Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5154

Empowering Client Selection with Local Knowledge Distillation for Efficient Federated Learning in Non-IID Data

Aissa Hadj Mohamed [**Universidade Estadual de Campinas** | ***a265189@dac.unicamp.br***]
Joahannes B. D. da Costa [**Federal University of São Paulo** | ***joahannes.costa@unifesp.br***]
Allan M. de Souza [**Universidade Estadual de Campinas** | ***allanms@unicamp.br***]
Leandro A. Villas [**Universidade Estadual de Campinas** | ***lvillas@unicamp.br***]
Julio C. Dos Reis [**Universidade Estadual de Campinas** | ***jreis@ic.unicamp.br***]

Instituto de Computação, Universidade Estadual de Campinas (Unicamp), Av. Albert Einstein, 1251, Cidade Universitária, Campinas, SP, 13083-852, Brazil.

Received: 03 November 2024 • **Accepted:** 12 May 2025 • **Published:** 02 October 2025

Abstract Federated Learning (FL) is a distributed approach in which multiple devices collaborate to train a shared, global model (GM). During its training, client devices must frequently communicate their gradients to the central server to update the GM weights. This incurs significant communication costs (bandwidth utilization and the number of messages exchanged). The heterogeneous nature of clients' local datasets poses an extra challenge to the model training. In this sense, we introduce FedSeleKDistill, Federated Selection and Knowledge Distillation Algorithm, to decrease the overall communication costs. FedSeleKDistill is an innovative combination of: (i) client selection, and (ii) knowledge distillation approaches with three main objectives: (i) reducing the number of devices training at every round; (ii) decreasing the number of rounds until convergence; and (iii) mitigating the effect of client's heterogeneous data on the GM effectiveness. In this paper, we extend the results obtained from the initial paper presenting FedSeleKDistill. The additional experimental evaluations on the MNIST and German Traffic Signs Benchmark datasets demonstrate that FedSeleKDistill is highly efficient in training the GM until convergence in heterogeneous FL. FedSeleKDistill reaches a higher accuracy score and faster convergence than state-of-the-art models. Our results also show higher performance when analyzing the accuracy scores on the clients' local datasets.

Keywords: Machine Learning, Federated Learning, Distributed Computing

1 Introduction

In recent years, there has been a significant development of computing devices, such as smartphones and Internet of Things devices de Souza *et al.* (2023). These devices generate vast datasets, which in turn have been used by researchers to achieve significant advances in training bigger, and more complex Deep Learning (DL) models, covering areas such as object/pattern recognition, natural language processing, and autonomous agents, like robots and self-driving cars. However, the use of vast amount of data for the DL field raises questions about ethics and privacy. Federated Learning (FL) has emerged as a solution to address these issues Li *et al.* (2020).

In essence, FL involves training DL models using data distributed across individual devices which we refer as clients. These clients devices train a DL model (called shared model or global model) using their local datasets while keeping them private. The FL training process begins with clients downloading the global model weights from the central server. Next, they train this model using their local data and transmit their local gradients to the central server. The central server updates the global model parameters through federated averaging (FedAvg McMahan *et al.* (2017)) and returns the newly updated model parameters to the FL clients. Then, a new round of training begins. This iterative process continues until the global model reaches a reasonable loss value. In this case, we say that the model has reached convergence, and a local minimum of its loss function.

In the context of FL, various challenges arise related to the communication between client devices and the central server (de Souza *et al.* (2024); Shahid *et al.* (2021)). Network-connected devices need to constantly share their updates, which can result in a communication bottleneck Mothukuri *et al.* (2021). Additionally, participating devices do not always have an adequate or reliable communication connection Lim *et al.* (2020). Due to limited bandwidth and energy resources of client devices, communication rounds can be time-consuming Li *et al.* (2020). Another challenge is the fact that the data of clients participating in the training process may not follow an independent and identically distributed (IID) distribution Li *et al.* (2018). This data reflects specific information of each client, such as usage patterns, user preferences, and local environment information Shahid *et al.* (2021). Therefore, the dataset of one client may not be representative of the data distribution of the entire population McMahan *et al.* (2017).

Therefore, optimizing communication efficiency in an FL environment becomes crucial Mothukuri *et al.* (2021). An efficient communication-aware distributed training algorithm for FL needs to meet the following requirements Sattler *et al.* (2020): (*i*) it should reduce communications between clients and the server, (*ii*) it should be robust to non-IID data, small batch sizes, and imbalanced data, and (*iii*) it should be robust to large numbers of devices and partial client participation.

In this sense, we present a new training algorithm called FedSeleKDistill which successfully addresses the communication challenges in FL. FedSeleKDistill is efficient in terms of the number of rounds required to reach convergence and effective in each round to select clients for training. Our empirical experiments on the MNIST and GTSB datasets show the superiority of FedSeleKDistill compared to Power-Of-Choice (POC) Cho *et al.* (2020) and FedAvg McMahan *et al.* (2017).

This article extends our previous work from Mohamed *et al.* (2024) by explaining the motivation behind the chosen approach to design FedSeleKDistill. Also, the article extends the number of FL clients in the MNIST experiments from 30 to 100. We conduct additional experiments by considering the German Traffic Sign recognition Benchmark (GTSB) Stallkamp *et al.* (2012). Therefore, the contributions of this work can be summarized as follows:

- We conduct an experiment to explain the Global Model (GM) forgetting problem in FL;
- We extend the initial experiments in Mohamed *et al.* (2024) for MNIST by considering a larger number of FL clients, and various levels of epochs per client per round;
- We consider the GTSB dataset for experiment for further performance of FedSeleKDistill against Power-Of-Choice (POC) and FedAvg.

This paper is organized as follows. The next section, Section 2, provides a presentation of the existing literature in FL for addressing the various communication-related challenges. Section 3 presents our proposed training algorithm, FedSeleKDistill. Section 4 evaluates the effectiveness of the proposed solution, while Sections Section 5 Section 6 discuss the obtained results and conclude this paper with final considerations.

