guodong Long[1] · Ming Xie[1] · Tao Shen[1] · Tianyi Zhou[2,3] · Xianzhi Wang[1] · Jing Jiang[1]

Abstract

Personalized decision-making can be implemented in a Federated learning (FL) framework that can collaboratively train a decision model by extracting knowledge across intelligent clients, e.g. smartphones or enterprises. FL can mitigate the data privacy risk of collaborative training since it merely collects local gradients from users without access to their data. However, FL is fragile in the presence of statistical heterogeneity that is commonly encountered in personalized decision making, e.g., non-IID data over different clients. Existing FL approaches usually update a single global model to capture the shared knowledge of all users by aggregating their gradients, regardless of the discrepancy between their data distributions. By comparison, a mixture of multiple global models could capture the heterogeneity across various clients if assigning the client to different global models (i.e., centers) in FL. To this end, we propose a novel multi-center aggregation mechanism to cluster clients using their models' parameters. It learns multiple global models from data as the cluster centers, and simultaneously derives the optimal matching between users and centers. We then formulate it as an optimization problem that can be efficiently solved by a stochastic expectation maximization (EM) algorithm. Experiments on multiple benchmark datasets of FL show that our method outperforms several popular baseline methods. The experimental source codes are publicly available on the Github repository (GitHub repository: https://github.com/mingxuts/multi-center-fed-learning).

Keywords Federated learning · Personalized modeling · Clustering

---

Guodong Long and Ming Xie contributed equally to this work.

✉ Jing Jiang
  jing.jiang@uts.edu.au

Extended author information available on the last page of the article

1 Introduction

The widespread of social networks and mobile APPs has witnessed a huge volume of data generated by end-users on mobile devices [4]. Generally, a service provider on the server side collect users' data and train a global machine learning model such as deep neural networks. Such a centralized machine learning approach causes severe practical issues, e.g., communication costs, consumption of device batteries, and the risk of violating the privacy and of user data. Moreover, each user or client may have different decision making preferences that may caused by the changes of their life events and dynamic networks.

The modern decision-making system is generally empowered by various deep neural network-based intelligent models that include two components: embedding and decision. The embedding component could be a backbone is composed of CNN [16, 53], GCN [30, 58], RNN/LSTM [12, 64], or Self-attention/Transformer modules [46, 55] to transform the multi-modal data, such as images, graphs, time series, sequential behavior and texts, into numeric vectors for further processing. The decision component usually takes a multi-layer fully connected neural network to model the complex relationship between inputs and decisions (outputs). Then, the embedding and decision components will be connected to form an entire deep learning model to be trained in an end-to-end manner.

Federated learning (FL) [43] is a new machine learning paradigm that learns models collaboratively using the training data distributed on remote devices to boost communication efficiency. There are three advantages that can make FL be the best option to implement a personalized decision-making system. First, the deep learning model requires a huge volume of training data that is usually impractical for a single client, thus, the FL-based framework enables the clients to collaboratively train a model with abundant training data. Second, the FL separates the model training and data storage, which can greatly reduce the privacy risk for participating clients. Third, each client can take part in the model training process of the FL system, and then it can take the opportunity to train a personalized decision-making model by leveraging both the globally shared knowledge and locally stored data.

The vanilla FL addresses a practical setting of distributed learning, where 1) the central server is not allowed to access any user data which protects users' privacy, and 2) the data distribution over different users is non-IID, which is a natural assumption of real-world applications. However, early FL approaches [43, 59] use only one global model as a single-center to aggregate the information of all users. The stochastic gradient descent (SGD) for single-center aggregation is designed for IID data, and therefore, conflicts with the non-IID setting in FL. The observed data in each device are generated or produced by the user. The dataset across devices are usually non-IID that reflects different preferences or decision logic of users. We assume the whole population could be partitioned into different clusters or groups in which the users have similar preferences. Therefore, learning an intelligent model customised for each group with non-IID data can assist users to make decision-making by considering personal preferences.

Recently, the non-IID or heterogeneity challenge of FL has been studied to improve the robustness of global models against outlier/adversarial users and devices [15, 25, 26]. Moreover, [50] proposed an idea of clustered FL (FedCluster) that addresses the non-IID issue by dividing the users into multiple clusters. However, the hierarchical clustering in FedCluster is achieved by multiple rounds of bipartite separation, each requiring the federated SGD algorithm to run until convergence. Hence, its computational and communication efficiency will become bottlenecks when applied to a

large-scale FL system. More recently, [42] and [14] proposed to cluster the local models according to the loss of hypothesis. In particular, each user will try all K global models representing K clusters, and then select the best global model as the cluster ID by considering the lowest loss of running the global model on local data. However, this posts high communication and computation overheads because the selected nodes will spend more resources for receiving and running multiple global models.

In this paper, we propose a novel multi-center FL framework that updates multiple global models by aggregating information from multiple user groups. In particular, the datasets of the users in the same group are likely to be generated or derived from the same or similar distribution. We formulate the problem of the multi-center FL as the joint clustering of users, and then optimizing of the global model for users in each cluster. In particular, (1) each user's local model is assigned to its closest global model, and (2) the global model in each cluster leads to the smallest loss over all the associated users. The proposed multi-center FL not only inherits the communication efficiency of the federated SGD but also retains the capability of handling non-IID data on heterogeneous datasets. Lastly, we propose a new optimization method in line with EM algorithm to train our model.

We summarize our main contributions as:

- We propose a novel multi-center aggregation approach (Section 4.1) to address the non-IID challenge of personalized decision-making system.
- We design an objective function, namely multi-center federated loss (Section 4.2), for user clustering in FL.
- We propose Federated Stochastic Expectation Maximization (FeSEM) (Section 4.3) to solve the optimization of the proposed objective function.
- We present the algorithm as an easy-to-implement and strong baseline for FL. Its effectiveness is evaluated on benchmark datasets. (Section 6)

2 Related work

2.1 Personalized decision-making

Decision making is the process of making choices by identifying a decision, gathering information, and assessing alternative resolutions. In most of the scenarios, each individual person usually makes a personal choice given the collected information. To model the personalized decision-making process [41, 47], a general solution is to collect the user's personal characteristics, e.g. demographics [45], behavior history [51], and social networks [63], as part of the input to be considered by a centralized intelligent model. This solution usually train a large-scale machine learning or recommendation models at cloud server using the collected personal data from users, thus it will cause privacy concerns. In recent, a new service architecture has been proposed to provide service based on a standalone on-device intelligent [23] in each smart device. In particular, a unique intelligent model customized for each user will be deployed to the user' smart device, so as to provide service independently while not relying on the decision from the cloud server. The user's personal data will be stored locally to train the intelligent model, thus no personal data will be uploaded to the server.

