






Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training With Non-IID Private Data

Sohei Itahara , *Student Member, IEEE*, Takayuki Nishio , *Senior Member, IEEE*,
Yusuke Koda , *Member, IEEE*, Masahiro Morikura , *Member, IEEE*,
and Koji Yamamoto , *Senior Member, IEEE*

Abstract—This study develops a federated learning (FL) framework overcoming largely incremental communication costs due to model sizes in typical frameworks without compromising model performance. To this end, based on the idea of leveraging an unlabeled open dataset, we propose a distillation-based semi-supervised FL (DS-FL) algorithm that exchanges the outputs of local models among mobile devices, instead of model parameter exchange employed by the typical frameworks. In DS-FL, the communication cost depends only on the output dimensions of the models and does not scale up according to the model size. The exchanged model outputs are used to label each sample of the open dataset, which creates an additionally labeled dataset. Based on the new dataset, local models are further trained, and model performance is enhanced owing to the data augmentation effect. We further highlight that in DS-FL, the heterogeneity of the devices' dataset leads to ambiguous of each data sample and lowering of the training convergence. To prevent this, we propose entropy reduction averaging, where the aggregated model outputs are intentionally sharpened. Moreover, extensive experiments show that DS-FL reduces communication costs up to 99 percent relative to those of the FL benchmark while achieving similar or higher classification accuracy.

Index Terms—Federated learning, knowledge distillation, non-IID data, communication efficiency

1 INTRODUCTION

FEDERATED Learning (FL) [1], [2], [3], [4] is an emerging machine learning (ML) framework to perform data-driven analysis or decision making, leveraging privacy-sensitive data from mobile devices. Typically, in FL, mobile devices collaboratively train their local ML model through the periodical exchange and aggregation of ML model parameters or gradients at central servers rather than exchanging their raw data. Thus, FL differs from typical ML in which raw data is acquired and stored in central servers where the private data of the mobile users can be exposed. Owing to the privacy advantage, FL can be applied to model training tasks with privacy-sensitive data. For example, Google-keyboard query suggestions from the typing history of mobile users, containing privacy-sensitive information such as the credit card information of the user or login credentials [5].

Despite the benefits of FL, relying on distributed mobile devices generally poses new inconveniences related to communication efficiency [1]. Specifically, the periodical model parameter exchange in typical FL entails communication

overhead that scales up according to the model size. This prohibits the use of large-sized models, particularly when the mobile devices are connected to wireless networks while competing for limited radio resources, which can be a crucial bottleneck for building practical ML models. Hence, an FL framework that can be scalable according to the size of the models in terms of communication efficiency is required.

Motivated by the inconvenience mentioned above, we aim to answer the following question: *How should an FL framework be designed scalable according to the model sizes in terms of communication efficiency while achieving model performance comparable to that of the benchmark FL designed in [4]?* Concisely, our answer is leveraging an unlabeled open data shared among the clients to enhance the model performance of model output exchange methods.

To achieve the scalability of the communication overhead, we leverage the principle of FL with *model output exchange* instead of model parameter exchange. Here, the exchanged model outputs of mobile devices are named “local logit” instead of model parameters. The local logits are ensembled in a central server into a “global logit” that is regarded as *teacher* knowledge. Moreover, this knowledge is transferred into local models as *students*. In the model output exchange, communication overheads depend only on the model output dimension, which is often substantially smaller than the number of model parameters and cannot scale up regarding model sizes.

Hence, answering the above question boils down to designing an FL framework with model output exchange to achieve similar performance to the benchmark FL designed

- The authors are with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan. E-mail: {itahara, koda}@imc.cce.i.kyoto-u.ac.jp, {nishio, morikura, kyamamoto}@i.kyoto-u.ac.jp.

Manuscript received 13 Aug. 2020; revised 17 Mar. 2021; accepted 26 Mar. 2021.

Date of publication 31 Mar. 2021; date of current version 5 Dec. 2022.

(Corresponding author: Sohei Itahara.)

Recommended for acceptance by J. Tang.