2 Related Work

The research works in the past decade have adopted various approaches to address the communication challenges in FL. These approaches can be categorized into different groups, as proposed by Shahid *et al.* (2021). In this section, we present some recent studies in FL, organized according to these categories.

The first approach aims to decrease the number of communications between the clients and the central server. FedAvg McMahan *et al.* (2017) reduces the number of messages exchanged by increasing the number of local iterations per client per round before sending the clients' local gradients to the central server to update the Global Model (GM). However, to guarantee convergence, FedAvg assumes that the data distribution among clients is IID.

Another approach considers compressing the messages exchanged between the central server and the clients. Quantization techniques (Amiri *et al.* (2020); Bernstein *et al.* (2018); Lin *et al.* (2020b)) and sparsification (Rothchild *et al.* (2020); Sattler *et al.* (2020); Ström (2015)) are examples of such methods. However, not all of these compression methods guarantee the convergence of the global model Shahid *et al.* (2021) in heterogeneous data. Client selection for training at each round is another approach. In this context, Power-Of-Choice (POC) Cho *et al.* (2020) is a communication and computation efficient selection framework that tends to select clients with higher local loss. Experimental evaluations have shown that POC achieves faster convergence, increases communication efficiency, and reduces overall communication costs.

Similarly, FOLB Nguyen *et al.* (2020) performs intelligent sampling of clients in each training round. As observed by the authors, specific clients provide more significant improvements to the global model than others during training. In Wang *et al.* (2020), the authors use reinforcement learning (RL) to train an agent called FAVOR to select an optimal set of clients in each training round. FAVOR intelligently chooses client devices to balance the bias introduced by non-IID data and accelerate model convergence. However, FAVOR requires retraining if exposed to new environmental conditions. Additionally, we cannot fully understand FAVOR's criteria for selecting the ideal clients in each round, as it is an uninterpretable black-box.

In Mohamed *et al.* (2023), the authors propose the CCSF framework that clusters clients into homogeneous groups. Then, using one of three selection strategies, CCSF creates a subset of clients for training in each round. Experimental results showed that CCSF trains a global model to convergence more quickly than FedAvg. However, a limitation of this solution is that CCSF requires all clients to be clustered into homogeneous groups.

Knowledge Distillation (KD) is an increasingly adopted approach in recent works Mora *et al.* (2022). According to the authors, KD can be employed for two purposes. The first is to allow participating clients to select different model architectures (model heterogeneity). The second is to mitigate the impact of data heterogeneity on the performance of the global model. As the focus of this work is to address the non-IID data problem, we present some recent works with the second objective.

Mora *et al.* (2022) distinguishes between server-side strategies that refine the FedAvg aggregation with a distillation phase and client-side techniques that locally distill global knowledge to handle client drift. As a server-side KD-based solution, FedDF Lin *et al.* (2020a) uses a proxy dataset to fine-tune the global model by imitating the output of the ensemble model of the clients. A limitation is that FedDF assumes that the central server and clients share a common dataset (proxy dataset).

In the absence of common data, Zhang *et al.* (2022) propose FedFTG which models the input space of the local models using an auxiliary generator at the central server, and then generates common data to transfer the knowledge in the local models to the global model, in order to improve performance.

On the other hand, FedGKD Yao *et al.* (2021) fuses the knowledge of historical global models for local training to mitigate the "client drift" problem. As clients perform local updates on heterogeneous data, their local models diverge. As a result, the global model may overfit to the local data of some clients. FedGKD uses client-side KD to guide the training of the local model by global teachers (past global models), where each client learns the global knowledge of past global models via adaptive knowledge distillation techniques.

The authors of Lee *et al.* (2022) observe that, when trained on the client's local data, the global model forgets the knowledge from previous rounds. Based on these findings, the authors propose Federated Not-True Distillation (FedNTD) to address the forgetting problem. FedNTD adopts local distillation of global knowledge to mitigate forgetting between subsequent rounds and alleviate the harmfulness of data heterogeneity. However, as stated by the authors, if the global model is biased, the trained local model is more prone to have a similar tendency.

According to He *et al.* (2022), when the global model has not fully converged and has not fully learned the distribution of the clients' local data, the performance of the global model may be better or worse for some classes. He *et al.* (2022) propose a Class-Adaptive Auto-Distillation (FedCAD) mechanism to address this problem. Under this solution, FedCAD uses class-adaptive terms to smooth the influence of the distillation loss according to the performance of the global model on each class.

Table 1 summarizes the solutions from the literature, categorized according to their approach Shahid *et al.* (2021), their advantages and limitations.

3 Proposal

This section introduces the FedSeleKDistill algorithm, which addresses three critical problems in FL: *(i)* non-IID data; *(ii)* high communication overhead; and *(iii)* limited client participation rate.

The first problem is addressed through knowledge distillation. The latter two problems are solved by a biased client selection strategy to increase the convergence speed of the model.

3.1 Motivation

To illustrate our motivation in combining client selection with knowledge distillation, we conduct a pre-experiment similar to He *et al.* (2022). We consider a FL environment with 100 clients sharing the MNIST dataset (see Section 4.1 for details). First, we train the global model (GM) for one training round on 5 (five) clients' data using FedAvg. We evaluate its performance on each class of the test set. Then, we train it again for one round on one client data using FedAvg. We evaluate it again on each class of the test set. Figure 1 presents the results of the pre-experiment.

