SEM: A Simple Yet Efficient Model-agnostic Local Training Mechanism to Tackle Data Sparsity and Scarcity in Federated Learning

Quang Ha Pham*, Nang Hung Nguyen*, Thanh Hung Nguyen*, Huy Hieu Pham†,
Phi Le Nguyen*§, Truong Thao Nguyen‡§
*School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
{ha.pq194546@sis, hung.nn184118@sis, hungnt@soict, lenp@soict}.hust.edu.vn
†College of Engineering & Computer Science and VinUni-Illinois Smart Health Center, VinUniversity; hieu.ph@vinuni.edu.vn
‡The National Institute of Advanced Industrial Science and Technology (AIST), Japan; nguyen.truong@aist.go.jp

***Abstract*—Recent years have witnessed the emergence of Federated Learning (FL) as a viable learning paradigm that permits the training of models without revealing sensitive data. FL systems typically consist of numerous clients that utilize their data to train models locally and an orchestration server responsible for combining local updates from the clients to produce a global model. Thus, the performance of a Federated learning system is highly dependent on the client data and the local model trained on these data. In this study, we present an early attempt at addressing the sparsity and scarcity of client data, which may lead to the overfitting phenomenon of local models and substantially reduce the overall accuracy of the global model. Specifically, we propose a novel local training strategy that explores transfer learning and allows each local model to be trained using the data of two randomly paired clients. The proposed method is orthogonal to other Federated Learning algorithms and can be integrated into most Federated Learning systems. Extensive experiments in various settings on MNIST, CIFAR-10, and CIFAR-100 datasets showed that using our proposed method can relatively enhance the accuracy of the global model by up to 12.48%. Our work, for the first time, offers a simple yet effective solution that reduces the undesired effects of data sparsity and scarcity in FL.**

***Index Terms*—Federated Learning, Data sparsity, Data scarcity, Transfer Learning, Model-agnostic Methods.**

I. Introduction

Conventionally, machine learning models are generated by collecting massive amounts of data and training on a centralized server system. Despite the fact that this method may produce highly accurate models thanks to a pool of data and computing resources, it suffers from the critical issue of privacy leakage. Federated Learning (FL) has recently emerged as a promising alternative for training machine learning models in a decentralized and privacy-guaranteed manner [1]. In an FL paradigm, multiple clients train models using their data locally and then send those local models to an orchestration server. The server aggregates all the local updates received from the clients to form a global model. The two factors that have the greatest impact on the performance of an FL model are the client-side training scheme and server-side aggregation algorithm.

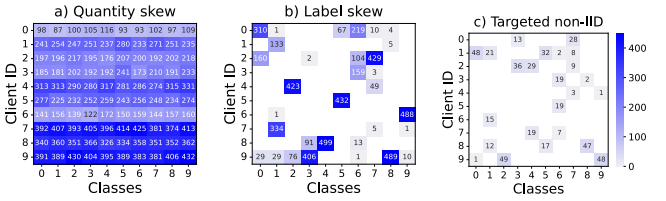

Fig. 1. Illustration of the conventional (a) quantity skew [2], (b) label-skew [16] non-IID types. In this work, We target to a hard scenario (c) where clients hold small number of classes (data sparsity) and a tiny amount of data (local data scarcity).

One of the most common FL algorithm is FedAvg [2], which trains the client models by using gradient descent and aggregates local updates through the weighted averaging. FedAvg works quite well if the clients' data are independent and identical (iid) and each client's data is large enough [2]. In practice, however, client data distribution is frequently not independent and identical (non-iid), i.e., in many instances [3], [4]. Such non-iid data may pose a challenge in designing an effective aggregation mechanism [5]. Specifically, standard FL methods such as FedAvg [2] has been reported to be not well-designed to address the challenges of non-IID data. It could degrade the training accuracy and increase the training time in practice [3], [5], [6]. In this context, numerous efforts have been devoted to tackle the non-iid issue in FL including (i) actively selecting clients involved in a training round for balancing the data distribution [7]–[9], (ii) re-weighting each client's impact factor when aggregation at the server-side [10]–[13], and (iii) optimizing the algorithms of local training scheme at the client-side [6], [11], [14], [15].