2.2 Federated learning

Federated learning (FL) enables users to leverage rich data for machine learning models without compromising their data. It has attracted a significant amount of research interest since 2017, with many studies investigating FL from several aspects, e.g., system perspective, personalized models, scalability [3], communication efficiency [24], and privacy [13]. Most of the related work addresses a particular concern such as security or privacy [6, 35, 49]. It has been applied to various industry applications, such as banking [37], healthcare [36, 48, 57], and mobile internet applications [21].

FL is designed for specific scenarios that can be further expanded to a standard framework to preserve data privacy in large-scale machine learning systems or mobile edge networks [32]. For example, [59] expanded FL by introducing a comprehensive, secure FL framework that includes horizontal FL, vertical FL, and federated transfer learning. The work in [28, 39] surveyed the FL systems in relation to their functions on privacy protection and security threats. [22] discussed the advances and open problems in FL. [5] proposed LEAF – a benchmark for federated settings with multiple datasets. [38] proposed an object detection-based dataset for FL.

Heterogeneity is a core challenge in the federated setting and has been widely studied from various perspectives. [17] conducted theoretical convergence analysis for FL with heterogeneous data. [19] measured the effects of non-IID data for federated visual classification. [60] proposed a heterogeneity-aware platform design for FL. [31] discussed the local representations that enable data to be processed on new devices in different ways according to their source modalities instead of using a single global model. The single global model might not generalize to unseen modalities and distributions of data. [27] proposed a new federated setting composed of a shared global dataset and many heterogeneous datasets from devices. [20] and [33] proposed to integrate knowledge distillation with FL to tackle the model heterogeneity. [61] proposed a general FL framework to align heterogeneous model architectures and functional neurons. [54] proposed to a prototype learning-based FL framework to tackle the challenges of task heterogeneity across devices. Particularly, it aggregates the prototypes rather than model parameters, and then the communication efficiency is much higher than gradient-based algorithm, such as FedAvg.

To solve the problem caused by non-IID data in a federated setting [5, 50] proposed clustered FL (FedCluster) by integrating FL and bi-partitioning-based clustering into an overall framework, and [14, 42] proposed a hypothesis-based federated clustering that assigns the cluster by considering the loss of running the global model on local data. [15] proposed a robust FL comprising three steps: 1) learning a local model on each device, 2) clustering model parameters to multiple groups, each being a homogeneous dataset, and 3) running a robust distributed optimization [29] in each cluster. [40] propose a general form to model the clustered FL problem into a bi-level optimization framework, and then conduct theoretical analysis on the convergence.

[26] proposed FedDANE by adapting the DANE [52] to a federated setting. In particular, FedDANE is a federated Newton-type optimization method. [25] proposed FedProx for the generalization and re-parameterization of FedAvg [43]. It adds a proximal term to the objective function of each device's supervised learning task, and the proximal term is to measure the parameter-based distance between the server and the local model. [1] added a personalized layer for each local model, i.e., FedPer, to tackle heterogeneous data. [8] propose to a structure-based model aggregation mechanism to enhance the personalized federated learning.
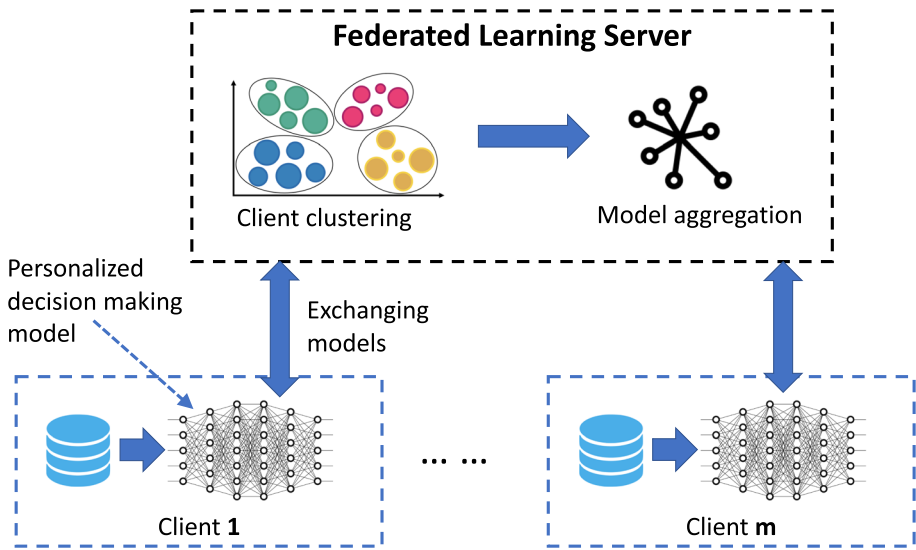
Fig. 1 Overall framework of multi-center Federated Learning

3 Background

3.1 Overall framework

This section will introduce the overall framework of the proposed multi-center Federated Learning. As shown in Figure 1, each client could be an intelligent device or computer in an enterprise, and they will collaboratively train an intelligent model via a coordinating server. First, the client initializes its local model using the global model from the federated learning server. Second, the client trains the local model using its own data. Third, the trained model will be uploaded into the server that will conduct client clustering across clients' model parameters, and then conduct model aggregation in each cluster. Fourth, the model in each cluster will be dispatched to the corresponding clients. The multi-center FL framework will repeat these four steps till convergence or the stop condition is satisfied.

In the following subsections, we will give a formal formulation of the proposed problem, and then explain the motivation of the design. The detailed method and optimization procedure is provided in Section 4.

3.2 Problem setting

In FL, each device-i has a private dataset $\mathcal{D}_i = \{\mathcal{X}_i, \mathcal{Y}_i\}$, where $\mathcal{X}_i$ and $\mathcal{Y}_i$ denote the input features and corresponding gold labels respectively. Each dataset $\mathcal{D}_i$ will be used to train a local supervised learning model $\mathcal{M}_i : \mathcal{X}_i \to \mathcal{Y}_i$. $\mathcal{M}$ denotes a deep neural model parameterized by weights W. It is built to solve a specific task, and all devices share the same model architecture.

