Annals of Operations Research
https://doi.org/10.1007/s10479-023-05203-x

ORIGINAL RESEARCH

HFML: heterogeneous hierarchical federated mutual learning on non-IID data

Yang Li[1] · **Jie Li**[1] · **Kan Li**[1]

Accepted: 18 January 2023

Abstract
Non-independent and identical distribution (Non-IID) data and model heterogeneity pose a great challenge for federated learning in cloud-based and edge-based systems. They are easy to lead to inconsistency of gradient updates during the training stage and mismatch of gradient dimensions during the aggregation stage, resulting in the degradation of the global model performance and the consumption of a lot of training time. To solve these problems, this paper proposes a Heterogeneous Hierarchical Federated Mutual Learning (HFML) method in an edge-based system. We design a model assignment mechanism in which clients and edge servers individually fork global models of different structures, and the untrained local models learn mutually with the edge models in deep mutual learning. We use partial periodic aggregation to approximate global aggregation to achieve fast convergence. Our experiments show that HFML obtains state-of-the-art performance than three approaches on common datasets like CIFAR-10/100. Our method improves accuracy up to 2.9% and reduces training time by 30% under homogeneous and heterogeneous models.

Keywords Federated learning · Non-independent and identical distribution (Non-IID) · Heterogeneous models · Deep mutual learning

1 Introduction

The recent development of centralized federated learning has drawn dramatic attention to some powerful computing platforms, e.g., cloud-based and edge-based systems. Federated learning applies in some fields, such as the Internet of Things (IoT), Natural Language Processing (NLP), and Image Processing. A central server accesses massive information from clients with excessive communication overhead in a cloud-based system. In an edge-

✉ Kan Li
likan@bit.edu.cn

Yang Li
liyangsmu@126.com

Jie Li
lijie0610@bit.edu.cn

[1] School of Computer Science of Technology, Beijing Institute Technology, No. 5, South Street, Zhongguancun, Haidian District, Beijing 100081, China

Published online: 02 February 2023

based system, a center server pushes computation resources to the edge servers, which allows clients to jointly train deep models within the communication range (Wang et al., 2019; Liu et al., 2020; Li et al., 2022). In centralized federated learning, each client runs stochastic gradient descent (SGD) locally and a central server aggregates parameter updates from clients for the next round until model convergence. Figure 1a shows the diagram of a round of federated averaging algorithm (FedAvg).

FedAvg (McMahan et al., 2017) is a gradient-based and well-established centralized federated learning algorithm, that allows clients to collaboratively train a model without raw data. When client data is independent and identical distribution (IID), local gradients are unbiased estimates of full gradients, which performs well under standard assumptions (Li et al., 2020; Kairouz & McMahan, 2021). However, this method relies heavily on data quantities and data distribution. When the client data collected from different sources is Non-IID, averaging different local models generates biased gradients and deviates from the true results. To alleviate this impact, the studies (Zhu et al., 2021; Wang et al., 2021) propose some solutions, such as sharing partial private data as public data (Zhao et al., 2018), fine-tuning (Li et al., 2021; Karimireddy et al., 2020) and distillation-based approaches (Jeong et al., 2018; Zhang et al., 2021; Feng et al., 2021). The method of fine-tuning adjusts the weight divergence of local and global models by adding the regulation term to improve model performance, but it's limited by model structures. However, the mismatch of gradient dimensions leads to degrading performance during the aggregation stage when different clients design different network structures according to computing power (Li & Wang, 2019). Model complexity affects the learning ability of the model and is affected by model size and data distribution (Mohri et al., 2018; Hu et al., 2021). Distillation-based approaches can compress the model's size and improve the performance of small models, which starts with a large and pre-trained teacher model and trains a smaller student model, which isn't limited by model structures, but the performance of the student model can't outperform the teacher model. Deep mutual learning (DML) (Zhang et al., 2018) isn't a one-way transfer method between static teacher and student models and is integrated with federated learning (Li et al., 2022; Shen et al., 2020), which ensembles of student models and learns collaboratively, and all model parameters are updated throughout the training process. In Fig. 1b, the client forks the initial global model as a meme model for local mutual training, and uploads a meme model to the cloud server.

In this paper, we propose a Heterogeneous Hierarchical Federated Mutual Learning (HFML) method in the edge-based framework. We introduce deep mutual learning to mine knowledge from local data and use partial aggregation to guide local updates per client when local and edge models are heterogeneous. Our contributions are listed as follows:

- We propose an edge-based federated learning framework and design a model assignment mechanism that allows the client frequently performs the local update and transfers local knowledge with the edge model by deep mutual learning through the edge layer to achieve fast convergence.
- We develop an easy-to-implement heterogeneous hierarchical federated mutual learning method named HFML in an edge-based system. We leverage partial model aggregation to reduce the number of local iterations and training time while maintaining stable accuracy when client data is Non-IID data.
- We conduct multiple experiments in Non-IID settings for image classification. The results show that HFML outperforms FedAvg, FedProx, and FML methods on metrics such as accuracy, training time, and model complexity.