However, the existing approaches do not fully cover common patterns of data distribution encountered in practice. Indeed, most previous approaches regard non-IID data as a context of (i) *quantity skew non-IID*, i.e., different clients hold different amounts of data, e.g., [2], [6], [10], [17]

§Corresponding authors

2379-1896/23/$31.00 ©2023 IEEE
DOI 10.1109/CANDAR60563.2023.00023

or (ii) *label skew non-IID*, i.e., the label distribution differs between clients such as by distributing samples of a particular label[1] to the clients following the power-law or Dirichlet distribution [5], [11], [12][2]. Unfortunately, none of the existing approaches inadequately considered the issue of **data sparsity and scarcity**. This is a situation in which each client may hold only a tiny amount of data (scarce data) and/or a small number of data labels (sparse data), i.e., a hard case of label skew non-IID. As shown in Figure 1(c) the number of labels in a client's dataset is significantly smaller compared to that of the collective labels among all clients. Furthermore, the number of samples per client is also small, e..g, less than 50 samples. Such data problem is rather intrinsic to FL, as edge devices can only retain a limited quantity of raw data, especially in the medical domain. In contrast to centralized techniques, we can not augment or enrich the local data due to restricted data accessibility. Thus, it relies on novel strategies to take advantage of collaborative training to alleviate the problem. Sparse and scarce data typically result in rapid overfitting phenomenon in the local training on the client side (i.e., local models overfit after only a few training epochs), resulting in a severe degradation in global model accuracy [19].

To bridge this gap, this paper is an early attempt at investigating the **D**ata **S**parsity and **S**carcity problems (hereafter, we name this problem as **DSS**) in FL, and proposes a novel training and aggregation approach that considerably enhances the FL model's accuracy. Notably, our proposed solution is orthogonal and can be combined with other FL algorithms to enhance the model's overall performance.

Our primary idea is to make each local model trained by the data from several clients before aggregating by the server. To accomplish this, the most naïve method is to exchange the client's data with each other. Nevertheless, this approach contradicts the privacy premise of the FL paradigm. To this end, we employ the transfer learning technique and propose a *two-step training strategy* on the client side. Particularly, the local training process in each communication round is divided into two steps: *initial training* and *transfer learning*. Intuitively, in the first step, every client receives the global model from the server and trains using their local data. The trained models obtained from the first step are not delivered to the server but rather passed to other clients to perform the second step. In the second step, upon obtaining a trained model from another client, each client executes transfer learning and incrementally trains the model using its own local data. This way, after the second step, all the local models are trained with the data from two clients. Finally, these trained models are uploaded to the server for aggregation.

Our main contributions are as follows.

- We investigate the data sparsity and scarcity in FL. We propose a simple yet efficient novel FL model named SEM, that can mitigate the challenges posed by data sparsity and scarcity and enhance the accuracy of the final global model. Our proposed method is orthogonal and can be combined with other techniques in FL.
- We perform extensive experiments against various data sparsity and scarcity settings to evaluate the proposed method's efficacy. The experiment results on MNIST, CIFAR-10 and CIFAR-100 demonstrate that SEM can improve the global model's accuracy by up to 12.48%.

The remainder of the paper is organized as follows. In Section II-B, we briefly discuss relevant works and their limitations in dealing with data sparsity and scarcity. We describe the detail of our proposed method in Section III, and evaluate its performance in Section IV section. Finally, we conclude our work in Section V.

II. Background and Related Works

A. Overview of Federated Learning

A traditional FL model [2] constitutes of a global server and N clients, each with its private data. Let $\mathcal{C} = \{c_1, c_2, ...c_N\}$, $\mathcal{D} := \{\mathcal{D}_1, \mathcal{D}_2, ...\mathcal{D}_N\}$ denote the set of clients and their corresponding local datasets, respectively. The workflow of the conventional FL is as follows. For every communication round t, the server samples a subset S_t of K clients participating in the training process, and sends the global model ω_g^t to those clients. The participating clients, upon receiving the global, perform local training on its data and update the local model using the following gradient descent algorithm:

$$\omega_i^{(t,e+1)} \leftarrow \omega_i^{(t,e)} - \eta \nabla \mathcal{L}_i(\omega_i^{(t,e)}),\ \omega_i^{(t,0)} \leftarrow \omega_g^t, \quad (1)$$

where, e denotes the epoch, η represents the learning rate, and $\mathcal{L}_i(x) := \mathbb{E}_{\zeta \in \mathcal{D}_i}[l(x, \zeta)]$ is the empirical error. After finishing the training process in E epochs, the clients send back the local upates to the server for aggregation:

$$\omega_g^{t+1} = \textbf{Aggregate}\big(\{\omega_i^{(t,E)}\}, \forall i \in S_t\big). \quad (2)$$

The explicit expression of the aggregating function may vary and is the subject of study in many works [3], [4], [10], [13]. The procedure iterates until convergence or when a desirable model is found. Generally, the objective of an FL system is to produce a model that minimizes the total empirical error over all clients:

$$\omega^* = arg\min_{x \in \mathbb{R}^d} \{f(x) := \frac{1}{N} \sum_{i=1}^{N} \big(\mathcal{L}_i(x)\big)\}. \quad (3)$$

B. Federated Learning with non-IID data

Federated Averaging (FedAvg) [2] is arguably the most widely used algorithm in Federated Learning paradigm due to its simplicity. However, such simplicity comes at the expense of data heterogeneity tolerance: FedAvg is sensitive towards non-iid data distribution amongst clients.