For the i-th device, given a private training set $\mathcal{D}_i$, the training procedure of $\mathcal{M}_i$ is represented in brief as

$$\min_{W_i} L_s(\mathcal{M}_i, \mathcal{D}_i, W_i), \tag{1}$$

where $L_s(\cdot)$ is a general definition of the loss function for any supervised learning task, and its arguments are model structure, training data and learnable parameters respectively, and W' denotes the parameters after training. In general, the data from one device is insufficient to train a data-driven neural network with satisfactory performance. An FL framework optimizes the local models in a distributed manner and minimizes the loss of the local data on each device.

Hence, the optimization in vanilla FL over all the local models can be written as

$$\min_{\{W_i\}_{i=1}^m} \alpha_i L_s(\mathcal{M}_i, \mathcal{D}_i, W_i)$$

where m denotes the number of devices, and $\alpha_i = \frac{|\mathcal{D}_i|}{\sum_j |\mathcal{D}_j|}$ is an importance weight that is measured by the number of samples on each clients.

On the server side, the vanilla FL aggregates all local models into a global one $\mathcal{M}_{global}$ which is parameterized by $\tilde{W}^g$. In particular, it adopts a weighted average of the local model parameters $[W_i]_{i=1}^m$, i.e.,

$$\tilde{W}^g = \sum_{i=1}^m \alpha_i W_i, \tag{2}$$

which is the nearest center for all $\{W_i\}_{i=1}^m$ in terms of a weighted L2 distance:

$$\tilde{W}^g \in \arg\min_{\tilde{W}} \sum_{i=1}^m \alpha_i \|\tilde{W} - W_i\|_2^2. \tag{3}$$

More generally, we can replace the L2 distance in (3) by other distance metric $\mathrm{Dist}(\cdot, \cdot)$ and minimize the difference between the global model and all the local models, i.e.,

$$\min_{\tilde{W}} \frac{1}{m} \sum_{i=1}^m \mathrm{Dist}(W_i, \tilde{W}). \tag{4}$$

The above aims to find a consistent solution across global model and local models. Note that a direct macro average is used here regardless of the weight of each device, which treats every device equally. The weights used in () can easily be incorporated for a micro average.

The divergence $\mathrm{Dist}(\cdot, \cdot)$ between the global model and local models plays an essential role in the FL objective. The simple L2 distance for $\mathrm{Dist}(\cdot, \cdot)$ does not take into account the fact that two models can be identical under the arbitrary permutation of neurons in each layer.

Hence, the lack of neuron matching may cause misalignment in that two neurons with similar functions and different indexes cannot be aligned across models [62]. However, the index-based neuron matching in FL [52] is the most widely used method and works well in various real applications. One potential reason for this is that the index-based neuron matching can also slowly align the function of neurons by repeatedly initializing all local models with the same global model. To simplify the description, we will discuss our method for index-based neuron matching.

3.3 Motivation

Federated learning (FL) usually aggregates all local models to a single global model. However, this single-center aggregation is fragile under heterogeneity. In contrast, we consider FL with multiple centers to better capture the heterogeneity by assigning nodes to different centers so only similar local models are aggregated. Consider two extreme cases for the number of centers, K: (1) when $K = 1$, it reduces to the FedAvg with a single global model, which cannot capture the heterogeneity and the global model might perform poorly on specific nodes; (2) When $K = m$, the heterogeneity problem can be avoided by assigning each node to one global model. But the data on each device used to update each global model can be insufficient and thus we lose the main advantage of FL. Our goal is to find a sweet point between these two cases to balance the advantages of federated averaging and the degradation caused by underlying heterogeneity.

Learning one unique model for each node has been discussed in some recent FL studies for better personalized models. They focus on making a trade-off between shared knowledge and personalizing. The personalizing strategy either applies fine-tuning of the global model [65] for each node, or only updates a subset of personalized layers for each node [1, 31], or deploys a regularization term in the objective [10, 11, 18]. In contrast, Multi-center FL in this paper mainly focuses to address the heterogeneity challenge by assigning nodes to different global models during aggregation. But it can be easily incorporated in these personalization strategies. In the following, we will start from the problem setting for the the vanilla FL, and then elucidate our motivation of improving FL's tolerance to heterogeneity by multi-center design.

4 Methodology

4.1 Multi-center model aggregation

To overcome the challenges arising from the heterogeneity in FL, we propose a novel model aggregation method with multiple centers, each associating with a global model $\tilde{W}^{(k)}$ updated by aggregating a cluster of user's models with nearly IID data. In particular, all the local models will be grouped to K clusters, denoted as $C_1, \cdots, C_K$, each covering a subset of local models with parameters $\{W_j\}_{j=1}^{m_k}$.

An intuitive comparison between the vanilla FL and our multi-center FL is illustrated in Figure 2. As shown in the left figure, there is only one center model in vanilla FL. In contrast, the multi-center FL shown in the right has two centers, $W^{(1)}$ and $W^{(2)}$, and each center represents a cluster of devices with similar data distributions and models. Obviously, the right one has a smaller intra-cluster distance than the left one. As discussed in the following Section 4, intra-cluster distance directly reflects the possible loss of the FL. Hence, a much smaller intra-cluster distance indicates our proposed approach potentially reduces the loss of FL.
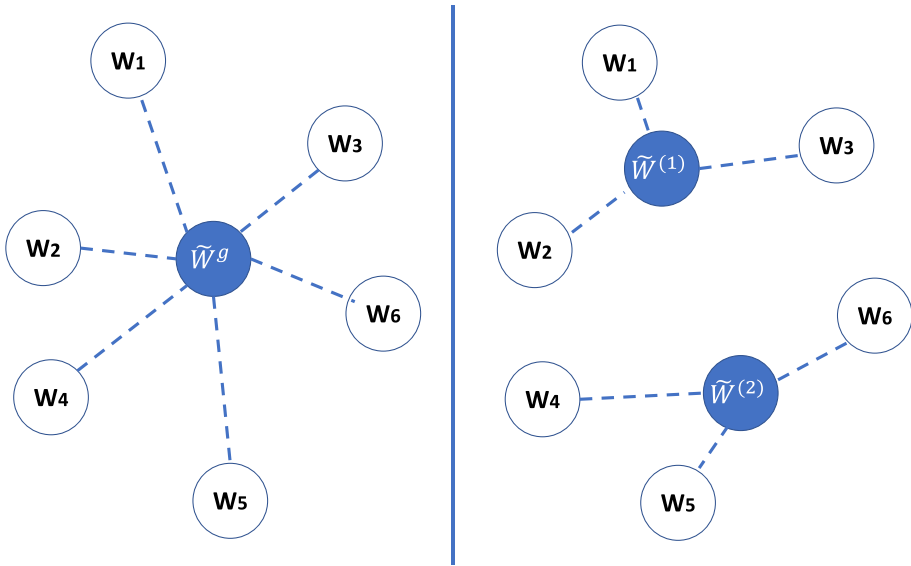
Fig. 2 Comparison between single-center aggregation in vanilla FL (left) and multi-center aggregation in the proposed one (right). Each W_i represents the local model's parameters collected from the i-th device, which is denoted as a node in the space. $\tilde{W}$ represents the aggregation result of multiple local models