Looking at the results of the second row, before training GM for a second time, its performance was high for classes [0, 1, 2, 3, 4, 5], above 82%. And null on classes [6, 7, 8, 9]. After training on one client, GM improved its accuracy score on the test set. Its performance increased for class 9 from 0% to 100%. Same for classes [6, 7, 8]. For class 0, GM maintained its accuracy score at 99%. In the meantime, GM performance deteriorated for all the other classes. The most notable drop is for class 4 where the accuracy score decreased from 97% to 44%. As observed by Lee *et al.* (2022), GM forgot the knowledge gained on classes [1, 2, 3, 4, 5] when trained on the client's data. Besides, a client selection-based algorithm using federated averaging similar to FedAvg will obtain the same results, and the same forgetting issue as for FedAvg.

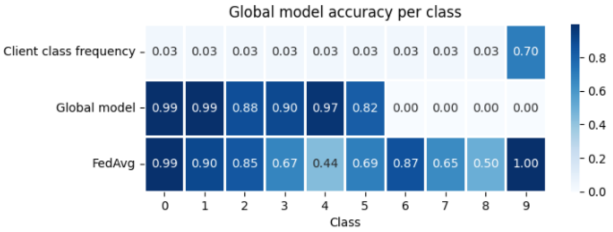

Figure 1. Results of the pre-experiment on the MNIST dataset. The second row shows the accuracy score of the global model on the test set on each class after one training on 5 (five) clients. The second row, after 1 (one) training on one client using the FedAvg algorithm. The top row corresponds to the client's class distribution relative to its sample size.

Hence, our present study proposes a local knowledge distillation combined with a client selection strategy to effectively train the shared model in Federated Learning.

3.2 Problem Statement

Consider a cross-device FL setup with a total of N clients, where client $n \in N$ has a local dataset D_n consisting of $|D_n|$ data samples. We aim the clients to train locally a global model with initial weights w_{init}, share their local weights w_n with the central server, which is responsible for aggregating the weights of each client, and collectively find the model parameter w^* that minimizes:

$$F(w) = \frac{1}{\sum_{n \in N} |D_n|} \sum_{n \in N} \sum_{d_n \in D_n} f(w_n, d_n) \tag{1}$$

Which can be written as:

$$F(w) = \sum_{n \in N} p_n F_n(w_n) \tag{2}$$

where $f(w_n, d_n)$ is the composite loss function for sample $d_n \in D_n$ and the weights w_n of client n. The term p_n is defined as:

$$p_n = \frac{|D_n|}{\sum_{n \in N} |D_n|} \tag{3}$$

which is the data fraction of client n, and

$$F_n(w_n) = \frac{1}{|D_n|} \sum_{d \in D_n} f(w, d) \tag{4}$$

is the local objective function of client n.

3.3 Description of FedSeleKDistill

As stated in Mohamed *et al.* (2024), the goals of FedSeleKDistill are to (*i*) train a global model to minimize the loss function (1) to a local minimum, considering the constraints in the FL environment (e.g., non-IID data, limited client participation, and available computational resources); (*ii*) achieve convergence with the fewest possible communication rounds; and (*iii*) have efficient client-server communication. FedSeleKDistill is based on a biased client selection

Table 1. Comparison of Federated Learning approaches

Training algorithms	Approach [a]	Objective	Advantages	Limitations
FedAvg McMahan *et al.* (2017)	reduced model updates	reduce client-to-server communication exchanges	simple to implement	assumes IID data among clients, or full client participation in non-IID data
DGC Lin *et al.* (2020b)	sparsification, faster convergence	solution to communication bandwidth, client training scalability, high latency, lower throughput, poor connections	highly reduce gradient communication volume	knowing gradient threshold for top-k selection, convergence not guaranteed for non-IID data
STC Sattler *et al.* (2020)	sparsification + ternarization + Golomb encoding of weight updates	compression of upstream and downstream communications	works on non-iid data, not all clients need to participate	convergence not guaranteed for non-IID data, requires caching resources in client devices
FetchSGD Rothchild *et al.* (2020)	sparsification	compression for upstream communication	works on non-iid data, not all clients need to participate	no downstream communication compression
SIGNSGD Bernstein *et al.* (2018)	quantization	transmitting just the sign of gradients	compressed gradients and SGD-level convergence rate	convergence not guaranteed for non-IID data
LFL Amiri *et al.* (2020)	quantization	quantizing models before broadcasting	significant communication cost saving, works on non-IID data	LFL compared to lossless scenario only
Power-Of-Choice Cho *et al.* (2020)	client selection	biased clients selection with higher local loss	faster convergence, assumes low clients participation, minimal communication and computation overhead, works on non-IID data	knowing how to set the trade-off between convergence speed and solution bias
FOLB Nguyen *et al.* (2020)	client selection, reduced model updates	reduce number of communication rounds, low client participation	works on non-IID data, takes into account statistical and system heterogeneity in clients	if the client participation rate is low, few clients to estimate true global gradient
FAVOR Wang *et al.* (2020)	client selection	faster convergence	works on non-IID data	Agent needs training for new FL settings, no deep understanding of the client selection criteria
CCSF Mohamed *et al.* (2023)	client selection, sparsification	reduce number of communication rounds, low client participation	works on non-IID data	works only if clients can be grouped into homogeneous clusters
FedDF Lin *et al.* (2020a)	knowledge distillation	tackles non-IID data, server-side KD	works on non-IID data	requires a proxy dataset
FedFTG Zhang *et al.* (2022)	knowledge distillation	tackles non-IID data	does not require a proxy dataset, works on non-IID data	central server must generate synthetic dataset as a proxy dataset
FedGKD Yao *et al.* (2021)	knowledge distillation	tackles non-IID data, client-side KD	does not require a proxy dataset, works on non-IID data	may require considerable computing ressources in case of very large, complex shared model
FedNTD Lee *et al.* (2022)	knowledge distillation	tackles non-IID data, client-side KD	addresses the forgetting problem, works on non-IID data	if the global model is biased, the trained local model is more prone to have a similar tendency Lee *et al.* (2022)
FedCAD He *et al.* (2022)	knowledge distillation	tackles non-IID data, client-side KD	works on non-IID data	requires a proxy dataset, central server must know the clients number of samples per-class
FedSeleKDistill (our solution) Mohamed *et al.* (2024)	knowledge distillation, client selection	tackles non-IID data, limited client participation	works on non-IID data, faster convergence, does not require a proxy dataset	mostly effective in case of highly heterogeneous data

[a] We describe the approach of each training method according to the classification in Shahid *et al.* (2021).

strategy that makes training efficient in terms of the number of rounds required to train the global model to convergence. To mitigate the negative effect of data heterogeneity among FL clients, FedSeleKDistill utilizes local knowledge distillation (KD).