Regarding the ubiquity of data heterogeneity in real-world scenarios, later works have proposed many techniques to mitigate its effects. Authors in [3], [10]–[13] proposed an approach to dynamically assign an impact factor a_i for each client i when aggregating the global model at the server, i.e., $\omega_g^{t+1} = \frac{1}{N} \sum_{i=1}^{K} a_i \omega_i^{(t,E)}$. Especially, FedDRL [3] trained a

[1]We use the terms *labels* and *classes* interchangeably in this work.

[2]See more detail of non-IID data's categories in [3], [16], [18].

reinforcement learning agent to re-weigh each client's impact factor based on their inference loss. The authors in [10] mitigated the impacts of non-iid data by reducing the gradient conflicts between local models based on their updating similarity. In another approach, FedProx [6] introduces an $L2$-normalization term between the global model and the local model as a proximal regularization loss. FedDyn [14] is based on the same client-side regularization approach, but rather uses an adaptive strategy to dynamically compute the regularization term. Scaffold [15] tries to correct the gradient-descent's direction at client-side using the collective gradient sum of all clients.

However, the aforementioned studies assume that each client has sufficient data to adequately train a model of average size. Nonetheless, this is rarely the case [20]–[24]; Data Sparsity and Scarcity (DSS) is always present. Despite the fact that data scarcity is not a novel topic of study in Computer Science, it has not been thoroughly investigated in the context of Federated Learning. Traditionally, two viable ways for centralised training are data augmentation [25] and domain adaption transfer learning [26], [27]. Although the former attempts to expand the amount of data by applying various data transformations, the later utilises pretrained large models and then applies few-shot training techniques to transfer the model to the new domain. Both of these approaches are highly effective at mitigating the effect of data scarcity, but neither is feasible within the Federated Learning framework. Firstly, data augmentation necessitates the compromising of client data, whose confidentiality is a core principle of FL. Secondly, client devices are not affordable for large pretrained models due to hardware-related computational limitations. On the other hand, data sparsity (or label sparsity) has several metrics that are specialised to particular disciplines (e.g. graph sparsity, matrix sparsity). In this work, we adopt the matrix sparsity measurement to the data distribution matrix across all clients, which is the ratio of non-zero elements to the total number of elements.

The problem of DSS in Federated Learning has been raised occasionally in recent works [20]–[24]. However, only the authors in [21] proposed a standalone solution for the issue. In contrast, we focus on utility, as our solution is a plug-in extension that can be integrated into any existing methods to improve their ability to withstand DSS's havoc.

III. Proposed Method

In the following, we first introduce our motivation in Section III-A and then present the details of our proposed method in Section III-B.

A. Motivation

As stated previously, in the conventional FL approach, each client trains its local model using only its own data. However, when a client's data is sparse and/or scarce, local models usually overfit the client's local data. This issue can be alleviated by training each local model with data from as many clients as possible. A naive method to accomplish this is to let

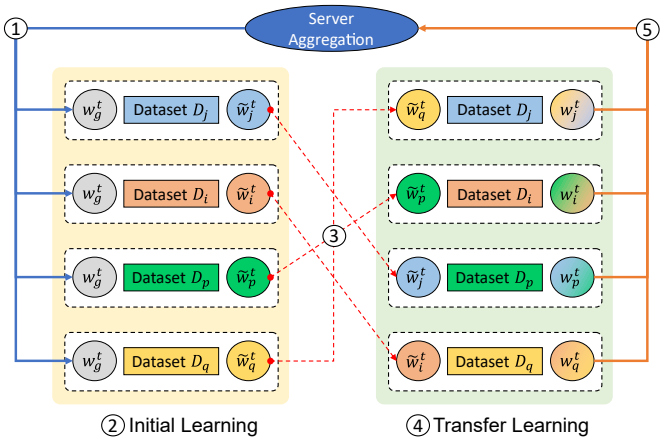

Fig. 2. **Overview of SEM**. ① Global model dissemination ② Initial Learning ③ Random matching ④ Transfer Learning ⑤ Aggregation

the clients exchange their data. However, this solution violates the privacy constraint of FL. To this end, our idea is to exploit the transfer learning technique, whereby each client will train a local model with its dataset and then send that trained model to another client, where it will be incrementally trained with the dataset of the new client. Delivering the learned local model to several clients and performing transfer learning can benefit the local model in acquiring more knowledge from numerous datasets. However, on the other hand, this transfer learning mechanism also may result in a so-called catastrophic forgetting phenomenon, in which the model forgets previously learned knowledge while attempting to acquire new information from a new dataset. Therefore, through empirical studies, we recommend using two clients to train each local model in each communication round. This number of clients strikes a balance between obtaining new knowledge from the new training data set while alleviating the catastrophic forgetting.