Digital Object Identifier no. 10.1109/TMC.2021.3070013

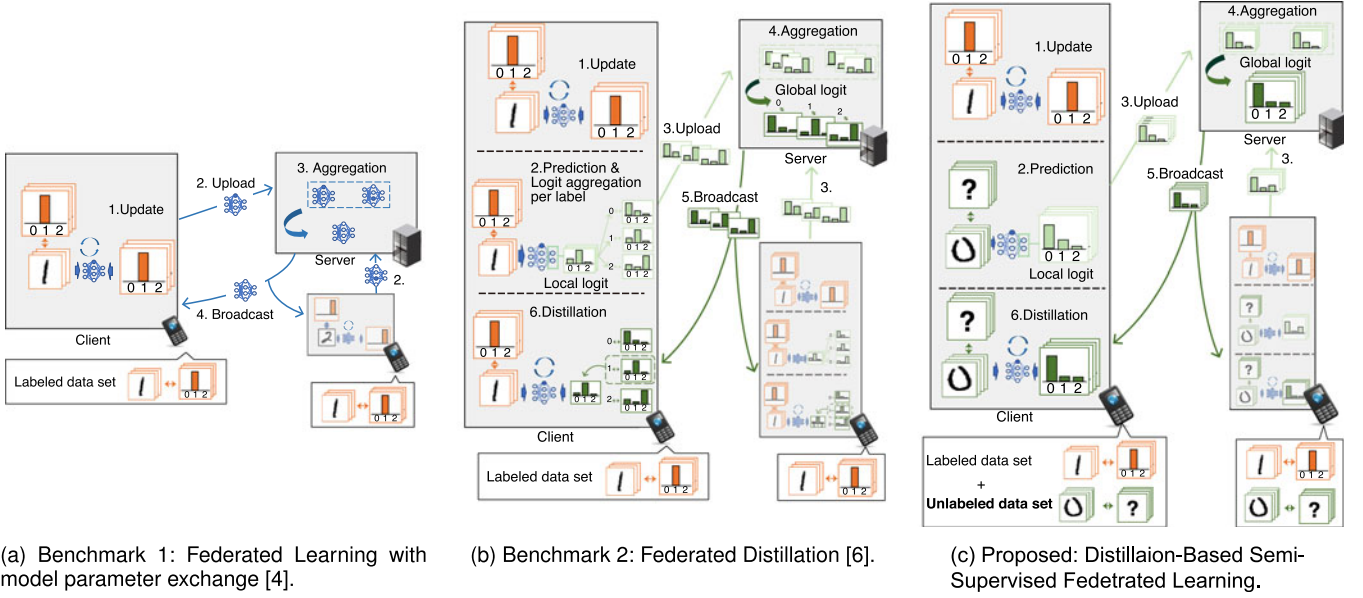


Fig. 1. Operational structures for benchmark schemes and proposed DS-FL.

in [4]. Although the FL with model output exchange is available in the literature, this task remains nontrivial because of the several challenges we describe below.

Leveraging unlabeled data towards performance similar to that of benchmark federated learning under non-IID. Typical FL with model output exchange termed federated distillation (FD) [6], [7], [8], [9] can achieve scalability of the model size; however, these methods provide poor models in general. In FD, each mobile device re-trains local model based on both the local labeled data and the global logits. However, under non-IID data distribution, where the local dataset of the mobile user does not represent the population distribution, the global logit retains similar information to the local labels already attached to each mobile device. Hence, this re-training can be almost identical to local model training. (see "1. Update" and "6. Distillation" procedures in Fig. 1b in Section 2 for detail). Thus, the model performance trained in FD is worse than that in an FL benchmark exchanging model parameters [8]. Moreover, achieving similar performance to the FL benchmark is challenging.

Our key idea is to share the unlabeled open data among mobile devices and leverage the data for distillation to overcome this challenge. In this regard, we propose a novel FL framework-exchanging model outputs named distillation-based semi-supervised FL (DS-FL). Unlike FD, in the proposed DS-FL, teacher knowledge is used to label the unlabeled data instead of local data already labeled. The procedure creates novel labeled data. Subsequently, through the re-training of the local models based on the dataset, the model performance is enhanced due to data augmentation effects (see "6. Distillation" procedure in Fig. 1c in Section 2 for detail). The ML experiments show that the proposed DS-FL achieves similar or higher performance compared to that of the FL benchmark with model parameter exchange while reducing communication overheads. To the authors' best knowledge, our approach has not been considered before.

Logit Aggregation Towards Faster Convergence. In DS-FL, uploaded logits are aggregated into a global logit, providing an inferred probability of each unlabeled data belonging to

a particular class. However, owing to non-IID data distributions, each uploaded logit exhibits heterogeneity. Moreover, the aggregated global logit may represent *ambiguous knowledge* where the global logit exhibits a high entropy. Thus, indicating that the global logit provides incorrect knowledge regarding which class each sample of the unlabeled data pertains, resulting in slower convergence of model training.