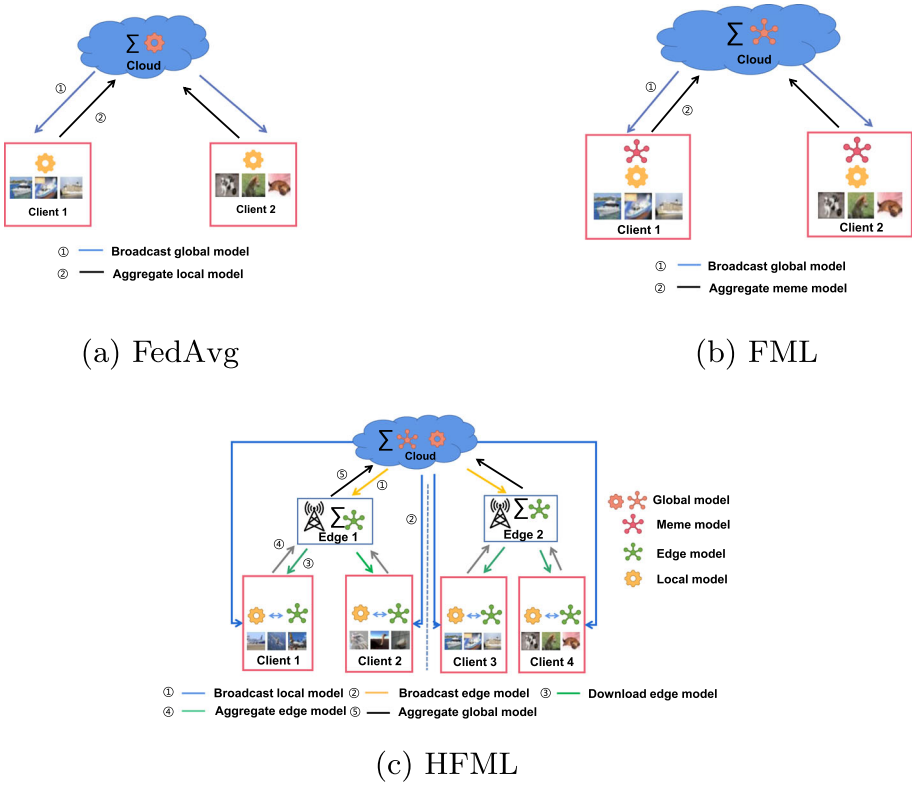


Fig. 1 Three centralized federated learning frameworks. **a** FedAvg **b** FML **c** HFML

In this paper, Sect. 2 reviews the relevant studies on Non-IID data and model heterogeneity. Section 3 describes related preliminaries. We propose the HFML scheme in Sect. 4. Section 5 describes the experiments and analyzes the results. In Sect. 6, we summarize the content of this paper.

2 Related work

2.1 Non-IID data

Data partition strategies are used to simulate real-world data distributions, including Dirichlet distribution, skewed feature distribution, skewed label distribution, and quantity imbalance (Caldas et al., 2018; Li et al., 2021; Hsieh et al., 2020; Hsu et al., 2020). Note that Non-IID means that local distribution hardly represents global distribution. Local and global models can be regarded as containers of knowledge and rely heavily on massive data and data distribution. In ref (Li et al., 2019), the convergence of the FedAvg algorithm for strongly convex problems in the Non-IID setting is proved. Fine-tuning reduces weight divergence discrepancy between local and global models with the same structure (Munir et al., 2022; Li et al., 2018; Karimireddy et al., 2020). Aggregating multiple local models trained on Non-

IID data is affected by inconsistent updates, which reduces model accuracy and convergence speed.

2.2 Model heterogeneity

Model heterogeneity reflects differences in data presentation and learning ability. Knowledge distillation integrated with federated learning is used to compress models (Hinton et al., 2015; Anil et al., 2018; Seo et al., 2020; Chan et al., 2021; Jiang et al., 2020; Afonin et al., 2022; Yu et al., 2022), which transfers knowledge from large models to small models and is suited to the low-memory device. FedGKD (Pan & Sun, 2021) fuses global historical information to guide local models and weakens the over-fitting of local models. FedUFO (Zhang et al., 2021) addresses optimization inconsistency and feature divergence issues by modifying two consensus losses and extracting group data information from global and local models. FedHeNN (Makhija et al., 2022) allows agnostic architecture across peer clients and guides the simultaneous training on each client.

2.3 Periodic aggregation

Periodic model aggregations reduce the communication cost in an edge-based system. Increasing parallel computing on clients can reduce communication times in a centralized federated learning framework (Konecny et al., 2016; Rothchild et al., 2020; Matsuda et al., 2022). Lin et al. (2018) proves that 99% of gradient exchange is redundant in the communication process, and the exchange of a large number of parameters increases the unnecessary communication cost and extends the aggregation period. Tier-based federated learning is segmented according to privacy levels and model performance to accelerate convergence under data heterogeneity (Wu et al., 2021; Chai et al., 2020, 2021; Mhaisen et al., 2022; Luo et al., 2020). HierFAVG (Liu et al., 2020) allows multiple edge servers to perform partial model aggregation on an edge-based system. For periodic aggregation optimization, FedBCD (Liu et al., 2019) presents that each client performs different local updates before uploading parameters to adjust the update direction. Ref (Lee et al., 2022) proposes a partial model averaging method to solve the problem of slow convergence due to model discrepancy across the clients.

3 Preliminaries

In this section, we describe the federated averaging algorithm and model complexity. Then, we introduce the deep mutual learning method and related federated learning schemes.

3.1 Federated averaging algorithm

Suppose the network includes K clients. Given dataset $S = \{x_i, y_i\}_{i=1}^{n}$ includes n samples of M classes, the k-th client holds the n_k samples over data distributions $D_k(x_i, y_i)$, $n = \sum_{k=1}^{K} n_k$, $p_k = \frac{n_k}{n}$. The global objective function $f(w)$ formulates as follow:

$$min_w f(w) = \sum_{k=1}^{K} p_k F_k(w) \tag{1}$$

$$F_k(w^k) = \frac{1}{n_k} \sum_{i=1}^{K} f_i(w^k) \tag{2}$$

where $F_k(\cdot)$ is local objective function of k-th client, $f_i(w^k) = l(x_i, y_i; w^k)$ represents loss function on samples n_k made with weight parameters of local model w^k.

$$w^k \leftarrow w^k - \eta \nabla F_k(w^k) \tag{3}$$

$$w^{global} \leftarrow \sum_{k=1}^{K} p_k w^k \tag{4}$$

where w^{global} is the weight of the global model. Note that when D_k is IID data, $\mid F_{SUM} - F_{FED} \mid \leq \delta$ and $E_{D_K}[F_k(w)] = f(w)$ holds, where federated learning performance approximates centralized computing, δ is a non-negative real number (Yang et al., 2019). Clients update local parameters by SGD method and aggregate local models at a server, performing the above operations to update the global model until converges during the whole process.