B. Two-step Local Training Mechanism

In the following, we present the details of our proposed method, SEM, which stands for **S**imple yet **E**fficient local training **M**echanism, to tackle the DSS (i.e., Data Sparsity and Scarcity) problem in FL. SEM comprises two training steps: *initial training* and *transfer learning*. Figure 2 illustrates the workflow of SEM, and Algorithm 1 provides the pseudo-code.

Let $S_t \subseteq [N]$ $(|S_t| = K)$ represent the set of participating clients at communication round t and E be the number of training epochs on the client side for each communication round. At the beginning of each communication round t, the server provides each client $i \in S_t$ the global model ω_g^t. In addition, the server employs a so-called *random pairing* algorithm that randomly identifies each client's pairing partner and sends this information to every participating client. Specifically, for each client, $i \in S_t$, its pairing partner in round t is denoted by p_i^t. Each client executes the *initial training* step by training ω_g^t with its own data in $E/2$ epochs to produce the local model $\tilde{\omega}_i^t$. At the end of the *initial training* step, each client i delivers its trained local model $\tilde{\omega}_i^t$ to its pairing partner, p_i^t. During the

Algorithm 1: SEM: Two-step training mechanism

Input : N clients with separated data-set, the number of clients performs training each round K, a pairing method $SEM(\cdot)$ and a baseline algorithm **BASE**.

Server: Initiate global model ω^0;
for *communication round* t *from* 0 *to* T **do**
  Select a subset S_t of random K clients;
  Server computes the matching map $\{i, SEM(i)\}$ for all client $i \in S_t$;
  for *each client* i *in* S_t ***in parallel*** **do**
    Receive the global model ω_g^t and its matching $SEM(i)$;
    Initialize $\omega_i^{(t,0)} \leftarrow \omega_g^t$
    A performs local training in $E/2$ epochs with batch b following **BASE**
    Send $(n_i, \tilde{\omega}_i^t)$ to its matching $SEM(i)$;
  end
  for *each matching client* $SEM(i)$ *in* S_t ***in parallel*** **do**
    Receive the model $\tilde{\omega}_i^t$ and the corresponding trained volume n_i;
    Initialize $\omega_{SEM(i)}^{(t,0)} \leftarrow \tilde{\omega}_i^t$
    Perform local training in $E/2$ epochs with batch b following **BASE**
    Send $(n_i + n_{SEM(i)}, \omega_{SEM(i)}^t)$ to the Server;
  end
  Server performs aggregation following **BASE**
end

transfer learning step, each client will use the model received from its pairing partner and incrementally train the model using its own data [3]. Finally, the locally trained models from our two-step training strategy are transmitted to the server for aggregation.

C. Assumption and Limitations

The assumptions and limitations of the SEM are as follows:

Targeted Federated Learning model: It is worth noticing that SEM does not require any adjustments or assumptions to the FL model. Consequently, it is compatible with any FL system. We demonstrate the flexibility of SEM in integration into various existing state-of-the-art FL methods in our evaluation in Section IV.

Client pairing algorithm: Although in this study we employ a random pairing policy to identify the paired partner for each client, different pairing methods may be used. In addition, in this work, we empirically recommend using only two clients to train a local model in each communication round to balance the knowledge gaining and forgetting in the transfer learning scheme. How to choose (dynamically) this number for different deep learning models, datasets, and DSS scenarios will be (theoretically) discussed in our future work.

Strategy for communication between clients: To address the issues of data sparsity and scarcity, we proposed a paradigm where different clients exchange their local model with other clients in each communication round. In some conventional centralized federated learning frameworks, direct communication between clients may not be permitted due to security and privacy issue. In this case, we suggest allowing clients to communicate through the server. That is, after the first step of the communication round t, the local models from K participated clients are sent to the server. The server then forwards these K local models to k other randomly-selected clients without aggregating. After that, the second step (transfer learning step) and aggregation are performed as usual.

[3]Similar to the transfer learning scheme, where a model, which is pre-trained on a big dataset, is trained on another dataset. The knowledge that the model learned from the first dataset is transferred to the training process of the second dataset.