3.4 Knowledge Distillation

Knowledge distillation is a transfer learning technique where a more complex and robust pre-trained model, known as the teacher, is used to train a smaller and simpler model, called the student (Bucila *et al.* (2006); Hinton *et al.* (2015)). The primary goal of knowledge distillation is to transfer the knowledge from the teacher to the student, allowing the student to acquire a more compact and efficient representation of the patterns learned by the teacher. This is particularly useful when the teacher is a large, computationally expensive model, and the student needs to be deployed on devices with limited resources, such as mobile devices or embedded systems.

In the context of federated learning (FL), local KD-based regularization aims to effectively reduce the influence of non-IID data Mora *et al.* (2022). The local objective function (1) on the client device becomes a linear combination of the cross-entropy loss and a KD-based loss, which evaluates the difference between the output of the global model (teacher model) and the output of the local model (student model) on the local data, using the Kullback-Leibler divergence, as described below:

$$L_{local} = (1-\lambda) \cdot L_{CE} + \lambda \cdot L_{KD} \tag{5}$$

where λ is the weight of the Kullback-Leibler divergence, with a value between 0 and 1 that determines the contribution of the KL loss term to the local loss of client k, L_{local}, L_{CE} is the cross-entropy loss and L_{KD} is the KD-based loss.

Using the notations from Mora *et al.* (2022), the function 5 for client k becomes:

$$\begin{split} f(w_k, d_k) = (1-\lambda) \left(-\frac{1}{|D_k|} \sum_{(x_i, y_i) \in D_k} y_i \cdot \log(\hat{y}i, wk, t+1) \right) \\ + \lambda \mathrm{KL} \left(P_{w_t}(y_i), ||, P_{w_{k,t+1}}(y_i) \right) \end{split} \tag{6}$$

where D_k is the local dataset of client k, (x_i, y_i) is the data sample i of D_k. w_t represents the weights of the global model (teacher model) at round t. $w_{k,t+1}$ represents the local model (student model) trained on the local dataset D_k at round t+1.

After being selected for training, the client device creates a copy of the global model. The original model acts as the teacher and transfers its knowledge to the copy (the student) through distillation using the client's local data. The student model is the local model trained with the local objective function 6.

Additionally, this knowledge distillation process preserves the privacy of the clients' data, since no information about the local data distribution, except the data size, is shared with other FL clients or the central server.

3.5 Client Selection

The second key idea is inspired by the Power-Of-Choice (POC) method Cho *et al.* (2020) for selecting clients for training. According to the authors, biasing client selection towards those with the highest local loss $f_n(w)$ can improve the convergence of the global model. For example, consider a scenario with two clients, one with high local loss and the other with significantly lower local loss. In a given training round, the global model learns more significantly by focusing on the local data of the client with the worst loss.

Mathematically, the client with the worst local loss has a gradient of greater magnitude than the client with the lower local loss. The term $\frac{\partial L}{\partial w}$ represents the gradient of the loss function with respect to the weights of the global model.

The magnitude of the gradient, $\left\| \frac{\partial L}{\partial w} \right\|$, refers to the size of the gradient. This term increases when the value of L, representing the local loss, is larger. Therefore, if we update the weights of the global model by selecting the gradient with the largest magnitude, the global model will make greater progress in the next update step towards the local minimum of the loss function 1. However, the more biased the client selection towards clients with the highest loss to achieve faster convergence, the greater the risk of a non-vanishing gap between the true optimal weights and the weights at convergence for the global model with this strategy Cho *et al.* (2020). This is the trade-off between convergence speed and solution bias mentioned by the authors Cho *et al.* (2020).

3.6 Algorithm Description

FedSeleKDistill combines the training algorithm proposed in the POC framework Cho *et al.* (2020) with local knowledge distillation. Therefore, FedSeleKDistill adopts two strategies. The first is a biased client selection strategy to train the global model until convergence with fewer rounds compared to FedAvg. The second approach mitigates the negative effect of data heterogeneity on client devices. Especially when the number of iterations (epochs) per client per round is high, the risk of the global model overfitting to the local data increases. Additionally, local KD has the added advantage of preventing the global model from forgetting knowledge acquired in previous rounds from past client data when trained with local data from a new client.

Based on the works proposed in Cho *et al.* (2020), only the local training algorithm needs to be modified. Algorithm 1 is originally proposed from Mohamed *et al.* (2024) and presented in the current paper for ease of understanding to the reader. The value of λ is selected from a range of 0 to 1. Higher values gives more weight to the KD component compared to the cross-entropy loss term. And vice versa when the λ value is low. Generally, the value of λ is set to 0.5.

4 Performance evaluation

This section describes the experimental methodology and metrics used to evaluate the effectiveness of FedSeleKDistill.

Algorithm 1: Client Device

```
// When client $n$ receives request to train
LocalTrain $(w)$:
    for $época \in \{1, 2, \ldots, E\}$ do
        foreach $batch \in D_n$ do
            // update the local model
            $w_n \leftarrow (1-\lambda)L_{CE}(w_t, batch) + \lambda L_{KD}(w_t, batch)$ ;
    // send the local gradients to central server
    return $w_n$;
```

4.1 Datasets Description

We selected two datasets for the experiments and created the clients' local data following the methodology proposed by Wang *et al*. (2020).