3.2 Deep mutual learning

Deep mutual learning (Zhang et al., 2018) can be viewed as bidirectional knowledge transfer between student networks and is suitable for training on heterogeneous models. At each iteration, we compute the predictions of the two models and update both models' parameters according to the predictions of the other.

Suppose there are two models θ_1 and θ_2. For multi-class image classification task, the probability of class m for sample x_i by model θ_1 is computed as

$$p_1^m(x_i) = \frac{exp(z_1^m)}{\sum_{m=1}^{M} exp(z_1^m)} \tag{5}$$

where the logit z_1^m is the output of the softmax layer in model θ_1. $p_1(\cdot)$ represents the prediction of the model θ_1, named soft targets. Deep mutual learning includes two losses: a conventional supervised learning cross-entropy loss L_{CE} between the hard labels and the soft targets and a mimicry loss $D_{KL}(\cdot)$, named Kullback Leibler (KL) Divergence which quantifies the match of the soft predictions.

$$L_{CE} = -\sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i) log(p_i^m(x_i)) \tag{6}$$

$$D_{KL}(p_2 \| p_1) = \sum_{i=1}^{K} \sum_{m=1}^{M} p_2^m(x_i) log \frac{p_2^m(x_i)}{p_1^m(x_i)} \tag{7}$$

where I is an indicator function. $I(y_i) = \begin{cases} 0 & y_i = m \\ 1 & y_i \neq m \end{cases}$, loss functions of model θ_1 and θ_2 can be computed as.

Table 1 Notations

S_{edge}, S_{client}	Edge/clients set
$z_{C_i}, z_{C_{i_j}}$	Edge/local model logit
$p_{C_i}, p_{C_{i_j}}$	Edge/local probability prediction
$L_{C_i}, L_{C_{i_j}}$	Edge/local model loss function
$w_{global}, w^m_{edge}, w^k_{local}$	Global/edge/local model parameter
C_i, C_{i_j}	Edge/local model

$$L_{\theta_1} = L_{C_1} + D_{KL}(p_2 \| p_1) \tag{8}$$
$$L_{\theta_2} = L_{C_2} + D_{KL}(p_1 \| p_2) \tag{9}$$

Deep mutual learning is integrated with federated learning from invisible data to learn knowledge. In FML (Shen et al., 2020), the meme model as a medium between the global models and the local models is used to solve the problem of data, objective, and model heterogeneity (DOM) in Fig. 1b. Student models can train mutually instead of learning from the pre-trained teacher model. The loss functions $L(\cdot)$ of C_{local} and C_{meme} describe as follows:

$$\begin{aligned} L_{local} &= \alpha L_{C_{local}} + (1-\alpha) D_{KL}(p_{meme} \| p_{local}) \\ L_{meme} &= \beta L_{C_{meme}} + (1-\beta) D_{KL}(p_{local} \| p_{meme}) \end{aligned} \tag{10}$$

where α and β are the hyper-parameters which use to control the proportion of knowledge transfer from data or model. When $\beta = 1$, the federated mutual learning algorithm would degrade into a typical federated averaging algorithm.

4 Methodology

In this section, we describe the Heterogerous Hierarchical Federated Mutual Learning (HFML) method in an edge-based federated learning system which includes cloud servers, edge servers, and clients. Fig. 1c shows the diagram of a round of HFML.

4.1 Formulation

Edge servers are denoted by $S_{edge} = \{m_i, i = 1, \cdots, M\}$ and clients are denoted by $S_{client} = \{c_{i_j}, i_j = 1, \cdots, N\}$. The edge models are marked as $\{C_i, i = 1, \cdots, N\}$. The client models connected to edge server m_i are marked as $\{C_{i_j}, i_j = 1, \cdots, K\}$, c_{i_j} represents j-th client connected to the i-th edge server and each edge server connects same number of clients (Table 1).

The rounds are marked as D. The global communication round sets T between cloud and edge servers, and the partial communication round sets t between clients and edge servers, the local epoch sets E. $p_{C_{i_j}}$ is computed as Eq. 5 and P_I represents periodic predictions aggregation.

$$p_I = \begin{cases} \frac{1}{K}\sum_{i_j=1}^{K} p_{C_{i_j}} & D \mod E = 0 \\ p_{C_{i_j}} & D \mod E \neq 0 \end{cases}$$

We rewrite the edge model loss function L_{C_i} and local model loss function $L_{C_{i_j}}$ as follows:

$$\begin{aligned} L_{C_{i_j}} &= L_{C_{i_j}} + D_{KL}(p_{C_{i_j}} \| p_I) \\ L_{C_i} &= \alpha L_{C_i} + \beta D_{KL}(p_I \| p_{C_{i_j}}) \end{aligned} \tag{11}$$

where $L_{C_{i_j}}$ and L_{C_i} are computed by Eq. 7. The hyper-parameters are used to adjust the strength of learning ability from local data, defaulting to 0.5 for all experiments. The edge and local models conduct DML and update model parameters.

$$w_{edge}^{m+1} \leftarrow w_{edge}^{m} - \eta \nabla L_{C_i}(w_{edge}^{m}, w_{local}^{k}) \tag{12}$$

$$w_{local}^{k+1} \leftarrow w_{local}^{k} - \eta \nabla L_{C_{i_j}}(w_{edge}^{m}, w_{local}^{k}) \tag{13}$$

Partial periodic aggregation at the edge layer and global model aggregation in the cloud are as follows.

$$w_{edge}^{m_i} \leftarrow \sum_{c_{i_j}=1}^{K} w_{local}^{c_{i_j}} \tag{14}$$

$$w_{global} \leftarrow \sum_{m_i=1}^{M} w_{edge}^{m_i} \tag{15}$$

In Fig. 1c, the edge servers download the global models (homogeneous or heterogeneous) from the cloud as edge models, and the client downloads the edge model and trains mutually and uploads the edge model to the edge layer. Finally, the cloud aggregates local models into a global model. HFML is compatible with heterogeneous models on Non-IID data, where the edge layer acts as a knowledge transfer hub between connected clients in Algorithm 1

5 Experiments

5.1 Models and datasets

We conduct extensive experiments on CIFAR-10 (Krizhevsky et al., 2014) and CIFAR-100 (Krizhevsky et al., 2009) datasets, which are widely used in image classification task. CIFAR-10 consists of 50000 training images and 10000 test images in 10 classes, with 5000 and 1000 images per class. CIFAR-100 has the same total number of images as CIFAR-10, but it has 100 classes. All images of CIFAR-10/100 are 3-channel 32x32 RGB images. We simulate two settings by sorting and assigning label classes.