IV. Evaluation

In this section, we evaluate the effectiveness of SEM when integrated into various existing state-of-the-art FL methods in different datasets and DSS scenarios. In this evaluation, we choose the state-of-the-art FL methods that focus on designing both aggregation mechanism at the server (FedAvg [2], FedFA [28], and FedFV [10]) and training algorithm at clients (FedProx [6] and Scaffold [15]) To ease the presentation, we refer to the original FL methods as (**Base**) and those with SEM as **w/SEM**. We demonstrate that **w/SEM** outperforms **Base** with respect to all scenarios. In the following, we first introduce the details of the DSS scenarios and the FL settings used in our experiments in Section IV-A. We then report and compare the testing accuracy of **w/SEM** and **Base** in Section IV-B. In all the experiments, SingleSet (the setting of gathering all the local data of clients into a centralized server and training) is used to refer to the testing accuracy's upper bound.

A. Datasets and Experimental Settings

In this work, we pick up datasets and DNN models used from prior works in FL [2], [3], [6], [14], [28]. In detail, we use three different FL datasets, i.e., MNIST [29], CIFAR-10, and CIFAR-100 [30]. For each dataset, we simulate the non-iid setting by partitioning the dataset and distributing the training samples among N clients according to the Dirichlet distribution. To study the efficiency of our method with different data sparsity and scarcity scenarios, we define the data sparsity and scarcity as follows. Let A be the matrix in which an element in column i and row j is the number of samples of client i belonging to the data label j. The fraction of zero elements in the matrix A is called the **data sparsity**. In addition, **data scarcity** is defined as the number of clients (in percentage) that have total samples no greater than a threshold T [4].

To adjust the data sparsity for each setting, we vary the hyper-parameter α of the Dirichlet distribution in $\{0.1, 0.3, 0.5, 100\}$. Fig. 3 illustrates the data distribution of the CIFAR-10 dataset (10 data labels or classes) in the case of 100 clients with different α and data sparsity. The bigger α, the fewer data sparsity. When $\alpha = 100$, any client holds all the data labels, leading to zero data sparsity. Furthermore, we

[4]We select $T = 50$ in this evaluation.

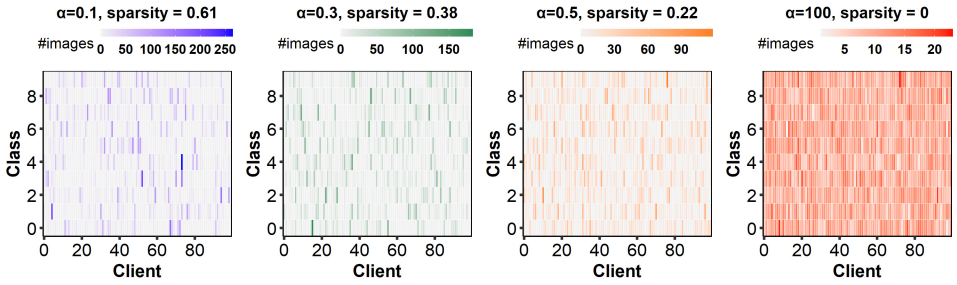

Fig. 3. Illustration of data distribution of CIFAR-10 dataset in the case of 100 clients with different non-IID degree α and data-sparsity.

TABLE I

TOP-1 TESTING ACCURACY IN PERCENTAGES OF OUR PROPOSED METHOD (**W/SEM**) AND THE BASELINE FL METHOD (**BASE**). THE VALUES SHOW THE BEST ACCURACY THAT EACH ALGORITHM REACHES WITHIN 1000 COMMUNICATION ROUNDS. **IMPR.** REFERS TO THE RELATIVE IMPROVEMENT OF OUR METHOD OVER THE BASELINE. **SP** : DATA SPARSITY.