Motivated by the challenge mentioned above, we propose entropy reduction aggregation (ERA) that intentionally reduces the global logit entropy. The ML experiments show that DS-FL with ERA leads to faster convergence while achieving higher classification accuracy than an FL benchmark under non-IID distribution. Another positive consequence of introducing ERA is to enhance the robustness against several attacks of malicious users uploading corrupted local logits, which is also verified in the ML experiments.

1.1 Our Contributions

The contributions of this paper are summarized as follows:

- Based on the fundamental idea of leveraging unlabeled open data, we propose an FL framework named DS-FL, which is communication-efficient and achieves a high model performance. In more detail, DS-FL exhibits a model performance similar to that of an FL benchmark even under non-IID data distributions while achieving scalability according to model size in terms of communication efficiency. With the exchanged model outputs, the unlabeled data acquire labels, based on which each local model can be further trained. Therefore, model performance is enhanced due to the data augmentation effects. Moreover, DS-FL exchanges model outputs instead of model parameters, where the communication overhead cannot scale up according to the model size. The ML experiments show that DS-FL reduces the communication overheads by up to

99 percent while achieving similar model performance relative to the FL benchmark even under non-IID data distributions.

- We develop a novel model-output aggregation method, named ERA, which is robust against the heterogeneity of the uploaded model output due to non-IID data distributions that result in a slow training convergence. First, we highlight that the heterogeneity of the uploaded model output leads to higher entropy of the aggregated model outputs, which is the principal cause of the challenging slower convergence. Hence, the key idea behind introducing ERA is to reduce the entropy of the global logit intentionally before distributing it into mobile devices. The ML experiments verify that ERA achieves faster convergence than a baseline considering simple averaging (SA) aggregation, thus, reducing the cumulative communication cost by up to 75 percent.

The scope of this paper is to design an FL framework, satisfying two requirements: to acquire scalability to the model size and perform well under non-IID data. The existing FL framework satisfying the former requirement is FD, whereas FD's performance is substantially low under non-IID data. Thus, designing an FL framework satisfying these two requirements contributes to the body of knowledge. Additional requirements, such as performing well under unbalanced and/or massively distributed data, are out-of-scope of the study.

In parallel with and independent of this work, a similar concept of sharing an unlabeled dataset among mobile devices has presented in [10]. However, the study relies on a different motivation; it is based on enhancing attack robustness from malicious mobile devices. Our initial study [11] was presented at a domestic conference in parallel to [10]. Meanwhile, unlike [10], in this study, we investigate how DS-FL can achieve similar performance to FL benchmark under non-IID distributions in an efficient communicational manner. More specifically, we provide a novel logit aggregation method, i.e., ERA, enhancing communication efficiency under a non-IID dataset. Moreover, we provide a comparison between DS-FL, FL benchmark, and FD through ML experiments using non-IID datasets, which were not considered in [10].

1.2 Related Work and Paper Organization

Federated Learning. FL [1], [2], [3], [4], [12] is a distributed learning framework enabling ML model training using privacy-sensitive datasets of mobile devices while keeping all the datasets local. In typical FL, mobile devices collaboratively train their local ML model through the periodical exchange and aggregation of ML model parameters or gradients at central servers rather than exchanging their raw data. Thus, the central server and mobile devices obtain a qualified ML model, trained using the private dataset on mobile devices without exposing the privacy-sensitive data.

Communication-Efficient Federated Learning With Model Parameter Exchange. Several studies have focused on reducing the communication cost in model parameter exchanges in FL. For example, the initial study that proposed FL addressed this problem by increasing the number of local model updates and exchanging the model parameters less frequently [4]. In

this approach, network traffic reduces drastically compared to that of algorithms considering iterating local model update and model parameter exchange alternately [13]. An alternative strategy is to limit the number of participating mobile users by selecting the users satisfying the stringent requirement for model update time [14], [15]. The network traffic can be reduced relative to all users participating in the FL. Other approaches generating an additional labeled dataset, distributed and used on the clients, are [6], [16]. Indeed, these approaches improve communication efficiency and model performance. Another stream of research proposed the model compression to reduce the communication cost required for the model parameter exchanges, which can be performed via several strategies, such as low-rank representation [17], model parameter quantization [17], [18], [19], neural network pruning [20], update reuse [21], parameter sparsification [17], [22], and the Chinese remainder theorem [23]. However, these studies relied on the model parameter exchange, where the communication-overhead increases proportionally to the model size. Unlike these studies, we aim to design an FL framework scalable for model size in terms of communication efficiency by designing an FL framework exchanging model outputs instead of the entire model parameters.