4.2 Problem formulation

4.2.1 Solving a joint optimization on a distributed network.

The multi-center FL problem can be formulated as an optimisation framework as below.

$$
\min_{\{W_i\},\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \sum_{i=1}^{m} \alpha_i L_s(\mathcal{M}_i, \mathcal{D}_i, W_i) \\
+ \frac{\lambda}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \mathrm{Dist}(W_i, \tilde{W}^{(k)}), \tag{5}
$$

where λ controls the trade-off between supervised loss and distance. We solve it by applying an alternative optimization between server and user: (1) on each node-i, we optimize the above objective w.r.t. W_i while fixing all the other variables; and (2) on the server, we optimize $\{r_i^{(k)}\}$, $\{\tilde{W}^{(k)}\}$ for $i \in [m]$ and $k \in [K]$ while fixing all local models $\{W_i\}$.

4.2.2 Multi-center assignment at the server end.

The second term in (5) aims to minimize the distance between each local model and its nearest global model. Under the non-IID assumption, the data located at different devices can be grouped into multiple clusters where the on-device data in the same cluster are likely to be generated from one distribution. As illustrated on the right of Figure 2, we optimizes the assignments and global models by minimizing the intra-cluster distance, i.e,

$$\min_{\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \mathrm{Dist}(W_i, \tilde{W}^{(k)}), \tag{6}$$

where cluster assignment $r_i^{(k)}$, as defined in (8), indicates whether device-i belongs to cluster-k, and $\tilde{W}^{(k)}$ is the parameters of the aggregated model for cluster-k.

4.2.3 Distance-constrained loss for local model optimization.

Because the distance between the local model and the global model are essential to our new loss, we don't expect the local model will be changed too much during the local updating stage. The new loss consists of a supervised learning loss and a regularization term to constrain the local model to ensure it is not too far from the global model. This kind of regularization term is also known as the proximal term in [25] that can effectively limit the impact of the variable local updates in FL. We minimize the loss below for each local model W_i as follows:

$$\min_{W_i} \alpha_i L_s(\mathcal{M}_i, \mathcal{D}_i, W_i) + \frac{\lambda}{m} \sum_{k=1}^{K} r_i^{(k)} \mathrm{Dist}(W_i, \tilde{W}^{(k)}) \tag{7}$$

4.3 Optimization algorithm

In general, Expectation-Maximization (EM) [2] is an optimisation framework to solve alternative updating of multiple parameters, for example, K-Means clustering with updates on cluster centers and cluster assignments. However, in contrast to the general objective of clustering, our proposed objective, as described in (6), has a dynamically changing W_i during optimization. Therefore, we adapt the Stochastic Expectation Maximization (SEM) [7] optimization framework by adding one step, i.e., updating W_i. In the modified SEM optimization framework, named federated SEM (FeSEM), we will iterative conduct a sequence of actions as below.

Firstly, for the **E-Step**, we update cluster assignment $r_i^{(k)}$ with fixed W_i. We calculate the distance between the cluster center and nodes – each node is the model's parameters W_i, then update the cluster assignment $r_i^{(k)}$ by

$$r_i^{(k)} = \begin{cases} 1, & if\ k = \arg\min_j \mathrm{Dist}(W_i, \tilde{W}^{(j)}) \\ 0, & otherwise. \end{cases} \tag{8}$$

Secondly, for the **M-Step**, we update the cluster center $\tilde{W}^{(k)}$ according to the W_i and $r_i^{(k)}$, i.e.,

$$\tilde{W}^{(k)} = \frac{1}{\sum_{i=1}^{m} r_i^{(k)}} \sum_{i=1}^{m} r_i^{(k)} W_i. \tag{9}$$

Thirdly, to **update the local models**, the global model's parameters $\tilde{W}^{(k)}$ are sent to each device in cluster k to update its local model, and then we can fine-tune the local model's parameters W_i using a supervised learning algorithm on its own private training data while considering the new loss as described in (7). The local training procedure is a supervised

learning task by adding a distance-based regularization term. The local model is initialized by the global model $\tilde{W}^{(k)}$ which belong to the cluster associated with the node.

Lastly, we repeat the three stochastic updating steps above until convergence. The sequential executions of the three updates comprise the iterations in FeSEM's optimization procedure. In particular, we sequentially update three variables $r_i^{(k)}$, $\tilde{W}^{(k)}$, and W_i while fixing the other factors. These three variables are jointly used to calculate the objective of our proposed multi-center FL in (6).

Algorithm 1 FeSEM – Federated Stochastic EM

1: Initialize $K, \{W_i\}_{i=1}^m, \{\tilde{W}^{(k)}\}_{k=1}^K$
2: **while** stop condition is not satisfied **do**
3:     **E-Step**:
4:     Calculate distance $d_{ik} \leftarrow \text{Dist}(W_i, \tilde{W}^{(k)}) \ \forall i, k$
5:     Update cluster assignment $r_i^{(k)}$ using d_{ik} (Eq. 8)
6:     **M-Step**:
7:     Update $\tilde{W}^{(k)}$ using $r_i^{(k)}$ and W_i (Eq. 9)
8:     **for** each cluster $k = 1, \ldots K$ **do**
9:         **for** $i \in C_k$ **do**
10:            Send $\tilde{W}^{(k)}$ to device i
11:            $W_i \leftarrow$ **Local_update**$(i, \tilde{W}^{(k)})$
12:        **end for**
13:    **end for**
14: **end while**

Algorithm 2 Local_update

---

i – device index
$\tilde{W}^{(k)}$ – the model parameters from server
W_i – updated local model
Initialization: $W_i \leftarrow \tilde{W}^{(k)}$
for N local training steps **do**
    Update W_i with training data $\mathcal{D}_i$ (Eq. 7)
end for
Return W_i to server

We implement FeSEM in Algorithm 1 which is an iterative procedure. As elaborated in Section 4.2, each iteration comprises of three steps to update three sets of variables that are cluster assignment r for each client, the representation of cluster centre $\tilde{W}$, and the local models' parameters W, respectively. In particular, the update of the local model is executed on the client by fine-tuning the model as in Algorithm 2.