	Baseline Method	$\alpha = 0.1$				$\alpha = 0.3$				$\alpha = 0.5$			
		Sp	**Accuracy**			**Sp**	**Accuracy**			**Sp**	**Accuracy**		
			Base	**w/SEM**	**Impr.**		**Base**	**w/SEM**	**Impr.**		**Base**	**w/SEM**	**Impr.**
MNIST	SingleSet	0.63	99.05	-	-	0.33	98.99	-	-	0.20	98.98	-	-
	FedAvg		97.75	98.10	**0.36%**		98.13	98.31	**0.18%**		98.13	98.30	**0.17%**
	FedProx		97.77	98.07	**0.31%**		98.13	98.28	**0.15%**		98.14	98.32	**0.18%**
	FedFA		97.77	98.04	**0.28%**		98.01	98.26	**0.26%**		98.17	98.27	**0.10%**
	FedFV		97.74	98.10	**0.37%**		98.05	98.27	**0.22%**		98.16	98.32	**0.16%**
	Scaffold		98.63	98.62	**0.00%**		98.50	98.48	**0.00%**		98.58	98.65	**0.07%**
CIFAR-10	SingleSet	0.61	78.72	-	-	0.38	78.04	-	-	0.22	78.00	-	-
	FedAvg		64.05	66.13	**3.24%**		65.15	67.39	**3.44%**		67.90	68.96	**1.56%**
	FedProx		64.00	65.76	**2.75%**		65.41	67.26	**2.83%**		67.66	68.42	**1.12%**
	FedFA		63.55	65.33	**2.80%**		65.79	67.55	**2.68%**		67.69	69.01	**1.95%**
	FedFV		64.03	65.33	**2.03%**		65.54	67.48	**2.96%**		67.63	69.04	**2.08%**
	Scaffold		67.05	67.44	**0.58%**		68.88	69.92	**1.50%**		70.07	71.21	**1.63%**
CIFAR-100	SingleSet	0.78	44.94	-	-	0.64	44.48	-	-	0.58	44.79	-	-
	FedAvg		29.72	33.01	**11.06%**		30.79	34.24	**11.20%**		31.05	34.34	**10.60%**
	FedProx		29.81	32.87	**10.27%**		30.76	34.09	**10.83%**		30.97	33.96	**9.65%**
	FedFA		29.33	32.99	**12.48%**		30.41	33.75	**10.98%**		31.11	34.54	**11.03%**
	FedFV		29.84	33.04	**10.72%**		31.40	33.73	**7.42%**		31.32	33.96	**8.43%**
	Scaffold		32.30	35.63	**10.31%**		33.46	36.97	**10.49%**		34.04	37.45	**10.99%**

introduce an additional parameter $U\%$ (named data volume ratio) to adjust the data scarcity. That is, we use only $U\%$ of the data samples of the targeted dataset when distributing it to the N clients. Data scarcity is higher when U is smaller or N is bigger.

In this evaluation, we use the hyper-parameter picked up from the prior work. Especially, we train simple convolutional neural networks (CNNs) on MNIST [2] and ResNet-9 [31] network on CIFAR-10/CIFAR-100 dataset using stochastic gradient descent (SGD) as the local optimizer. We set the local epochs $E = 8$, a learning rate of 0.001, and a local batch size $B = 8$. We also use the suggested hyper-parameters from the original papers of the baseline FL methods. For example, we set the proximal term $\mu = 0.01$ for the FedProx method. We use $\gamma = 0$ and $\beta = 1.0$ for FedFA; and $\tau = 0$ for FedFV. For Scaffold, we set $\eta = 1.0$.

B. Experimental Results

1) Top-1 Accuracy: Table I presents the experimental results of the best top-1 testing accuracy that a method reaches within 1000 communication rounds. Specifically, integrating our proposed method with all the baseline FL methods (w/SEM) achieves better testing accuracy than those using only the original baseline FL methods (Base) in most of the experiment settings. For example, when $\alpha = 0.1$, using the proposed method helps to relatively increase the accuracy around $0.28 - 0.37\%$, $2.03 - 3.24\%$, and $10.27 - 12.48\%$ for MNIST, CIFAR-10, and CIFAR-100 datasets, respectively. The accuracy improvement in the MNIST dataset is trivial in comparison with the improvement in the CIFAR-10 and

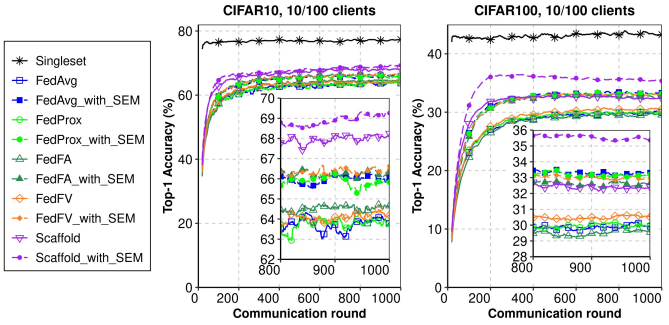

Fig. 4. Top-1 testing accuracy vs. communication round on the CIFAR-10 ($\alpha = 0.3$, sparsity = 0.38) and CIFAR-100 dataset ($\alpha = 0.3$, sparsity = 0.64). The results are plotted with the average-smoothed of every 20 round to have a better visualization.

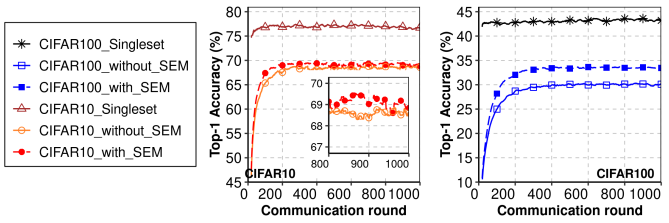

Fig. 5. Top-1 testing accuracy vs. communication round on the CIFAR-10 and CIFAR-100 dataset with $\alpha = 100$ and sparsity of 0 and 0.37 respectively. The results are plotted with the average-smoothed of every 20 round to have a better visualization.