Distributed Training With Model Output Exchange in Data Center Application Over Shared Dataset. Co-distillation (CD) [24], [25] is a basic distributed learning method with model output exchange. In CD, distributed ML models are trained over a shared labeled dataset. Subsequently, each local logit from the trained distributed ML models is exchanged and aggregated into global logit acting as teacher knowledge. Finally, the teacher knowledge is transferred into each distributed ML model acting as a student by re-training the ML model using the global logit. Note that this framework is an extension of the knowledge distillation presented in [26] for multi-party training. Another distributed learning method similar to CD, named private aggregation of teacher ensembles (PATE), was proposed in [27], [28], where the teacher knowledge transfer is performed over the unlabeled dataset instead of the labeled dataset. While both CD and PATE improve each distributed ML model in a communication-efficient manner, the assumption that each distributed ML model is trained over a shared labeled dataset is suitable for parallel model training in the data center. However, the assumption is not suitable for model training with data generated on mobile devices. Unlike these training methods, we design an FL framework with model output exchange, enabling model training with mobile device-generated data subjected to challenging non-IID data distributions, as discussed in the previous section.

Federated Learning With Model Output Exchange Over Mobile Device-Generated Dataset. FD is proposed in [6], [7], [8], [9] as an FL framework with model output exchange that trains ML models considering mobile device-generated dataset. Unlike CD and PATE that trains distributed ML models using a shared dataset, each mobile device trains each ML model using a local dataset, enabling ML model training with mobile device-generated data. While FD performs well when the mobile device-generated data is identically and independently distributed, FD exhibits lower performance than the FL benchmark with model parameter exchange in non-IID data distributions. This was experimentally verified

in [6], [7], [8], [9] and the experiments presented in Section 4. To fill this gap, we design an FL framework with model output exchange achieving similar or higher performance than previously proposed approaches even when subjected to non-IID data distributions.

Semi-Supervised Federated Learning. A few semi-supervised FL frameworks [29, 30, 31], using both unlabeled and labeled data, have been proposed. The work in [29] aimed to improve vertical FL (VFL), which builds a machine learning model based on vertically partitioned data (e.g., multi-view images). In this setting, the model parameters or gradients are not generally uploaded, and hence, the communication cost is negligible. Being different from [29], as in the benchmark frameworks [4], [6], this study considers horizontal partitioned data and addresses the communication costs for uploading model parameters or gradients. Other works [30], [31] considered training the model using labeled data on the server and unlabeled data on the clients. While the communication costs in [30] and [31] increase with the model size, we aim to achieve communication efficiency scalable to the model sizes.

Paper Organization. The remainder of this paper is organized as follows: Section 2 describes the proposed DS-FL framework. Section 3 presents the proposed logit aggregation method. Section 4 provides the experimental results where a comparison between DL-FL, FL benchmark, and FD is presented. Finally, the concluding remarks are presented in Section 5.

2 DISTILLATION-BASED SEMI-SUPERVISED FEDERATED LEARNING METHOD

We propose a DS-FL aiming at communication efficiency while achieving similar or higher model performance than several benchmarks. We summarize the benchmark schemes and the proposed DS-FL in Fig. 1, detailed as follows.

2.1 Benchmark 1. Federated Learning With Model Parameter Exchange

In the FL with model parameter exchange, mobile users, called *clients* as per terminology, collaboratively train ML models while exchanging the model parameters, as shown in Fig. 1a. Specifically, the training procedure in FL with model parameter exchange includes four steps: “1. Update,” “2. Upload,” “3. Aggregation,” and “4. Broadcast.” These steps follow an iterative process until training converges. In “1. Update” step, every client trains its local ML model using its own labeled dataset. The “1. Update” step is common to the DS-FL. Subsequently, in “2. Upload” step, the clients share the model parameters with a remote server. Finally, the server aggregates the uploaded model parameters to build the global model in the “3. Aggregation” step and broadcasts the parameters of the global model to the clients in the “4. Broadcast” step.

The detailed procedure of FL is described below. In the following, we consider that each client $k = 1, 2, \dots, K$ holds the labeled private dataset $(\mathbf{d}_{i,k}^p, \mathbf{t}_{i,k})_{i=1}^{I_k}$, where $\mathbf{d}_{i,k}^p$ represents the vectorized input samples. Moreover, I_k denotes the number of samples in the labeled dataset. Considering N_L as the number of objective class, the term $\mathbf{t}_{i,k} = [t_{i,k,1}, \dots, t_{i,k,N_L}]^T$ is the vectorized form of the label attached

to the sample $\mathbf{d}_{i,k}^p$ and is in the one-hot representation, wherein the element $t_{i,k,n}$ equals 1 if the n th label is the ground-truth and 0 otherwise. For shorthand notation, let $N_S \times I_k$ matrix \mathbf{D}_k^p denote the concatenation of $(\mathbf{d}_{i,k}^p)_{i=1}^{I_k}$, where N_S represents the dimension of input samples.