Global and local models include combinations of convolution neural network (CNN1, CNN2, and Multi-Layer Perceptron (MLP)). CNN1 is a convolution neural network with two 3x3 convolution layers (the first with 6 channels, the second with 16 channels, each followed with 2x2 max pooling and ReLu activation) and two fully connected layers, and a convolution neural network CNN2 with three 3x3 convolution layers (each with 128 channels followed with 2x2 max pooling and ReLu activation) and one fully connected layer. MLP

Algorithm 1 HFML Algorithm.

Input: S_{client}, S_{edge}, w^0_{global}, w^1_{global}.
Parameters: local epochs E and edge epochs T.
Output: w_{global}.
Cloud Server Executes
Init: global model w^0_{global} and w^1_{global}
Edge Server Executes
Fork: edge model $w_{edge} \leftarrow w^0_{global}$
for each edge $m \in S_{edge}$ **do**
  for each round $t = 1, \cdots, T$ **do**
    for each client $k \in S_{client}$ **in parallel do**
      $w_{local,t+1}, w_{edge,t+1} \leftarrow$**Updateclient**$(w_{local,t}, w_{edge,t})$
    end for
  end for
  Merge: $w_{edge} \leftarrow \sum w_{edge,t}$
end for
Merge: $w_{global} \leftarrow \sum w_{edge}$

Updateclient$(w_{local,t}, w_{edge,t})$
Fork: local model $w_{local} \leftarrow w^1_{global}$
Download: edge model w_{edge}
for each round $n = 1, \cdots, E$ **do**
  conduct *DML* according to Eq. 12 and Eq. 13.
end for

is a special convolution neural network with three fully connected layers that contain the nonlinear activation function ReLU. CNN1/CNN2 represents that the global model is CNN1 and the local models are CNN2; CNN2/CNN1 represents the global model is CNN2 and local models are CNN1; CNN1 represents the global model and local models are CNN1, and CNN2 represents that the global model and local models are CNN2, MLP/CNN2 represents that global model is MLP and local models are CNN2; CNN2/MLP represents that global model is CNN2 and local models are MLP.

5.2 Experiment settings

For a fair comparison, our experiment performs image classification tasks on CIFAR-10/100 datasets with Non-IID settings of skewed label partitions. We compare proposed HFML with FedAvg (McMahan et al., 2017), FML (Shen et al., 2020), and FedProx (Li et al., 2020) schemes under the same conditions. The metrics include accuracy, training time, and model size. We consider an edge-based federated learning system and assume each edge server connects the same number of clients. The total communication round sets T between cloud and edge servers, and partial communication round sets t between clients and edge servers, and local epochs E. Two settings mean that each client has overlap label classes, such as label classes and sample size are similar {3:3:4} for CIFAR-10, and label classes and sample size are similar {30:30:40} for CIFAR-100, marked as setting 1; label classes and sample size are large difference {6:2:2} for CIFAR-10, and label classes and sample size are large difference {60:20:20} for CIFAR-100, marked as setting 2. The parameters include momentum = 0.9, weight_decay = 5×10^{-4}, learning rate $\eta = 10^{-3}$ and batch size $B = 128$.

Table 2 Training time of four approaches in setting 1

CIFAR-10	CNN1	CNN2	MLP	CNN1/CNN2	CNN2/CNN1
FedAvg	6h22min	6h28min	6h31min	–	–
FedProx	7h4min	7h12min	6h47min	–	–
FML	7h34min	7h18min	7h27min	7h39min	7h38min
HFML	**4h25min**	**4h30min**	**4h28min**	**4h23min**	**4h39min**
CIFAR-100	CNN1	CNN2	CNN1/CNN2	CNN2/CNN1	CNN2/MLP
FedAvg	6h47min	6h30min	–	–	–
FedProx	7h12min	7h21min	–	–	–
FML	7h30min	7h37min	7h32min	7h11min	7h16min
HFML	**4h15min**	**5h38min**	**5h11min**	**4h26min**	**4h42min**

Bold values indicate the best results

5.3 Results

To compare the performance of the proposed method and the baseline methods, we run multiple experiments using the homogeneous and heterogeneous models in all settings and evaluate the training time, accuracy, and model complexity. In this paper, the accuracy rate recorded in all figures is the best value for each round in the training stage, once it exceeds the existing best value, and it is recorded and updated, otherwise, it remains unchanged.

5.3.1 Comparison of training time

We use deep mutual learning to improve performance during the training phase and partial period aggregations to approximate global aggregations during the inference phase. Table 2 shows that HFML spends less training time than FedAvg, FedProx, and FML methods on settings 1 under homogeneous and heterogeneous models with the same conditions. While the training time between different algorithms under homogeneous models on the same dataset varies widely, HFML can reduce training time up to 30%. The training time of the same method on datasets of different sizes is affected, such as CIFAR-10 and CIFAR-100. The difference in the training time is only a few minutes under different models using the same algorithm, such as MLP, CNN1, and CNN2.