CIFAR-100 datasets. Especially, there is no improvement in accuracy by using our random matching method in the case of the Scaffold method with the MNIST dataset. Because of the simplicity of the image classification tasks on the MNIST dataset, all the baseline FL methods, especially Scaffold, achieve accuracy nearly asymptotic to the upper bound (SingleSet), thus no/trivial room for improvement. In harder classification tasks, e.g., in CIFAR-10 and CIFAR-100, there is more room for improvement, and our proposed method achieves significant improvements in accuracy. This result emphasizes that our proposed method can address the challenge of non-iid or heterogeneous data more effectively.

2) Convergence Analysis: In the previous sub-section, we focus on the top-1 testing accuracy to gauge the goodness of the trained model over the global distribution. We now estimate the number of communication rounds necessary to achieve the desired top-1 accuracy to show how the proposed method effectively reduces local computation at clients. Figure 3 displays the trend of top-1 accuracy versus communication rounds for the CIFAR-10 and CIFAR-100 datasets and the non-iid degree α of 0.3. We smoothed the curve by averaging the accuracy in the last 20 communication rounds to have a better visualization. As shown in Figure 4, the convergence rate of the methods that used random matching is faster than those of the corresponding baseline method. For instance, Scaffold requires 887 communication rounds to reach its best top-1 accuracy, i.e., 68.88%. With the same setting, the Scaffold method with random matching needs only 507 communication rounds, yielding $1.75\times$ speedup. We confirm the same trends with other baseline methods and all the settings mentioned in Table I.

TABLE II
IMPACT OF RANDOM MATCHING WITH FEDAVG WITHIN 1000 COMMUNICATION ROUND IN CIFAR-10 DATASET WHEN VARYING THE DATA VOLUME RATIO U.

N	K	α	U	Sp	$Scarcity_{T=50}$	SingleSet	FedAvg	FedAvg w/SEM	Impr.
100	10	0.1	5%	0.72	0.85	67.69	52.40	53.61	**2.31%**
			10%	0.67	0.58	73.00	57.44	59.04	**2.79%**
			15%	0.63	0.45	76.56	61.16	63.43	**3.71%**
			20%	0.61	0.30	78.72	64.05	66.13	**3.25%**
			100%	0.54	0.08	88.55	71.05	73.81	**3.88%**

TABLE III
IMPACT OF RANDOM MATCHING WITH FEDAVG WITHIN 1000 COMMUNICATION ROUND IN CIFAR-10 DATASET WHEN THE NUMBER OF CLIENT N INCREASES.

N	K	α	U	Sp	$Scarcity_{T=50}$	SingleSet	FedAvg	FedAvg w/SEM	Impr.
500	50	0.5	80%	0.23	0.21	87.03	75.43	77.27	**2.44%**
1000	100			0.34	0.74	86.74	74.45	76.31	**2.50%**
1500	150			0.40	0.94	86.07	73.59	75.81	**3.02%**
2000	200			0.45	0.99	86.15	72.55	74.94	**3.29%**

3) Robustness to the Data Sparsity and Scarcity: We study the robustness of the proposed method with different degrees of data sparsity. Table I shows that the data sparsity decreases when reducing the non-iid level, i.e., by increasing α. It leads to a higher top-1 accuracy achieved by all the baseline methods. It also shows that our proposed method helps to improve the accuracy not only in the case of high data sparsity, e.g., CIFAR-100, $\alpha = 0.1$ but also in the case of low data sparsity, e.g., CIFAR-10, $\alpha = 0.5$. Even with the zero data sparsity [5], e.g., in the case of CIFAR-10, $\alpha = 100$, our proposed method still outperforms the accuracy of FedAvg around 1.2% (Figure 5). This result proves the effectiveness of the proposed method in both non-iid and iid datasets.

To study the impact of data scarcity on our method, we vary the data volume ratio U and the number of clients N while fixing α. Table II (Table III) presents the data scarcity and the top-1 accuracy of our method versus FedAvg in 1000 communication rounds when U (N) is changed. The result shows that the higher the data scarcity, the lower the accuracy achieved. For example, with the data scarcity of 0.85, i.e., ($U = 5\%$) FedAvg and our method reach 52.4% and 53.61% of accuracy, respectively. Those are 71.05% and 73.81% when the data scarcity is 0.08 ($U = 100\%$). However, It is worth noting that with all the significance of data scarcity, the proposed method still helps to improve the accuracy around $2-3\%$. The result implies that our proposed method has a good performance with different types of datasets and FL settings.