1. *Update.* In this step, each client updates its model with its private dataset based on the stochastic gradient descent algorithm [32]. The initial values of the model \mathbf{w}_0 is distributed from the server before each “1. Update” step. Specifically, the model parameter is updated as follows:

$$\mathbf{w}_k \leftarrow \mathbf{w}_0 - \eta \nabla \phi(\mathbf{D}_k^p, \mathbf{T}_k | \mathbf{w}_0), \quad (1)$$

where $\phi(\cdot, \cdot | \mathbf{w}_k)$ denotes the loss function that is minimized in this step. The loss function is exemplified in classification problems by the cross-entropy. In this case, $\phi(\mathbf{D}_k^p, \mathbf{T}_k | \mathbf{w}_k)$ is given as follows:

$$\phi(\mathbf{D}_k^p, \mathbf{T}_k | \mathbf{w}_k) = - \sum_{i \in \mathcal{I}_k^{\text{rd}}} \sum_{n \in \mathcal{N}_L} t_{i,k,n} \log F_n(\mathbf{d}_{i,k}^p | \mathbf{w}_k), \quad (2)$$

where $F_n(\cdot | \mathbf{w}_k)$ denotes the n th element of $F(\cdot | \mathbf{w}_k)$. In (1) and (2), η represents the learning rate, $\mathcal{N}_L := \{1, 2, \dots, N_L\}$, and $\mathcal{I}_k^{\text{rd}} \subset \{1, 2, \dots, I_k\}$ is the index set of the minibatch that is randomly sampled from $(\mathbf{d}_{i,k}^p)_{i=1}^{I_k}$. The update procedure is an iterative process until a terminating condition, such as convergence or a predefined number of iteration times, is satisfied.

2. *Upload.* The updated model parameters \mathbf{w}_k or its gradients $\mathbf{g}_k = \mathbf{w}_k - \mathbf{w}_0$ are uploaded from each client to the server.

3. *Aggregation and 4. Broadcast.* The server aggregates the uploaded models from clients to update the global model \mathbf{w}_0 , as follows:

$$\mathbf{w}_0 = \sum_{k=1}^K \frac{I_k}{I} \mathbf{w}_k, \quad (3)$$

where $I = \sum_{k=1}^K I_k$. Subsequently, the server broadcasts the global model to all the clients via multicast channels. These procedures are iterated for a finite number of rounds.

2.2 Benchmark 2. Federated Distillation

Fig. 1b shows the process of FD [6], i.e., one of the FL algorithms with model output exchange, where clients share per-class logits instead of model parameters. In the FD, each client treats itself as a student, while aggregated logits act as teachers, where each local client model is trained using the aggregated logits. The specific procedure in FD consists of the following six steps: “1. Update,” “2. Prediction & Logit aggregation per label,” “3. Upload,” “4. Aggregation,” “5. Broadcast,” and “6. Distillation,” as shown in Fig. 1b. The differences between FD from the benchmark 1 are in the second and sixth steps. After the “1. Update” step, using the trained local ML model, every client calculates the local logit representing the inferred probability that each data sample is classified into each class. Subsequently, each local logit is aggregated to each client on a per-label basis in the “2. Prediction & Logit aggregation per label” step. The aggregated logits are uploaded to a remote server in the “3. Upload” step. The uploaded logits are aggregated in the server in the “4. Aggregation” step, whereas the aggregated logits are broadcast to the clients in the “5. Broadcast” step.

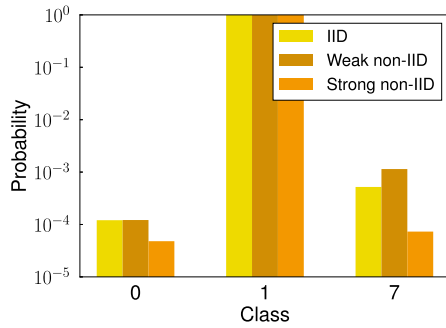


Fig. 2. A part of FD’s global logit after 16 rounds, using MNIST dataset, under IID, weak non-IID, and strong non-IID data distribution. That is, probabilities of class “0,” “1” and “7” of the logit for class “1.”

Finally, in the “6. Distillation” step, each client re-trains its local model using both the pre-attached labels and the broadcasted logits.