5 Some possible extensions

To further handle heterogeneous data in FL scenario, our multi-center FL approach can be easily extended with other packages. We discuss two beneficial techniques here.

5.1 Model aggregation with neuron matching

The vanilla FL algorithm, FedAvg [43], uses model aggregation with index-based neuron matching which may cause the incorrect alignment. Neurons with similar functions usually take different indexes in two models. Recently, a function-based neuron matching [56] in FL is proposed to align two models by matching the neurons with similar functions. In general, the index-based neuron matching can gradually align the neuron's functionality across nodes by repeatedly forcing each local model to be initialized using the same global model. However, the function-based neuron matching can speed up the convergence of neuron matching and preserve the unique functional neuron of the minority groups.

In this work, we integrate layer-wise matching and then averaging(MA) [56] into ours to increase the capacity to handle heterogeneous challenges. In the model aggregation step at the FL server, we need to align the layer-wise parameters among models from different clients. For example, given a layer with three neurons {A1, A2, A3} in model A, and a layer with neurons {B1, B2, B3} in model B, the neuron matching is to find an alignment between two sets that could be seen as a bi-party alignment problem. If we use index-based neuron matching, e.g. FedAvg [43], we just simply aggregate the parameters of A1 and B1 according to their index number in each model. If we choose function-based neuron matching, e.g. FedAvg [56], we probably will aggregate the neuron A1 to B3 that they maybe share similar functions with different indexes in each model. The distance between the local model and the global model is the neuron matching score that is calculated by estimating the maximal posterior probability of the j-th client neuron l generated from a Gaussian with mean W_i, and ϵ and $f(\cdot)$ are guided by the Indian Buffet Process prior [62].

5.2 Selection of K

The selection of K, the number of centers, is essential for a multi-center FL. In general, the K is defined based on the prior experience or knowledge of data. If there is no prior knowledge, the most straightforward solution is to run the algorithm using different K and then select the K with the best performance in terms of accuracy or intra-cluster distance. Selecting the best K in a large-scale FL system is time consuming, hence we simplify the process by running the algorithm on a small number of sampled nodes with several communication rounds. For example, we can randomly select 100 nodes and test K in FL with three communication rounds only, and then apply the K to the large-scale FL.

Table 1 Statistics of datasets

DATASET	FEMNIST	FedCelebA
# of data points	805,263	200,288
# of device	3,550	9,343
# of Classes	62	2
Model architecture	CNN	CNN

6 Experiments

As a proof-of-concept scenario to demonstrate the effectiveness of the proposed method, we experimentally evaluate and analyze FeSEM on two datasets.

6.1 Training setups

6.1.1 Datasets

We employed two publicly-available federated benchmarks datasets introduced in LEAF [5]. LEAF is a benchmarking framework for learning in federated settings. The datasets used are Federeated Extended MNIST (FEMNIST)[1] [9] and Federated CelebA (FedCelebA)[2] [34]. We follow the setting of the benchmark data in LEAF. In FEMNIST, images is split according to the writers. For FedCelebA, images are extracted for each person and developed an on-device classifier to recognize whether the person smiles or not. A statistical description of the datasets is described in Table 1.

6.1.2 Local model

We use a CNN with the same architecture from [34]. Two data partition strategies are used: (a) an ideal IID data distribution using randomly shuffled data, (b) a non-IID partition by use a $\mathbf{p}_k \sim Dir_J(0.5)$. Part of the code is adopted from [56]. For FEMINST data, the local learning rate is 0.003 and epoch is 5. and for FedCelebA, 0.03 and 10 respectively.

6.1.3 Baselines

In the scenario of solving statistical heterogeneity, we choose FL methods as follows:

1. **NonFed**: We will conduct the supervised learning task at each device without the FL framework.
2. **FedSGD**: uses SGD to optimize the global model.
3. **FedAvg**: is an SGD-based FL with weighted averaging. [43] .
4. **FedCluster**: is to enclose FedAvg into a hierarchical clustering framework [50].
5. **HypoCluster(K)**: is a hypothesis-based clustered-FL algorithm with different K [42].

---

[1] http://www.nist.gov/itl/products-and-services/emnist-dataset

[2] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Table 2 Comparison of our proposed FeSEM(K) algorithm with the baselines on FEMNIST. Note the number in parenthesis following "FeSEM" denotes the number of clusters, K

| Dataset | FEMNIST | | | |
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	79.0±2.0	67.6±0.6	81.3±1.9	51.0±1.2
FedSGD	70.1±2.2	61.2±3.4	71.5±1.8	46.7±1.2
FedAvg [43]	84.9±2.0	67.9±0.4	84.9±1.6	45.4±1.9
FedDist [44]	79.3±0.8	67.5±0.5	79.8±1.1	50.5±0.5
FedDist+WS	80.4±0.8	67.2±1.6	80.6±1.2	51.7±1.1
Robust(TKM) [15]	78.4±1.0	53.1±0.5	77.6±0.7	53.6±0.7
FedCluster [50]	84.1±1.1	64.3±1.3	84.2±1.0	**64.4**±1.6
HypoCluster(3) [42]	82.5±1.7	61.3±0.6	82.2±1.3	61.6±0.9
FedDane [26]	40.0±2.9	31.8±3.1	41.7±2.4	31.7±1.6
FedProx [25]	72.6±1.8	62.8±1.6	74.3±2.1	50.6±1.2
FeSEM(2)	84.8±1.1	65.5±0.4	84.8±1.6	52.0±0.5
FeSEM(3)	87.0±1.2	68.5±2.0	86.9±1.2	41.7±1.5
FeSEM(4)	**90.3**±1.5	70.6±0.9	**91.0**±1.8	53.4±0.6
FeSEM-MA(3)	**90.4**±1.5	**71.4**±0.5	87.0±2.0	64.3±0.5

6. **Robust** our implementations based on the proposed method in [15], see this baseline settings in Appendix.
7. **FedDANE**: this is an FL framework with a Newton-type optimization method. [26].
8. **FedProx**: this is our our own implementations following [25]. We set scaler of proximal term to 0.1.
9. **FedDist**: we adapt a distance based-objective function in Reptile meta-learning [44] to a federated setting.
10. **FedDWS**: a variation of FedDist by changing the aggregation to weighted averaging where the weight depends on the data size of each device.
11. **FeSEM(K)**: our multi-center FL implemented on federated SEM with K clusters.