5.3.2 Accuracy comparison

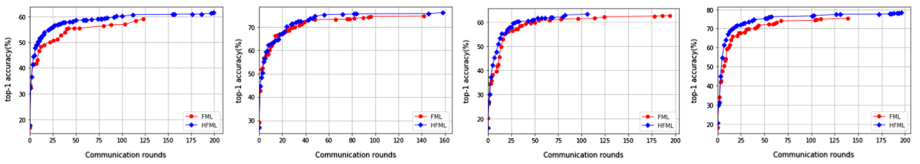
As shown in Table 3, the accuracy of CNN2 is higher than CNN1 and MLP for four approaches under homogeneous models in all settings. When global model and local models are heterogeneous on CIFAR-10/100, such as CNN1/CNN2, CNN2/CNN1, and CNN2/MLP, the accuracy of HFML has 2.9% improvement than FML on CIFAR-10/100 in Figs. 2 and 3, and improves 2.04% accuracy than other approaches when global model and local models are MLP in setting 1. The model performance is related to the degree of label skew and the model structure. When the global and local models are homogeneous, the accuracy of the deep global model (CNN2) is higher than that of the shallow model (CNN1, MLP).

FedAvg and FedProx train failure between local and global models due to mismatch of gradient dimension. FedAvg depends on initializing the global model and updates model weights

Table 3 Top-1 accuracy (%) of the global model on CIFAR-10/100 using homogeneous and heterogeneous models in setting 1 and 2 for four approaches

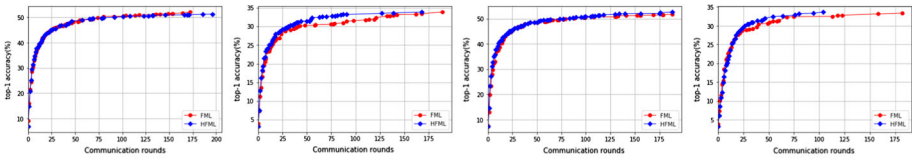
	CIFAR10							CIFAR100				
Edge model	CNN1	CNN2	CNN1	CNN2	MLP	CNN2	MLP	CNN1	CNN2	CNN1	CNN2	CNN2
Local models	CNN2	CNN1	CNN1	CNN2	MLP	MLP	CNN2	CNN1	CNN2	CNN2	CNN1	MLP
Setting(1)												
FedAvg	×	×	58.28	75.54	45.95	×	×	33.16	50.36	×	×	×
FedProx	×	×	51.48	72.15	42.83	×	×	27.60	51.48	×	×	×
FML	59.06	74.82	60.52	76.01	47.64	72.32	51.99	33.07	51.86	33.85	52.1	49.32
HFML	**61.46**	**76.29**	**60.71**	**76.08**	**49.68**	**72.33**	**53.59**	**33.65**	**53.92**	**33.87**	**52.29**	**50.44**
Setting(2)												
FedAvg	×	×	59.09	78.95	45.38	×	×	33.13	50.36	×	×	×
FedProx	×	×	44.84	68.45	44.08	×	×	31.92	49.51	×	×	×
FML	63.15	75.40	62.72	79.53	49.41	75.27	51.80	32.40	54.90	33.39	51.70	49.69
HFML	**65.61**	**78.30**	**63.27**	**80.09**	**50.89**	**76.93**	**53.15**	**33.65**	**55.57**	**33.7**	**52.60**	**51.36**

Bold values indicate the best results



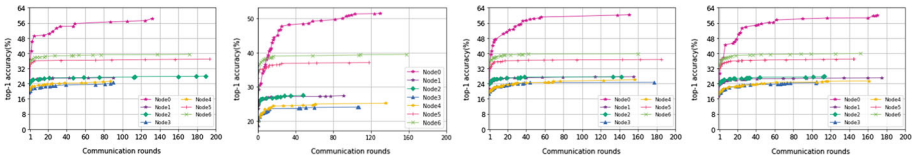
(a) CNN1/CNN2 (b) CNN2/CNN1 (c) CNN1/CNN2 (d) CNN2/CNN1

Fig. 2 The best accuracy on CIFAR-10 under heterogeneous models. **a** The global model is CNN1 and local models are CNN2 in setting 1. **b** The global model is CNN2 and local models are CNN1 in setting 1. **c** The global model is CNN1 and local models are CNN2 in setting 2. **d** The global model is CNN2 and local models are CNN1 in setting 2



(a) CNN1/CNN2 (b) CNN2/CNN1 (c) CNN1/CNN2 (d) CNN2/CNN1

Fig. 3 The best accuracy on CIFAR-100 under heterogeneous models. **a** The global model is CNN1 and local models are CNN2 in setting 1. **b** The global model is CNN2 and local models are CNN1 in setting 1. **c** The global model is CNN1 and local models are CNN2 in setting 2. **d** The global model is CNN2 and local models are CNN1 in setting 2



(a) FedAvg (b) FedProx (c) FML (d) HFML

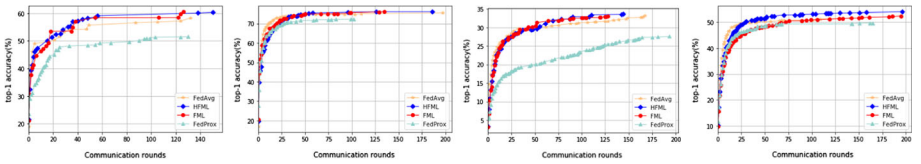
Fig. 4 The accuracy of four methods on CIFAR-10 under heterogeneous models in setting 1. Node 0 represents as a global model and Node 1-6 respectively represent as local models

according to sample size at clients. FedProx modifies local and global updated weight by adding a regularization term. FML aggregates meme models at the cloud server and controls the proportion of the rate of data and model by hyper-parameters. HFML adjusts knowledge fusion by partial periodic aggregation edge models to approximate global aggregation in an edge-based system.

Figure 4 describes the accuracies of a global model and six local models on CIFAR-10 when they are heterogeneous models (CNN1/CNN2, CNN2/CNN1, MLP/CNN2, and CNN2/MLP). HFML improves the accuracy of both the global model and the local models. The local performance is affected by the local data distribution and label classes, while the global model is affected by the distribution gap between clients.