Figure 2. Some of the images belonging to the German Traffic Sign Benchmark (GTSB).

The German Traffic Sign Benchmark (GTSB) Stallkamp *et al*. (2012) is a dataset used for recognizing traffic signs (43 classes). Figure 2 displays some examples of images from the GTSB dataset. We split the data between 100 clients.

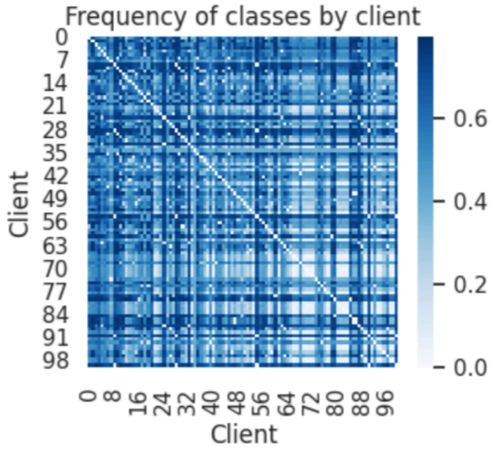

Figure 3. The Kolmogorov-Smirnov tests are performed on each pair of clients present in the dataset (GTSB). The higher the value, from 0 to 1, the greater the level of heterogeneity between the datasets of two clients.

The first 90 clients train the shared model, and the last 10 constitute the test set. In Figure 3, we conduct Kolmogorov-Smirnov tests between every pair of clients' local data. 95% of all pairs of training clients have significantly different distributions.

The shared model for this classification task is a convolutional neural network consisting of 3 sets of convolutional layers, batch normalization and max pooling, followed by a fully connected layer, a dropout layer and a dense layer with a softmax activation function.

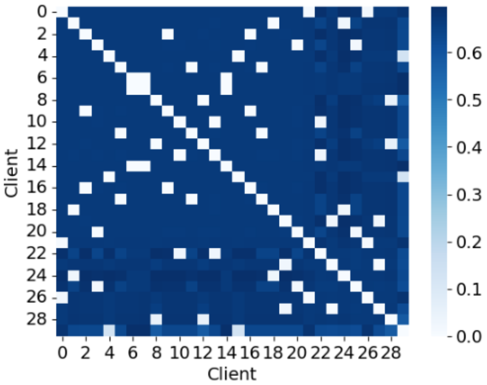

Figure 4. The Kolmogorov-Smirnov tests are performed on each pair of clients present in the dataset (MNIST). The higher the value, from 0 to 1, the greater the level of heterogeneity between the datasets of two clients.

The second dataset MNIST contains a total of 70,000 grayscale images, divided into a training set of 60,000 images and a test set of 10,000 images. Each 28x28 pixel image depicts a handwritten digit (0 to 9). Details about the data can be found in LeCun *et al*. (1998).

The task is a classification problem, where we train a deep learning model to classify the images into 10 classes (0 to 9). For the experiment, we set the total number of FL clients to 30. We divided the 60,000 images equally among all clients. Each local dataset contains 70% of one class, and the remaining 30% is a uniform distribution of the other classes. The local datasets created to be distributed to all FL clients followed the methodology used in Wang *et al*. (2020).

The global model selected to learn the MNIST dataset is a sequential deep neural network composed of 2 convolutional layers and pooling, followed by a flattening layer, dropout regularization, and a dense layer with softmax activation for class prediction.

4.2 Assessed Metric

The total communication costs in training an ML model to convergence is measured by: (1) the number of clients training per round; (2) the size and number of messages exchanged between the clients and the central server; and (3) the number of rounds required to reach convergence. In our experimental evaluations, we focus on the number of rounds required to train an ML model until convergence. The lower the number of rounds, the lower the communication costs and the more effective is FedSeleKDistill.

We evaluate FedSeleKDistill, POC and FedAvg using the accuracy score on the test set over communication rounds as the assessed metric. The fewer rounds spent to achieve the maximum accuracy, the faster the solution's convergence is and the more effective the training algorithm.

In a centralized environment, the global model reaches an accuracy score of 96% on the test set after 10 epochs for the German Traffic Sign Benchmark, and 98.94% on the test set after 5 training epochs for MNIST.

Therefore, the objective of the two classification tasks in FL is to reach an accuracy score on the test set as close as possible to the accuracy scores in the centralized settings.

4.3 Experimental Settings

Table 2 presents in details the parameters for the two experiments. We consider 2 state-of-the-art training algorithms as baselines: (i) Power-Of-Choice (POC), (ii) and FedAvg.

In the first experiment (GTSB), we assessed FedSeleKDistill on the test set. After 35 rounds of training, we compared the effectiveness of the model trained by the 3 algorithms on the clients' local datasets. In the second experiment (MNIST), we increased the number of epochs per client per round and assessed the performance of FedSeleKDistill against the 2 baselines.

The experimental evaluations were conducted on Google Colab using the TensorFlow framework. In our simulation settings, we assume that all the clients have the same computing resources. They only differ in their local data distributions. We use the TensorFlow function `tf.keras.losses.KLD` to compute the Kullback-Leibler divergence loss between the outputs of the global and the client models. The KD loss function on the client-side is a linear combination of the categorical cross-entropy loss (multi-class classifications) and the Kullback-Leibler divergence loss weighted by λ.

4.4 Results

In the first experiment (GTSB), the 3 training algorithms successfully train the shared model (see Figure 5, left plot).

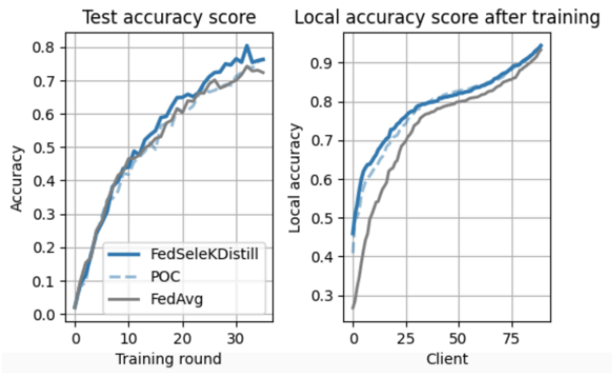

Figure 5. Accuracy scores on the test set and local data after 35 training rounds on the GTSB experiment.