6.1.4 Training settings

We used 80% of each device's data for training and 20% for testing. For the initialization of the cluster centers in FeSEM, we conducted pure clustering 20 times with randomized initialization, and then the "best" initialization, which has the minimal intra-cluster distance, was selected as the initial centers for FeSEM. For the local update procedure of FeSEM, we set N to 1, meaning we only updated W_i once in each local update.

6.1.5 Evaluation metrics

Given numerous devices, we evaluated the overall performance of the FL methods. We used classification accuracy and F1 score as the metrics for the two benchmarks. In addition, due to the multiple devices involved, we explored two ways to calculate the metrics, i.e., micro and macro. The only difference is that when computing an overall metric, "micro" calculates a weighted average of the metrics from devices where the weight is proportional to the data amount, while "macro" directly calculates an average over the metrics from devices.

Table 3 Comparison of our proposed FeSEM(K) algorithm with the baselines on FedCelebA. Note the number in parenthesis following "FeSEM" denotes the number of clusters, K

Dataset	FedCelebA			
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	83.8±1.4	66.0±0.4	83.9±1.6	67.2±0.6
FedSGD	75.7±2.3	60.7±2.4	75.6±2.0	55.6±2.6
FedAvg [43]	86.9±0.5	**78.0**±1.0	86.1±0.4	54.2±0.6
FedDist [44]	71.8±0.9	61.0±0.8	71.6±1.0	61.1±0.7
FedDist+WS	73.4±1.7	59.3±0.9	73.4±1.9	50.3±0.5
Robust(TKM) [15]	90.1±1.3	68.0±0.7	90.1±1.3	68.3±1.1
FedCluster [50]	86.7±0.7	67.8±0.9	87.0±0.9	67.8±1.3
HypoCluster(3) [42]	76.1±1.5	53.5±1.0	72.7±1.8	53.8±1.9
FedDane [26]	76.6±1.1	61.8±2.0	75.9±1.0	62.1±2.2
FedProx [25]	83.8±2.0	60.9±1.2	84.9±1.8	65.7±1.2
FeSEM(2)	89.1±1.3	64.6±1.0	89.0±1.3	56.0±1.3
FeSEM(3)	88.1±1.9	64.3±0.8	87.5±2.0	55.9±0.8
FeSEM(4)	**93.6**±2.7	**74.8**±1.5	**94.1**±2.2	**69.5**±1.1
FeSEM-MA(3)	84.5±0.8	64.1±0.7	85.1±1.0	63.0±1.3

6.2 Experimental study

6.2.1 Comparison study

As shown in Tables 2 and 3, we compared our proposed FeSEM with the baselines and found that FeSEM achieves the best performance in most cases. But, it is observed that the proposed model achieves an inferior performance for Micro F1 score on the FedCelebA dataset. A possible reason for this is that our objective function defined in (6) does not take into account the number of samples per device as important weights. Hence, our model is able to deliver a significant improvement in terms of "macro" metrics. Furthermore, as shown in the last four rows in Tables 2 and 3, we found that FeSEM with a larger number of clusters empirically achieves a better performance, which verifies the correctness of the non-IID assumption of the data distribution. It is worth noting that the FeSEM-MA is comparable or worsen than FeSEM in our experiment. Because the Matched Averaging method usually works well that each client is randomly initialised while the FedAvg and our FeSEM initialized all clients using the global model. Different initialization strategies will cause different scales of diversity across local models, thus contributing to the performance in different ways.

We also notice that FedAvg and NoFed perform better than some clustered FL baseline. The reason is that our experiment using the original federated learning setting that only split the dataset according to users or by the random split. Therefore, the non-IID and clustering assumption is not obvious in this situation. Therefore, the FedAvg and noFed perform well. In recent, some advanced data split methods have been used to further disturb the non-IID distribution across clients. For example, exaggerate the label distribution across devices by aggregating sample imbalance, or randomly assign different label set to each device. In these new settings, the FedAvg will perform worse, and the clustered FL methods will perform relatively better. Due to the limited space, we will use the original FL setting, and leave the new setting for future work.

Fig. 3 Convergence analysis for the proposed FeSEM with different cluster number (in parenthesis) in terms of micro-accuracy
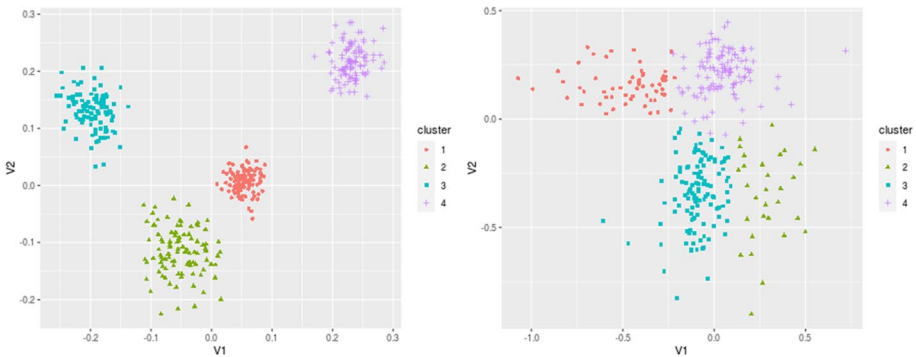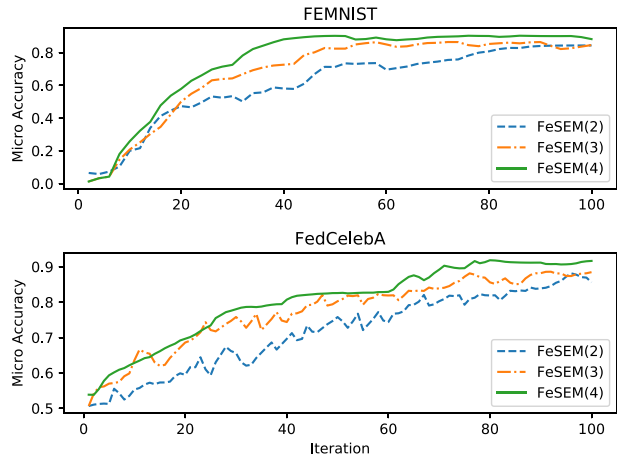

Fig. 4 Clustering analysis for different local models (using PCA) derived from FeSEM(4) using FEMNIST and Celeba data

6.2.2 Convergence

To verify the convergence of the proposed approach, we conducted a convergence analysis by running FeSEM with different cluster numbers K (from 2 to 4) in 100 iterations. As shown in Figure 3, FeSEM can efficiently converge on both datasets and it can achieve the best performance with the cluster number $K = 4$.