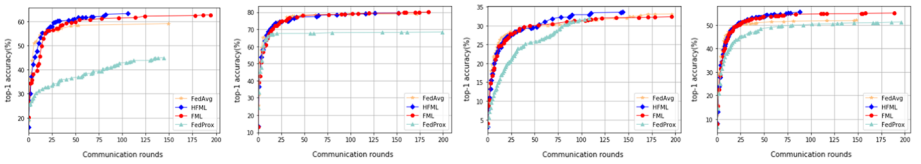
Figures 5 and 6 show that the best accuracy of the global model for homogeneous models in setting 1 and setting 2 of four algorithms using CNN1 and CNN2, it can be seen that global classification accuracy of the proposed method is higher than that of the baseline methods. Figure 7a and b show that HFML achieves the best results on all cases for homogeneous models (MLP) of four algorithms.

Figure 7c and d show that HFML improves the accuracy when global and local models are heterogeneous. In Fig. 8a and b, the results show that the accuracy is affected by hyper-



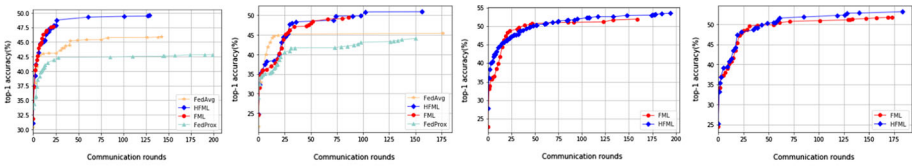
(a) CNN1/10 (b) CNN2/10 (c) CNN1/100 (d) CNN2/100

Fig. 5 The best accuracy of global model on CIFAR-10/100 under homogeneous models in setting 1. **a** The global model and local models are CNN1 on CIFAR-10. **b** The global model and local models are CNN2 on CIFAR-10. **c** The global model and local models are CNN1 on CIFAR-100. **d** The global model and local models are CNN2 on CIFAR-100



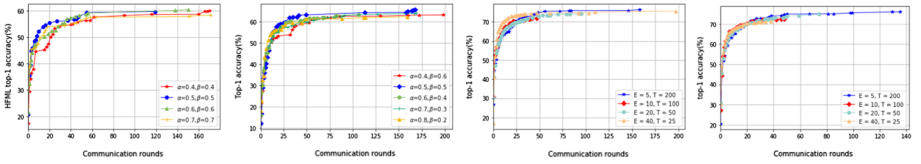
(a) CNN1/10 (b) CNN2/10 (c) CNN1/100 (d) CNN2/100

Fig. 6 The best accuracy of the global model on CIFAR-10/100 under homogeneous models in setting 2. **a** The global model and local models are CNN1 on CIFAR-10. **b** The global model and local models are CNN2 on CIFAR-10. **c** The global model and local models are CNN1 on CIFAR-100. **d** The global model and local models are CNN2 on CIFAR-100



(a) MLP/1 (b) MLP/2 (c) MLP/CNN2/1 (d) MLP/CNN2/2

Fig. 7 The best accuracy of the global model on CIFAR-10. **a** The global model and local models are MLP in setting 1. **b** The global model and local models are MLP in setting 2. **c** The global model is MLP and local models are CNN2 in setting 1. **d** The global model is MLP and local models are CNN2 in setting 2



(a) CNN1 (b) CNN2/CNN1 (c) CNN1/CNN2 (d) CNN2

Fig. 8 The best accuracy of HFML with hyper-parameters on CIFAR-10 using homogeneous and heterogeneous models. **a** The accuracy of homogeneous models in setting 1. **b** The accuracy of heterogeneous models in setting 2. **c** The accuracy of partial aggregations in setting 1. **d** The accuracy of partial aggregations in setting 1

Table 4 Model complexity on Non-IID settings

Model_Dataset	Parameters	MAdd	Flops	MemR+W	Memory
Setting1/2 _CIFAR10					
CNN1	38626	660.29K	340.13K	215.41K	153KB
CNN2	303882	60.79M	30.54M	1.84M	1189KB
MLP	656810	1.31M	656.4K	2.52M	2567KB
FedAvg/FedProx_CNN1_CNN2	–	–	–	–	–
FML/HFML CNN1_CNN2	38626	660.29K	340.13K	215.41K	153KB
FedAvg/FedProx_CNN2_CNN1	–	–	–	–	–
FML/HFML CNN2_CNN1	303882	60.79M	30.54M	1.84M	1189KB
FML/HFML CNN2_MLP	303882	60.79M	30.54M	1.84M	1189KB
FML/HFML MLP_CNN2	656810	1.31M	656.4K	2.52M	2567KB
Setting1/2_CIFAR100					
CNN1	44476	671.72K	345.89K	238.62K	176KB
CNN2	350052	60.88M	30.58M	2.02M	1369KB
FedAvg/FedProx_CNN1_CNN2	–	–	–	–	–
FML/HFML CNN1_CNN2	44476	671.72K	345.89K	238.62K	176KB
FedAvg/FedProx_CNN2_CNN1	–	–	–	–	–
FML/HFML CNN2_CNN1	350052	60.88M	30.58M	2.02M	1369KB
FML/HFML CNN2_MLP	350052	60.88M	30.58M	2.02M	1369KB

parameters but slightly. The edge layer acts as a knowledge transfer hub between clients to guide local updates. We consider local epochs and partial periodic aggregation relatedness in Fig. 8c and d. The results show that the accuracy decreases as the rounds of edge aggregations.