Although the client datasets appear to be mostly non-IID (see Figure 3), after 35 communication rounds, the shared model attains an accuracy score above 70% on the test set with FedAvg. The relatively high heterogeneity level in clients' data did not prevent FedAvg from training the shared model until convergence. FedAvg performance is on par with POC. From round 10, we notice a positive gap in performance between FedSeleKDistill and the 2 baselines. The shared model attains a higher, faster accuracy score with FedSeleKDistill, with a maximum accuracy score slightly above 80% versus less than 75% for POC and FedAvg, after 35 training rounds.

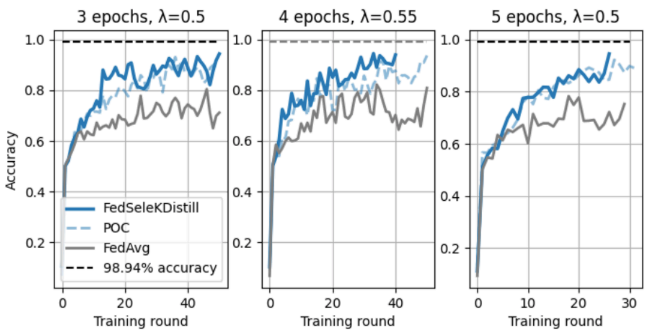

Figure 6. Accuracy scores on the test set on the MNIST experiment, with 3, 4 and 5 epochs per client per round.

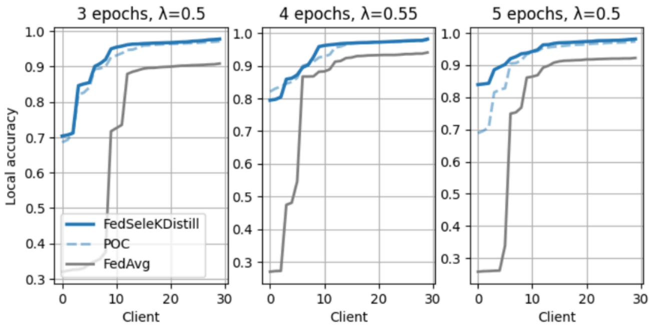

Figure 7. Shared model effectiveness on clients' local datasets after training with FedSeleKDistill, POC and FedAvg in MNIST experiment.

In Figure 6, we present the results obtained on the MNIST dataset with 30 FL clients. Those results are taken from Mohamed *et al*. (2024). We notice that FedAvg struggles to train the shared model compared to POC and FedSeleKDistill. For each experiment with 3, 4 and 5 epochs, the accuracy score of the shared model trained with FedAVG stays below 80% in the first 50 FL rounds. With POC and FedSeleKDistill, the shared model attains an accuracy score above 80% in the first 20 rounds with 3 and 4 epochs, and after 15 rounds with 5 epochs. Those results can be explained by the relatively high heterogeneity in clients' data. In this experiment, the performance gap between FedSeleKDistill and POC is much wider than in the case of GTSB. FedSeleKDistill attains a higher accuracy score in fewer rounds than POC. Once we train the shared model for more than 4o rounds by each algorithm, we evaluate its performance on the clients' local data. Figure 7 shows the results. Since we have 30 clients, we have 30 accuracy scores per training algorithm (FedSeleKDistill, POC and FedAvg). For easier analysis, we ordered the local accuracy scores from low to high. The higher the local accuracy scores, the more effective is the training algorithm in learning the local distributions. Thanks to its KD component, FedSeleKDistill improved the learning process of the shared model on the local datasets (see Figure 7).

Table 3 presents the highest accuracy scores attained and the corresponding training round by each solution on GTSB (with 2 epochs/client) and MNIST (with 3 to 5 epochs/client) with 30 clients. In each experiment, FedSeleKDistill allows the shared model to attain the highest accuracy scores in fewer rounds than the 2 baseline models.

Overall, when we increase the heterogeneity level of clients' data (from GTSB to MNIST), we also increase the performance gap between FedSeleKDistill and POC and FedAvg. FedSeleKDistill allows the shared model to learn more effectively the local data distributions compared to POC and FedAvg thanks to its knowledge distillation component.

Table 2. Details about the experiments

Feature	Dataset	
	German Traffic Sign	MNIST
Data samples per client	392	2000
Total number of clients	90	30
Participation rate (clients/round)	8.88%	33%
Selection rate (clients training/round)	4.44%	16.5%
Number of epochs/client/round	2	3 to 5

Table 3. Detailed results for GTSB and MNIST (M - number of epochs) experiments

Strategy	GTSB	M-3	M-4	M-5
FedAvg	74.25% (32)	80.42% (46)	82.20% (34)	78.25% (18)
POC	76.60% (35)	93.04% (36)	93.67% (33)	92.17% (27)
FedSeleKDistill	80.45% (32)	94.31% (50)	94.44% (33)	94.45% (26)

4.5 Limitations

In this section, we present the limitations of FedSeleKDistill when we select a non-optimal value for λ, or in the case of FL environments with a relatively low level of heterogeneity in clients datasets.

4.5.1 Value of λ

In this experiment, we consider the MNIST dataset and 100 clients, with $\alpha = 70\%$. We distributed the $60,000$ image samples between the clients following the methodology proposed by Wang *et al*. (2020).