6.2.3 Clustering analysis

To check the effectiveness of our proposed optimization method and whether the devices grouped into one cluster have similar model, we conducted a clustering analysis via an illustration. We used two-dimensional figures to display the clustering results of the local models derived from FeSEM(4) on the FEMNIST dataset. In particular, we randomly chose 400 devices from the dataset and plotted each device's local model as one point in the 2D space after PCA dimension reduction. As shown in Figure 4, the dataset suitable for four clusters that are distinguishable to each other. It is worth noting
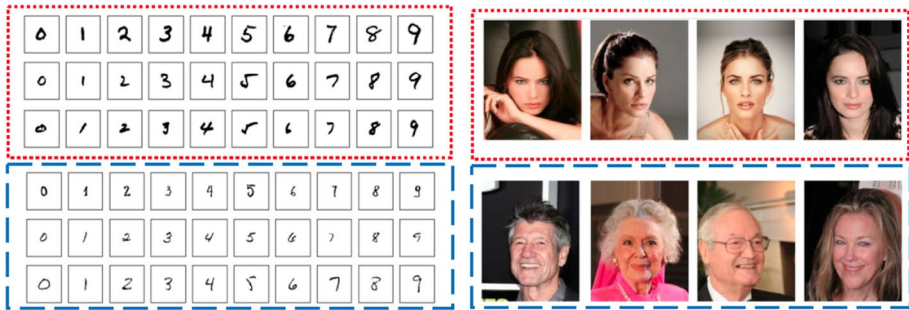
Fig. 5 Cluster demonstrations on FEMINIST (left) and CIFAR-10 (right). Two rectangles represent different cluster discovered by the proposed method

that the clustering algorithm will converge very fast. In general, it takes no more than 10 iterations to converge. One can find detailed theoretical and empirical analyses on the convergence for general clustered FL in a recent work [40].

6.2.4 Case study on clustering

To intuitively judge whether nodes grouped into the same cluster have a similar data distribution, we conducted case studies on a case of two clusters that are extracted from a trained FeSEM(2) model. For FMNIST, as shown on the top of Figure 5, cluster on the right consists writers who are likely to recognize hand-writings with a smaller font, and on the left consists writers who are likely to recognize hand-writing with a bolder and darker font. For FedCelebA, see full face images in Appendix Section 2, the face recognition task in cluster1 is likely to handle the smiling faces with a relatively simple background, also exhibits to be young people. While cluster on the right is likely to handle the faces with more diverse background and also seems to be more older people.

7 Case study

The cluster results are based on the distance metric for client-wise model parameters. The model could be viewed as a map function $f : x \rightarrow y$ that is to map the input x into an output y. The function is parameterised by model parameters w, thus a similar w pair indicate that the two clients are likely to have similar preferences in decision-making. As shown in Figure 5, there are two clusters for FEMNIST and CIFAR-10 datasets respectively. The left part shows the clustering effect of FeSEM on dataset MINIST by writers, on the red rectangle (upper) are three writers handwritten digits which are smaller and lighter than those in the blue rectangle (bottom). The right figure shows the clustering of CIFAR-10 data with 10 classes in which one class is about people. In the class of people, our algorithm finds one cluster to represent young beauties as shown in the red rectangle (upper) and another cluster to represent the aging people in the blue rectangle (bottom).

8 Conclusion

This work proposes a novel FL algorithm to tackle the non-IID challenge, which is towards better personalization of decision-making in non-IID FL. Since the proposed method uses Kmeans as the clustering algorithm, it also suffers the drawbacks of Kmeans, e.g., computational efficiency for high dimension data and robustness to outliers. Moreover, the clustering algorithm usually requires the full participation of all clients in the training process. A more practical clustered federated learning could be developed to address the large-scale federated learning system, and also to explore how to adapt to new clients out of the training set.

Compliance with ethical standards

Conflicts of interest All authors declare that they have no conflicts of interest.

References

1. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv:1912.00818 (2019)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, ??? (2006)
3. Bonawitz, K., Eichner, H., et.al.: Towards federated learning at scale: System design. arxiv:1902.01046 (2019)
4. Cai, T., Li, J., Mian, A.S., Sellis, T., Yu, J.X., et al.: Target-aware holistic influence maximization in spatial social networks. IEEE Transactions on Knowledge and Data Engineering (2020)
5. Caldas, S., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A.: Leaf: A benchmark for federated settings. arXiv:1812.01097 (2018)
6. Cao, T.-D., Truong-Huu, T., Tran, H., Tran, K.: A federated learning framework for privacy-preserving and parallel training. arXiv:2001.09782 (2020)
7. Cappé, O., Moulines, E.: On-line expectation-maximization algorithm for latent data models. Journal of the Royal Statistical Society **71**(3), 593–613 (2009)
8. Chen, F., Long, G., Wu, Z., Zhou, T., Jiang, J.: Personalized federated learning with structure. arXiv:2203.00829 (2022)
9. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: Emnist: Extending mnist to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2921–2926 (2017). IEEE
10. Deng, Y., Kamani, M.M., Mahdavi, M.: Adaptive personalized federated learning. arXiv:2003.13461 (2020)
11. Dinh, T.C., Tran, N., Nguyen, J.: Personalized federated learning with moreauenvelopes. Advances in Neural Information Processing Systems **33**, 21394–21405 (2020)
12. Fintz, M., Osadchy, M., Hertz, U.: Using deep learning to predict human decisions and cognitive models to explain deep learning models. bioRxiv (2021)
13. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. arXiv:1712.07557 (2017)

14. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. arXiv:2006.04088 (2020)
15. Ghosh, A., Hong, J., Yin, D., Ramchandran, K.: Robust federated learning in a heterogeneous environment. arXiv:1906.06629 (2019)
16. Graziani, M., Andrearczyk, V., Marchand-Maillet, S., Müller, H.: Concept attribution: Explaining cnn decisions to physicians. Computers in biology and medicine **123**, 103865 (2020)
17. Haddadpour, F., Mahdavi, M.: On the convergence of local descent methods in federated learning. arXiv:1910.14425 (2019)
18. Hanzely, F., Richtárik, P.: Federated learning of a mixture of global and local models. arXiv:2002.05516 (2020)
19. Hsu, T.-M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv:1909.06335 (2019)
20. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.-L.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv:1811.11479 (2018)
21. Jiang, J., Ji, S., Long, G.: Decentralized knowledge acquisition for mobile internet applications. World Wide Web, 1–17 (2020)
22. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends in Machine Learning **14**(1–2), 1–210 (2021)
23. Konečný, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. arxiv:1610.02527 (2016)
24. Konecný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arxiv:1610.05492 (2018)
25. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. arXiv:1812.06127 (2018)
26. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smithy, V.: Feddane: A federated newton-type method. In: ACSSC, pp. 1227–1231 (2019). IEEE
27. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv:1910.03581 (2019)
28. Li, Q., Wen, Z., He, B.: Federated learning systems: Vision, hype and reality for data privacy and protection. arXiv:1907.09693 (2019)
29. Li, L., Xu, W., Chen, T., Giannakis, G.B., Ling, Q.: Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In: AAAI (2019)
30. Li, Z., Wang, X., Li, J., Zhang, Q.: Deep attributed network representation learning of complex coupling and interaction. Knowledge-Based Systems **212**, 106618 (2021)
31. Liang, P.P., Liu, T., Ziyin, L., Salakhutdinov, R., Morency, L.-P.: Think locally, act globally: Federated learning with local and global representations. arXiv:2001.01523 (2020)
32. Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., Miao, C.: Federated learning in mobile edge networks: A comprehensive survey. IEEE Commun. Surv, Tutor (2020)
33. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. arXiv:2006.07242 (2020)
34. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
35. Liu, Y., Ma, Z., Liu, X., Wang, Z., Ma, S., Ren, K.: Revocable federated learning: A benchmark of federated forest. arXiv:1911.03242 (2019)
36. Long, G., Shen, T., Tan, Y., Gerrard, L., Clarke, A., Jiang, J.: Federated learning for privacy-preserving open innovation future on digital health. In: Humanity Driven AI, pp. 113–133. Springer, ??? (2022)
37. Long, G., Tan, Y., Jiang, J., Zhang, C.: Federated learning for open banking. In: Yang, Q., Fan, L., Yu, H. (eds.) Federated Learning - Privacy and Incentive. Lecture Notes in Computer Science, vol. 12500, pp. 240–254. Springer, ??? (2020). https://doi.org/10.1007/978-3-030-63076-8_17
38. Luo, J., Wu, X., Luo, Y., Huang, A., Huang, Y., Liu, Y., Yang, Q.: Real-world image datasets for federated learning. arXiv:1910.11089 (2019)
39. Lyu, L., Yu, H., Yang, Q.: Threats to federated learning: A survey. arXiv:2003.02133 (2020)
40. Ma, J., Long, G., Zhou, T., Jiang, J., Zhang, C.: On the convergence of clustered federated learning. arXiv:2202.06187 (2022)
41. Mandl, M., Felfernig, A., Teppan, E., Schubert, M.: Consumer decision making in knowledge-based recommendation. Journal of Intelligent Information Systems **37**(1), 1–22 (2011)

42. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. arXiv:2002.10619 (2020)

43. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017)

44. Nichol, A., Schulman, J.: Reptile: a scalable metalearning algorithm. arXiv:1803.02999**2**(2018)

45. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, pp. 325–341. Springer, ??? (2007)

46. Peng, X., Long, G., Shen, T., Wang, S., Jiang, J.: Self-attention enhanced patient journey understanding in healthcare system. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 719–735 (2020)

47. Pomytkina, L., Gudmanian, A., Kovtun, O., Yahodzinskyi, S.: Personal choice: strategic life decision-making and conscience. In: E3S Web of Conferences, vol. 164, p. 10021 (2020). EDP Sciences

48. Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al.: The future of digital health with federated learning. NPJ digital medicine **3**(1), 1–7 (2020)

49. Rouhani, B.D., Riazi, M.S., Koushanfar, F.: Deepsecure: Scalable provably-secure deep learning. In: The 55th Annual Design Automation Conference, p. 2 (2018). ACM

50. Sattler, F., Müller, K.-R., Samek, W.: Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. arXiv:1910.01991 (2019)

51. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The Adaptive Web, pp. 291–324. Springer, ??? (2007)

52. Shamir, O., Srebro, N., Zhang, T.: Communication-efficient distributed optimization using an approximate newton-type method. In: ICML, pp. 1000–1008 (2014)

53. Shrestha, Y.R., Krishna, V., von Krogh, G.: Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. Journal of Business Research **123**, 588–603 (2021)

54. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning over heterogeneous devices. AAAI (2022)

55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)

56. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. In: ICLR (2020)

57. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. Journal of Healthcare Informatics Research **5**(1), 1–19 (2021)

58. Xue, G., Zhong, M., Li, J., Chen, J., Zhai, C., Kong, R.: Dynamic network embedding survey. arXiv:2103.15447 (2021)

59. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. TIST (2019)

60. Yang, C., Wang, Q., Xu, M., Wang, S., Bian, K., Liu, X.: Heterogeneity-aware federated learning. arXiv:2006.06983 (2020)

61. Yu, F., Zhang, W., Qin, Z., Xu, Z., Wang, D., Liu, C., Tian, Z., Chen, X.: Heterogeneous federated learning. arXiv:2008.06767 (2020)

62. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, T.N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. arXiv:1905.12022 (2019)

63. Zhang, Q., Wu, J., Zhang, Q., Zhang, P., Long, G., Zhang, C.: Dual influence embedded social recommendation. World Wide Web **21**(4), 849–874 (2018)

64. Zhang, X., Liu, L., Long, G., Jiang, J., Liu, S.: Episodic memory governs choices: An rnn-based reinforcement learning model for decision-making task. Neural Networks **134**, 1–10 (2021)

65. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv:1806.00582 (2018)

Authors and Affiliations

Guodong Long[1] ⬤ **· Ming Xie[1] · Tao Shen[1] · Tianyi Zhou[2,3] · Xianzhi Wang[1] · Jing Jiang[1]**

> Guodong Long
> guodong.long@uts.edu.au

> Ming Xie
> ming.xie-1@student.uts.edu.au

> Tao Shen
> tao.shen@uts.edu.au

> Tianyi Zhou
> tianyizh@uw.edu

> Xianzhi Wang
> xianzhi.wang@uts.edu.au

[1]   Australian AI Institute, University of Technology Sydney, Sydney, Australia

[2]   University of Washington, Seattle, WA, USA

[3]   University of Maryland, Maryland, MD, USA