The detailed FD procedures are as follows. In the following, we consider that k th client’s private dataset can be divided into N_L set $(D_{k,n_l})_{n_l=1}^{N_L}$, where the sample pertaining to the class n_l is categorized to set D_{k,n_l} , and N_L is the number of objective classes.

2. *Prediction & Logit Aggregation Per Label.* Each client k predicts logit of its data input and calculate local-average logit t_{k,n_l} for each class, as follows:

$$t_{k,n_l} = \frac{1}{|D_{k,n_l}|} \sum_{\{d,t\} \in D_{k,n_l}} F(d | w_k). \quad (4)$$

If the client dose not have any sample pertaining to class n_l , $t_{k,n_l} = 0$.

3. *Upload*, 4. *Aggregate*, and 5. *Broadcast*. In the following “3. Upload” step, the local-average logit t_{k,n_l} are uploaded from each client to the server. In the “4. Aggregate” step, the uploaded logits are aggregated to create the global-average logit t_{g,n_l} per class as follows:

$$t_{g,n_l} = \frac{1}{|K_{n_l}|} \sum_{k \in K_{n_l}} t_{k,n_l}, \quad (5)$$

where K_{n_l} is the subset of clients having any sample pertaining to class n_l . Subsequently, the server broadcasts the global-average logit to all the clients via multicast channels in the “5. Broadcast” step.

6. *Distillation.* Each client k updates the model parameters using the pre-attached labels and the distillation logits: T_k and \hat{T}_k , where $\hat{T}_k = \{\hat{t}_{k,i}\}_{i=0}^L$. The distillation logit T_k is obtained using the broadcasted global-average logit and the local-average logit. In more detail, considering that the sample $d_{k,i}^p$ pertains to the class n_l , $\hat{t}_{k,i}$ is obtained as follows:

$$\hat{t}_{k,i} = \frac{1}{|K_{n_l}| - 1} (|K_{n_l}| t_{g,n_l} - t_{k,n_l}). \quad (6)$$

Using T_k and \hat{T}_k , the model is updated as follows:

$$w_k \leftarrow w_k - \eta \nabla \{ \phi(D_k^p, T_k | w_k) + \gamma \phi(D_k^p, \hat{T}_k | w_k) \}, \quad (7)$$

where γ is a weight parameter for the distillation regularizer. All the procedures, except for “1. Update,” are iterated for a finite number of rounds.

Negative Effect of Federated Distillation Under Non-IID Data.

The model performance trained by FD is much lower than FL under non-IID data. The reason is that the global logits of FD retain similar information to local labels already attached to each mobile device. The global logit is calculated as the average of the local per-label average logits among the clients. In the following, we explain the method to obtain global logit in FD more formally. We consider that k th client’s private dataset can be divided into N_L subset $(D_{k,n_l})_{n_l=1}^{N_L}$, according to the ground-truth labels, where N_L is the number of objective classes. The global logit for class n_l can be represented as follows:

$$t_{g,n_l} = \frac{1}{|K_{n_l}|} \sum_{k \in K_{n_l}} \frac{1}{|D_{k,n_l}|} \sum_{\{d,t\} \in D_{k,n_l}} F(d | w_k), \quad (8)$$

where K_{n_l} is the subset of clients having any class n_l sample, and w_k is the model weights of the k th client. Considering the strong non-IID data, where each client only has one or two class sample, $F(d | w_k)$ is almost the same as the one-hot label t , which is the ground-truth label already attached to d . This is because of the over-fitting of t , due to the use of few samples. Thus, t_{g,n_l} , which is the average of the almost one-hot logit $F(d | w)$, is also similar to the one-hot labels.

To support the statement, we analyzed FD’s global logits under three data distributions; IID, weak non-IID, and strong non-IID. The number of clients K was fixed to ten. The strong non-IID distribution used in our original manuscript implies that each client has a dataset consisting of two or three classes’ samples. The week non-IID distribution implies that the clients have the datasets consisting of ten classes, and the number of data samples on a few of the classes is much smaller than that of the other classes.

Fig. 2 shows a part of the FD’s global logit under three data distribution using MNIST, i.e., class probabilities of class “0,” “1,” and “7” of the global logit for class “1.” Obviously, the largest probability of the logit is class “1.” Under strong non-IID data, the probability of class “7” is as large as that of “0” and smaller than that under IID or weak non-IID. This implies that the global logit under strong non-IID is more similar to one-hot than that of IID or week non-IID. This may be the reason that the FD model performance is much lower than that of FL under strong non-IID data.