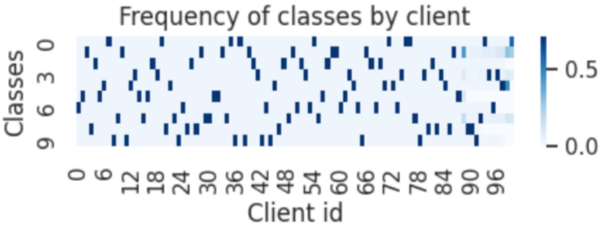

Figure 8. Local per-class data distributions between 100 clients on the MNIST dataset with $\alpha = 70\%$.

Figure 8 displays the frequency of the 10 classes among the FL clients. As we can see, the clients' data distributions are highly heterogeneous.

We train the same DL model that we selected for the MNIST experiments with 30 clients. We conduct the experiments with 2, 3, 4 and 5 epochs per round per client. For the KD term in the local loss function, we set the value of λ equal to 0.5 for FedSeleKDistill.

Figure 9 shows the performance of the shared model on the test set when trained with FedSeleKDistill, POC and FedAvg. The lowest performance obtained is with FedAvg. The gap in performance with the two other training algorithms underlines the level of client's data heterogeneity present in our FL experiment. Looking at the four plots, FedSeleKDistill trains the shared model more effectively than POC overall. However, if we want better optimal results, especially in the case of three epochs, one should change the value for λ similar to what we did in the experiments of MNIST with 30 clients.

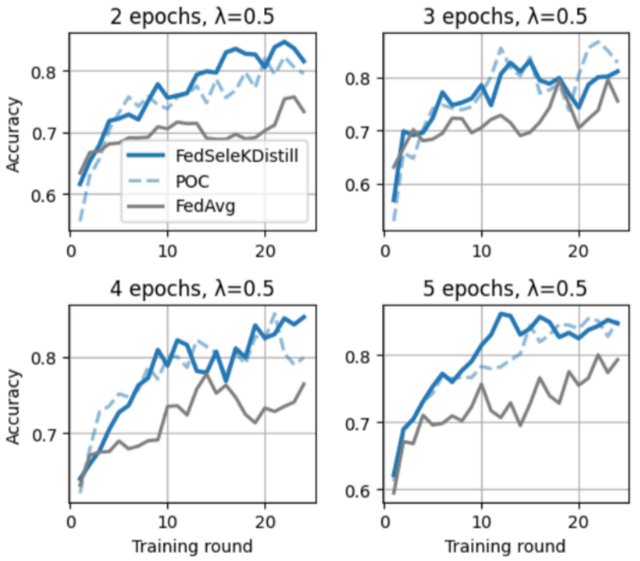

Figure 9. Accuracy scores on the test set on the MNIST experiment, with 2, 3, 4 and 5 epochs per client per round with 100 clients.

4.5.2 Decreasing the heterogeneity level in clients' data

In Mohamed *et al.* (2024), the authors conducted an experiment by considering the Human Activity Recognition (HAR) dataset[1]. It is a dataset collected from 30 study participants performing daily activities while carrying a smartphone mounted on their waist with movement sensors. According to Mohamed *et al.* (2024), the knowledge distillation component of FedSeleKDistill slows down the progress of the global model towards the local minimum of the loss function.

The explanation is the following. The global model's weights are randomly initialized at round 0. In the initial rounds, by decreasing the risk of forgetting the shared model knowledge, KD prevents the model from making significant

[1] `https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones`

progress towards the local minimum of the loss function compared to POC. Therefore, FedSeleKDistill lags behind POC in terms of accuracy score in the early rounds. As a result, POC trains the global model much faster to convergence. This suggests that, for local datasets with a relatively low level of heterogeneity, it is preferable to select POC over FedSeleKDistill to achieve faster convergence. Another possibility to fix the convergence delay is to variate the value of λ depending on the shared model knowledge as measured by its accuracy score on the test set. If the test accuracy is low, one could set a relatively low value of λ. And vice versa, when the test accuracy score is high.

5 Discussion

In this paper, we present FedSeleKDistill and conduct additional experimental evaluations from the initial results found in Mohamed *et al*. (2024). FedSeleKDistill is a novel training algorithm within the Federated Learning (FL) paradigm. It addresses three significant challenges in FL: the communication bottleneck, the heterogeneity of local data present on FL clients' devices, and the limited number of clients available for training in each round. FedSeleKDistill combines a client selection approach with local knowledge distillation (KD).

The experimental evaluations conducted on the MNIST and GTSB datasets demonstrate that FedSeleKDistill achieves greater performance than baseline models. By combining a biased client selection strategy with knowledge distillation, FedSeleKDistill trains the shared model to convergence in much fewer rounds compared to the same biased client strategy without KD (POC).

However, FedSeleKDistill has some limitations. For best performance, it is required to set an optimal value of weight λ for the KD loss term in the local loss function on the client-side. Otherwise, the shared model performance, measured by its accuracy score, may not be greater than for the case of POC. In addition, our experimental findings showed that FedSeleKDistill works more effectively when the clients' data distributions are highly heterogeneous. In this FL situation, the risk of the shared model to overfit the client's local data is greater. This client drift results in decreased performance of the shared model overall accuracy score. FedSeleKDistill fixes this overfit issue thanks to its KD component.

6 Conclusion

This paper presented a novel training algorithm, called FedSeleKDistill, within the Federated Learning (FL) paradigm. This FL algorithm addresses three significant challenges in FL: the communication bottleneck, the heterogeneity of local data present on FL clients' devices, and the limited number of clients available for training in each round. FedSeleKDistill combines a client selection approach with local knowledge distillation (KD). The first approach, based on Power-Of-Choice (POC), aims to train the global model with faster convergence. To reduce the risk of client drift, FedSeleKDistill adopts local KD on the client-side to mitigate the negative effect of non-IID data and prevent the global model from forgetting knowledge acquired in previous rounds. In this paper, we extended the results from the initial paper presenting FedSeleKDistill. Additional evaluations conducted on the MNIST and GTSB dataset demonstrate that combining a biased client selection strategy with knowledge distillation results in better performance in terms of accuracy scores compared to the same biased client strategy without KD (POC).