[5]the data distribution is nearly iid (Figure 3).

V. Conclusion

In this paper, we addressed the DSS problem in FL. We proposed SEM, a simple yet effective training mechanism on the client side to enhance the data trained for each local model in every communication round. The main idea is to train each local model with data from two randomly paired clients in every round. We performed extensive experiments with various data sparsity and scarcity settings on MNIST, CIFAR-10, and CIFAR-100. The experiment findings indicated that by integrating SEM, the global model's accuracy can be relatively increased by up to 0.37% against MNIST, 3.44% against CIFAR-10, and 12.48% against CIFAR-100, respectively. The future effort will be devoted to improving the algorithm for pairing clients.

VI. Acknowledgments

This work was supported by JSPS KAKENHI under Grant Number JP21K17751. This work also was funded by Vingroup Joint Stock Company (Vingroup JSC), Vingroup, and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2021.DA00128

References

[1] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[2] B. McMahan, E. Moore *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[3] N. H. Nguyen, P. L. Nguyen, T. D. Nguyen, T. T. Nguyen, D. L. Nguyen, T. H. Nguyen, H. H. Pham, and T. N. Truong, "Feddrl: Deep reinforcement learning-based adaptive aggregation for non-iid data in federated learning," in *Proceedings of the 51st International Conference on Parallel Processing*, 2022, pp. 1–11.

[4] N. H. Nguyen, D. L. Nguyen, T. B. Nguyen, T.-H. Nguyen, H. H. Pham, T. T. Nguyen, and P. L. Nguyen, "Cadis: Handling cluster-skewed non-iid data in federated learning with clustered aggregation and knowledge distilled regularization," *arXiv preprint arXiv:2302.10413*, 2023.

[5] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[6] T. Li, A. K. Sahu *et al.*, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[7] Y. J. Cho, J. Wang *et al.*, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[8] H. Wang, Z. Kaplan *et al.*, "Optimizing federated learning on non-IID data with reinforcement learning," in *IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 1698–1707.

[9] D. Y. Zhang, Z. Kou, and D. Wang, "FedSens: A federated learning approach for smart health sensing with class imbalance in resource constrained edge computing," in *IEEE Conference on Computer Communications (INFOCOM 2021)*, 2021, pp. 1–10.

[10] Z. Wang, X. Fan, J. Qi, C. Wen, C. Wang, and R. Yu, "Federated learning with fair averaging," *2021 International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[11] J. Wang, Q. Liu *et al.*, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5234–5249, 2021.

[12] W. Huang, T. Li *et al.*, "Fairness and accuracy in federated learning," *arXiv preprint arXiv:2012.10069*, 2020.

[13] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1078–1088, 2021.

[14] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.

[15] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[16] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[17] P. Xiao, S. Cheng, V. Stankovic, and D. Vukobratovic, "Averaging is probably not the optimum way of aggregating parameters in federated learning," *Entropy*, vol. 22, no. 3, p. 314, 2020.

[18] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 13–18 Jul 2020, pp. 4387–4398.

[19] A. Barros, D. Rosário, E. Cerqueira, and N. L. da Fonseca, "A strategy to the reduction of communication overhead and overfitting in federated learning," in *Anais do XXVI Workshop de Gerência e Operação de Redes e Serviços*. SBC, 2021, pp. 1–13.

[20] D. Ng, X. Lan, M. M.-S. Yao, W. P. Chan, and M. Feng, "Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets," *Quantitative Imaging in Medicine and Surgery*, vol. 11, no. 2, p. 852, 2021.

[21] M. Kamp, J. Fischer, and J. Vreeken, "Federated learning from small datasets," *arXiv preprint arXiv:2110.03469*, 2021.

[22] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-task network anomaly detection using federated learning," in *Proceedings of the 10th international symposium on information and communication technology*, 2019, pp. 273–279.

[23] Y. Zhang, G. Tang, Q. Huang, Y. Wang, K. Wu, K. Yu, and X. Shao, "Fednilm: Applying federated learning to nilm applications at the edge," *IEEE Transactions on Green Communications and Networking*, 2022.

[24] Y. Zhao, J. Chen, Q. Guo, J. Teng, and D. Wu, "Network anomaly detection using federated learning and transfer learning," in *Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, October 30–November 1, 2020, Proceedings*. Springer, 2020, pp. 219–231.

[25] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[26] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[27] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[28] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, and T. Huang, "Fairness and accuracy in horizontal federated learning," *Information Sciences*, vol. 589, pp. 170–185, 2022.

[29] Y. Lecun, L. Bottou *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Dep. of Comp. Sci. Univ. of Toronto*, 2009.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.