Noted that, under IID or weak non-IID data, the probability of class “7” is large than that of “0,” which results in the success of the FD. This difference between class “0” and class “7” indicates that the digit “1” image is more similar to the digit “7” than the digit “0,” which is essential to the success of the knowledge distillation as supported in [26]. Hence, we can conclude that the intensity of non-IID data distributions exactly affects the model performances.

2.3 Proposed Distillation-Based Semi-Supervised Federated Learning

2.3.1 Background and Overview of Distillation-Based Semi-Supervised Federated Learning

The proposed DS-FL is motivated by the lower performance of the model trained following the FD benchmark presented in the previous section. In FD, the global logits are used to distinguish the class to which each sample in the local

dataset belongs. However, as the dataset is already labeled, the “6. Distillation” step can result in a similar model to the one trained in the previous “1. Update” step (see Fig. 1b and compare “1. Update” and “6. Distillation”). Hence, the models trained in FD exhibit similar performance to the local model training, which is lower than the benchmark 1, i.e., FL, with model parameter exchange.

Hence, the fundamental idea behind the proposed DS-FL is to share the unlabeled dataset and use global logit to identify what class each sample in the unlabeled dataset pertains. This creates a new labeled dataset, based on which the local ML model is further trained, as shown in “6. Distillation” step in Fig. 1c. Due to this training procedure, the proposed DS-FL avoids the similarity in training between “1. Update” and “6. Distillation” steps can enhance the model performance benefitting from data augmentation effects. In other words, DS-FL uses unlabeled distillation, where a neural network model is trained using other models’ prediction of the unlabeled data. In the following, we detail the training procedure of the proposed DS-FL.

2.3.2 Detailed Procedure of Distillation-Based Semi-Supervised Federated Learning

The detailed procedure of DS-FL is depicted in Fig. 1c. In the following, we consider that each client $k = 1, 2, \dots, K$ does not only hold the labeled private dataset $(d_{i,k}^p, t_{i,k})_{i=1}^{I_k}$, but also the shared unlabeled dataset $(d_j^o)_{j=1}^{I^o}$, where $d_{i,k}^p$ and d_j^o denote the vectorized input samples in the labeled and unlabeled datasets, respectively. Moreover, I_k and I^o denote the number of samples in the labeled and unlabeled datasets, respectively. Additionally, $\mathbf{o}_r \subset \{1, 2, \dots, I^o\}$ represents index set of the unlabeled dataset, where r indicates round index. Moreover, I^{or} represents the size of \mathbf{o}_r . Considering N_L as the number of objective class, the term $t_{i,k} = [t_{i,k,1}, \dots, t_{i,k,N_L}]^T$ is the vectorized form of the label attached to the sample $d_{i,k}^p$ and is in the one-hot representation, wherein the element $t_{i,k,n}$ equals 1 if the n th label is the ground-truth and 0 otherwise. For shorthand notation, let $N_S \times I_k$ and $N_S \times I^o$ matrices D_k^p and D^o denote the concatenations of $(d_{i,k}^p)_{i=1}^{I_k}$ and $(d_j^o)_{j=1}^{I^o}$, respectively, and let $N_L \times I_k$ matrix T_k denote the concatenation of $(t_{i,k})_{i=1}^{I_k}$, where N_S denote the dimension of the input samples. Additionally, let $N_S \times I^{or}$ matrix D^{or} denotes the subset of the unlabeled dataset $(d_j^o)_{j \in \mathbf{o}_r}$. The index set \mathbf{o}_r is determined randomly by the server and shared among the clients before “2. Prediction” step.

1. *Update*. In the “1. Update” step, each client updates its model with its private dataset, as shown in (1).

2. *Prediction*. Based on the model learned in the previous step, each client predicts the local logit, i.e., the labels for data samples in a shared unlabeled dataset. More specifically, given the model parameter w_k and \mathbf{o}_r , each client predicts local logits $\hat{t}_{j,k}$ for $j \in \mathbf{o}_r$ as follows:

$$\hat{t}_{j,k} = F(d_j^o | w_k). \quad (9)$$

For shorthand notation, the $N_L \times I^{or}$ matrix \hat{T}_k denotes the concatenation of $(\hat{t}_{j,k})_{j \in \mathbf{o}_r}$.

3. *Upload*. The local logits \hat{T}_k are uploaded from each client to the server, differing from FL with model parameter exchange that uploads the model parameters w_k .

4. *Aggregation and 5. Broadcast*. The server aggregates the logits from clients to create global logits \hat{T} . The procedure for aggregating uploaded logits is described in Section 3. Subsequently, the server broadcasts global logits to all the clients via multicast channels.