Furthermore, FedSeleKDistill requires fewer FL rounds to train the shared model to convergence compared to FedAvg. The novelty of the proposed solution FedSeleKDistill lies in the combination of a client selection strategy with KD to mitigate the negative effect of the clients' data heterogeneity. To the best of our knowledge, no recent work in the literature explores the combination of the two methods for efficient training in FL. Even though we have adopted a basic KD approach, the experimental results show improvements compared to POC. The experimental results also showed that for low levels of heterogeneity in clients' data, the KD component of FedSeleKDistill delays the convergence of the shared model. In those FL settings, POC reaches convergence faster. This suggests that we can improve the KD component of FedSeleKDistill. In future studies, we plan to refine our local KD strategy by varying the value of λ throughout the FL training rounds, or by setting a value of λ depending on the performance of the global model on the clients' local data.

Acknowledgements

This project was supported by the Brazilian Ministry of Science, Technology and Innovations, with resources from Law nº 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024-10.

Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare no conflicts of interest.

Availability of data and materials

The data and materials used in the article are available upon request.

References

Amiri, M. M., Gunduz, D., Kulkarni, S. R., and Poor, H. V. (2020). Federated learning with quantized global model updates. *arXiv preprint arXiv:2006.10672*. DOI: 10.48550/arxiv.2006.10672.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. (2018). signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR. DOI: 10.48550/arXiv.1802.04434.

Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 535–541. ACM. DOI: 10.1145/1150402.1150464.

Cho, Y. J., Wang, J., and Joshi, G. (2020). Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*. DOI: 10.48550/arxiv.2010.01243.

de Souza, A. M., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14. SBC. DOI: 10.5753/sbrc.2023.499.

de Souza, A. M., Maciel, F., da Costa, J. B., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2024). Adaptive client selection with personalization for communication efficient federated learning. *Ad Hoc Networks*, page 103462. DOI: 10.2139/ssrn.4654118.

He, Y., Chen, Y., Yang, X., Zhang, Y., and Zeng, B. (2022). Class-wise adaptive self distillation for federated learning on non-iid data (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12967–12968. DOI: 10.1609/aaai.v36i11.21620.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. DOI: 10.48550/arxiv.1503.02531.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. DOI: 10.1109/5.726791.

Lee, G., Jeong, M., Shin, Y., Bae, S., and Yun, S.-Y. (2022). Preservation of the global knowledge by not-true distillation in federated learning. *Advances in Neural Information Processing Systems*, 35:38461–38474. DOI: 10.48550/arxiv.2106.03097.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60. DOI: 10.1109/msp.2020.2975749.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018). Federated optimization in heterogeneous networks. *CoRR*, abs/1812.06127. Available at:`http://arxiv.org/abs/1812.06127`.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063. DOI: 10.1109/COMST.2020.2986024.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. (2020a). Ensemble distillation for robust model fusion in federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2351–2363. Curran Associates, Inc. Available at:`https://proceedings.neurips.cc/paper_files/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf`.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, W. J. (2020b). Deep gradient compression: Reducing the communication bandwidth for distributed training. DOI: 10.48550/arxiv.1712.01887.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR. DOI: 10.48550/arxiv.1602.05629.

Mohamed, A., Souza, A., Costa, J., Villas, L., and Reis, J. (2024). Fedselekdistill: Empoderando a escolha de clientes com a destilação do conhecimento para aprendizado federado em dados não-iid. In *Anais do VIII Workshop de Computação Urbana*, pages 71–84, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/courb.2024.3238.

Mohamed, A. H., de Souza, A. M., da Costa, J. B. D., Villas, L., and dos Reis, J. C. (2023). Ccsf: Clustered client selection framework for federated learning in non-iid data. In *Proceedings of the 16th IEEE/ACM Utility and Cloud Computing Conference (UCC)*, UCC '23, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3603166.3632563.

Mora, A., Tenison, I., Bellavista, P., and Rish, I. (2022). Knowledge distillation for federated learning: a practical guide. *arXiv preprint arXiv:2211.04742*. DOI: 10.48550/arxiv.2211.04742.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640. DOI: 10.1016/j.future.2020.10.007.

Nguyen, H. T., Sehwag, V., Hosseinalipour, S., Brinton, C. G., Chiang, M., and Poor, H. V. (2020). Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218. DOI: 10.1109/jsac.2020.3036952.

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., and Arora, R. (2020). Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265. PMLR. DOI: 10.48550/arxiv.2007.07682.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. (2020). Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413. DOI: 10.1109/TNNLS.2019.2944481.

Shahid, O., Pouriyeh, S., Parizi, R. M., Sheng, Q. Z., Srivastava, G., and Zhao, L. (2021). Communication efficiency in federated learning: Achievements and challenges. *arXiv preprint arXiv:2107.10996*. DOI: 10.48550/arxiv.2107.10996.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. volume 32, page 323–332, GBR. Elsevier Science Ltd.. DOI: 10.1016/j.neunet.2012.02.016.

Ström, N. (2015). Scalable distributed dnn training using commodity gpu cloud computing. In *Interspeech 2015*. DOI: 10.21437/interspeech.2015-354.

Wang, H., Kaplan, Z., Niu, D., and Li, B. (2020). Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1698–1707. IEEE. DOI: 10.1109/INFOCOM41043.2020.9155494.

Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Jin, H., Xu, Z., and Sun, L. (2021). Local-global knowledge distillation in heterogeneous federated learning with non-iid data. *arXiv preprint arXiv:2107.00051*. DOI: 10.48550/arxiv.2107.00051.

Zhang, L., Shen, L., Ding, L., Tao, D., and Duan, L.-Y. (2022). Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10164–10173. DOI: 10.1109/CVPR52688.2022.00993.