5.3.3 Model complexity comparison

We run the code in the same hardware and software environments. In Table 4, we found that model parameters are related to model structures and dataset size, and different algorithms in the same setting train under the same models to have the same numbers of model parameters. For example, the global model parameters of CNN2/MLP are the same as CNN2, only the local model parameters are different. The number of parameters of CNN2 is 10 times that of CNN1. When the global model size is larger than the local models, such as CNN2/MLP or CNN2/CNN1, the accuracy of the global model approximates CNN2 and consumes lower resources, respectively 2.11M, 11.22M, and 17.55M, which reduces gradients size during aggregation stage. FedAvg and FedProx train collaboratively a global model under homogeneous models. FML and HFML focus on training on Non-IID data under homogeneous and heterogeneous models.

5.4 Discussion

In this paper, we discuss three aspects: data distribution, model structures, and training time. We compare performance of four methods from the perspective of data heterogeneity under heterogeneous and homogeneous models.

5.4.1 Impact of non-IID data

HFML focus on training on Non-IID data under homogeneous and heterogeneous models. Fine-tuning mainly occurs in the training phase and depends on model structures, and deep mutual learning can transfer the knowledge of the last layer. Performance is affected by model structures and data distribution during local multiple iterations and global aggregations. The significant update deviation causes the global model to deviate from the true optimization results.

5.4.2 Impact of heterogeneous models

Models are regarded as containers for storing knowledge from different data, and model complexity is affected by model structures, model size, data distribution, and dataset size. Increasing the number of hidden units or parameters, which leads to generalization errors. When different algorithms train the model on the same condition, such as the model complexity can be measured using LANN (Hu et al., 2020), e.g., $CNN2 > MLP > CNN1$, the accuracy of the deep model is better than a shallow model, but training time is longer. Table 4 shows that model complexity depends on global model structure and parameters, such as CNN2 has the same model parameters as CNN2_CNN1. Deep mutual learning transfers bidirectionally knowledge and is suitable for heterogeneous models. When the global model size is smaller than the local model, the accuracy and storage of the global model are better than that of the homogeneous model CNN1, but it loses accuracy more than CNN2 and can't trade off training time and accuracy.

5.4.3 Impact of edge layer

We set the same number of clients connected to the edge server in an edge-based system. The hyper-parameters α and β are regarded as the knowledge transfer rate of data and models.

6 Conclusion

In this work, we propose a method named heterogeneous hierarchical federated mutual learning (HFML) in an edge-based system, which solves the problems of the inconsistency of gradient updates during the training stage and mismatch of gradient dimensions during the aggregation stage in Non-IID settings. We use deep mutual learning to transfer and jointly mine invisible knowledge from local models and edge models and achieve model updates. We leverage partial aggregations to achieve fast convergence and reduce training time. In terms of accuracy and training time, HFML outperforms than FedAvg, FedProx, and FML schemes.

Acknowledgements This research was supported by Beijing Natural Science Foundation, China (Nos. 4222037, L181010) and National Key R & D Program of China (No. 2016YFB0801100).

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

Afonin, A., Karimireddy, S.P. (2022). Towards model agnostic federated learning using knowledge distillation. In: International conference on learning representations (ICLR) (2022). https://openreview.net/forum?id=lQI_mZjvBxj

Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G.E., Hinton, G.E. (2018). Large scale distributed neural network training through online distillation. In: International conference on learning representations (ICLR). https://openreview.net/forum?id=rkr1UDeC-

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H.B., Smith, V., Talwalkar, A. (2018). LEAF: A benchmark for federated settings. CoRR arXiv:1812.01097

Chai, Z., Ali, A., Zawad, S., Truex, S., Anwar, A., Baracaldo, N., Zhou, Y., Ludwig, H., Yan, F., Cheng, Y. (2020). TiFL: A tier-based federated learning system. In: Proceedings of the 29th international symposium on high-performance parallel and distributed computing. HPDC' 20, pp. 125–136, New York, NY, USA. https://doi.org/10.1145/3369583.3392686

Chai, Z., Chen, Y., Anwar, A., Zhao, L., Cheng, Y., Rangwala, H.: FedAT: A high-performance and communication-efficient federated learning system with asynchronous tiers. In: Proceedings of the international conference for high performance computing, networking, storage and analysis. SC' 21, New York, NY, USA (2021). https://doi.org/10.1145/3458817.3476211

Chan, Y.H., Ngai, E.C.H. (2021). FedHe: Heterogeneous models and communication-efficient federated learning. In: 17th International Conference on Mobility, Sensing and Networking (MSN), pp. 207–214. https://doi.org/10.1109/MSN53354.2021.00043

Feng, S., Chen, H., Ren, X., Ding, Z., Li, K., Sun, X. (2021). Collaborative group learning. In: Proceedings of the AAAI conference on artificial intelligence, vol.35, pp. 7431–7438. https://ojs.aaai.org/index.php/AAAI/article/view/16911

Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the knowledge in a neural network. In: Conference on neural information processing systems (NIPS) Workshop. arXiv:1503.02531

Hsieh, K., Phanishayee, A., Mutlu, O., Gibbons, P. (2020). The non-iid data quagmire of decentralized machine learning. In: Proceedings of the 37th international conference on machine learning (ICML), vol.119, pp. 4387–4398, https://proceedings.mlr.press/v119/hsieh20a.html

Hsu, H., Qi, H., Brown, M. (2020). Federated visual classification with real-world data distribution. In: European conference on computer vision arXiv:2003.08082

Hu, X., Liu, W., Bian, J., & Pei, J. (2020). Measuring model complexity of neural networks with curve activation functions. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD' 20, pp. 1521–1531, New York, NY, USA, https://doi.org/10.1145/3394486.3403203

Hu, X., Chu, L., Pei, J., Liu, W., & Bian, J. (2021). Model complexity of deep learning: A survey. *Knowledge and Information Systems, 63*(10), 2585–2619.

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.(2018). Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. CoRR arXiv:1811.11479

Jiang, D., Shan, C., Zhang, Z. (2020). Federated learning algorithm based on knowledge distillation. In: International conference on artificial intelligence and computer engineering (ICAICE), pp. 163–167. https://doi.org/10.1109/ICAICE51518.2020.00038

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

Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: SCAFFOLD: Stochastic controlled averaging for federated learning. In: Proceedings of the 37th international conference on machine learning, vol.119, pp. 5132–5143 (2020). https://proceedings.mlr.press/v119/karimireddy20a.html

Konečný, J., McMahan, H.B., Ramage, D., Richtarik, P.(2016). Federated Optimization: Distributed machine learning for on-device intelligence. arXiv:1610.02527

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Nair, V., Hinton, G. (2014). The CIFAR-10 dataset. http://www.cs.toronto.edu/kriz/cifar.html

Lee, S., Sahu, A.K., He, C., & Avestimehr, S. (2022). Partial model averaging in federated learning: Performance guarantees and benefits. In: AAAI 2022 Workshop on Trustable, verifiable and auditable federated learning (FL-AAAI-22).

Li, D., & Wang, J. (2019). Fedmd: Heterogenous federated learning via model distillation. CoRR arxiv:1910.03581

Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. In: IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978 (2022). https://doi.org/10.1109/ICDE53745.2022.00077

Li, Q., He, B., Song, D. (2021). Model-contrastive federated learning. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 10708–10717 https://doi.org/10.1109/CVPR46437.2021.01057

Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z. (2020). On the convergence of fedavg on non-iid data. In: International conference on learning representations.(ICLR) https://openreview.net/forum?id=HJxNAnVtDS

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V. (2020). Federated optimization in heterogeneous networks. In: Proceedings of machine learning and systems (MLSys), vol. 2, pp. 429–450 https://proceedings.mlsys.org/paper/2020/file/38af86134b65d0f10fe33d30dd76442e-Paper.pdf

Li, Z., He, Y., Yu, H., Kang, J., Li, X., Xu, Z., & Niyato, D. (2022). Data heterogeneity-robust federated learning via group client selection in industrial iot. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2022.3161943.

Li, C., Li, G., & Varshney, P. K. (2022). Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal, 9*(2), 1136–1147. https://doi.org/10.1109/JIOT.2021.3078543.

Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W.J.: Deep gradient compression: Reducing the communication bandwidth for distributed training. In: International conference on learning representations (ICLR) (2018). https://openreview.net/forum?id=SkhQHMW0W

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37*(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749.

Liu, L., Zhang, J., Song, S.H., & Letaief, K.B. (2020) Client-edge-cloud hierarchical federated learning. In: IEEE International Conference on Communications (ICC), pp. 1–6. https://doi.org/10.1109/ICC40277.2020.9148862

Liu, Y., Zhang, X., Kang, Y., Li, L., Chen, T., Hong, M., & Yang, Q. (2022). FedBCD: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing*. https://doi.org/10.1109/TSP.2022.3198176.

Luo, S., Chen, X., Wu, Q., Zhou, Z., & Yu, S. (2020). Hfel: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning. *IEEE Transactions on Wireless Communications, 19*(10), 6535–6548. https://doi.org/10.1109/TWC.2020.3003744.

Makhija, D., Han, X., Ho, N., Ghosh, J.: Architecture agnostic federated learning for neural networks. In: Proceedings of the 39th international conference on machine learning (ICML) (2022)

Matsuda, K., Sasaki, Y., Xiao, C., Onizuka, M.: Fedme: Federated learning via model exchange. In: Proceedings of the 2022 SIAM international conference on data mining (SDM), pp. 459–467 (2022). https://epubs.siam.org/doi/abs/10.1137/1.9781611977172.52

McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics (AISTATS) (2017). arXiv:1602.05629

Mhaisen, N., Abdellatif, A. A., Mohamed, A., Erbad, A., & Guizani, M. (2022). Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints. *IEEE Transactions on Network Science and Engineering, 9*(1), 55–66. https://doi.org/10.1109/TNSE.2021.3053588.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning. Adaptive computation and machine learning* (2nd ed.). MIT Press.

Munir, M.T., Saeed, M.M., Ali, M., Qazi, Z.A., Qazi, I.A. (2022). Fedprune: Towards inclusive federated learning.

Pan, W., & Sun, L. (2021) Global knowledge distillation in federated learning. arXiv:2107.00051

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., Arora, R. (2020). FetchSGD: Communication-efficient federated learning with sketching. In: Proceedings of the 37th International conference on machine learning, vol.119, pp. 8253–8265. https://proceedings.mlr.press/v119/rothchild20a.html

Seo, H., Park, J., Oh, S., Bennis, M., Kim, S. (2020). Federated knowledge distillation. CoRR arxiv:2011.02367

Shen, T., Zhang, J., Jia, X., Zhang, F., Huang, G., Zhou, P., Wu, F., Wu, C. (2020). Federated mutual learning. CoRR arXiv:2006.16765

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H.B., et al: A field guide to federated optimization. CoRR arXiv:2107.06917 (2021)

Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., & Chen, M. (2019). In-edge AI:Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network, 33*(5), 156–165. https://doi.org/10.1109/MNET.2019.1800286.

Wu, J., Liu, Q., Huang, Z., Ning, Y., Wang, H., Chen, E., Yi, J., Zhou, B.: Hierarchical personalized federated learning for user modeling. In: Proceedings of the web conference 2021. WWW '21, pp. 957–968.

Association for computing machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442381.3449926

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*. https://doi.org/10.1145/3298981.

Yu, S., Qian, W., Jannesari, A. (2022). Resource-aware federated learning using knowledge extraction and multi-model fusion. 2208–07978 arXiv:2208.07978

Zhang, L., Luo, Y., Bai, Y., Du, B., Duan, L.-Y.: Federated learning for non-iid data via unified feature learning and optimization objective alignment. In: 2021 IEEE/CVF International conference on computer vision (ICCV), pp. 4400–4408 (2021). https://doi.org/10.1109/ICCV48922.2021.00438

Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H. (2018). Deep mutual learning. In: Proceedings of the IEEE conference on computer vision and pattern Recognition (CVPR)

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

Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-iid data: A survey. *Neurocomputing, 465,* 371–390.