6. *Distillation*. The clients update their local model based on the broadcasted global logits \hat{T} and shared unlabeled dataset D^{or} . More concretely, the model parameters are updated as follows:

$$w_k \leftarrow w_k - \eta_{\text{dist}} \nabla \phi(D^{or}, \hat{T} | w_k), \quad (10)$$

where η_{dist} is the learning rate in the proposed distillation procedure. In addition to the clients’ local models, the server has a global model w_g . The server updates the global model based on the broadcasted global logits \hat{T} and shared unlabeled dataset D^{or} . More concretely, the model parameters are updated as follows:

$$w_g \leftarrow w_g - \eta_{\text{dist}} \nabla \phi(D^{or}, \hat{T} | w_g). \quad (11)$$

In Section 4, the global model is used for evaluating the performance of the DS-FL framework.

These procedures are iterated for a finite number of rounds. The overall procedures are summarized in Algorithm 1.

Algorithm 1. DS-FL

0. **Initialization:**

Initialize all the client models w_k and the global model w_g
Distribute the open data D^o to all clients

1. **Update:**

for Each client k in parallel do

Update the local model parameter w_k via (1)

end for

2. **Prediction:**

All the clients share the index set \mathbf{o}_r

for Each client k in parallel do

Calculate local logits \hat{T}_k via (9)

end for

3. **Upload:**

Each client uploads the local logits \hat{T}_k

4. **Aggregation:**

Server aggregates the logits to create the global logit \hat{T}
according to (13) in ERA (proposed)
or (16) in SA (baseline)

5. **Broadcast:**

Broadcast \hat{T} to all clients

6. **Distillation:**

for Each client k in parallel do

Update the local model parameter w_k via (10)

end for

Steps 1–6 are iterated for multiple rounds

3 ENTROPY REDUCTION AGGREGATION

This section presents the proposed logit aggregation method, i.e., ERA, which intentionally reduces the entropy of global logits. We define the entropy of a logit t as follows:

$$f_e(t) = - \sum_{n=1}^{N_L} t_n \log t_n. \quad (12)$$

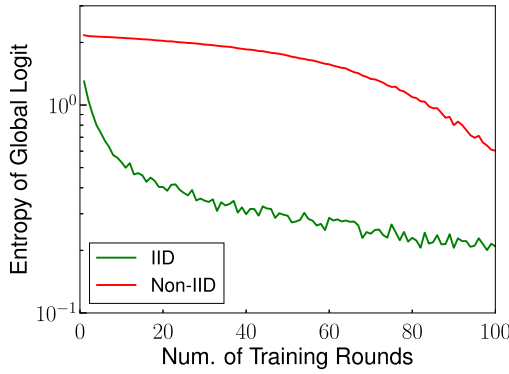


Fig. 3. Entropy of the global logit in SA method versus training rounds. For IID and non-IID data, using MNIST dataset.

First, we detail our motivation to propose ERA by highlighting that the simple baseline-aggregation-method, involving averaging only the local logits, results in a higher entropy of global logits in the heterogeneity of data distributions among clients than those without such heterogeneity. Subsequently, we detail how to reduce entropy in ERA.

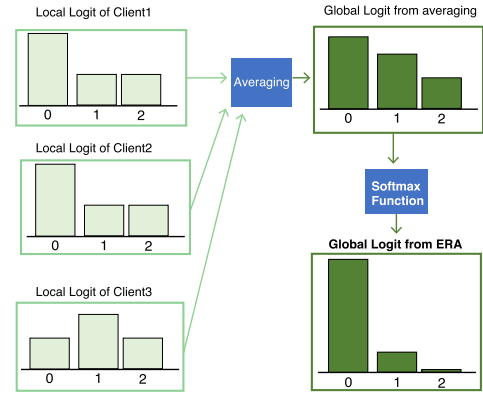
3.1 Motivation for Entropy Reduction Aggregation

The motivation for reducing the entropy of global logits is to accelerate and stabilize DS-FL, particularly in non-IID data distributions. In the collaborative learning with non-IID data, the entropy of global logits is much larger than appropriate ones. Fig. 3 shows the comparison of global logits yielded via the simple aggregation method involving the averaging of the uploaded local logits for IID and non-IID data distributions. Under non-IID data, the entropy in the early stage of training is higher than 2.0, which is approximately the upper limit of the entropy in the ten-class classification problem. These maximum values of the entropy are meaningless because they do not identify to what class each input sample pertains. Hence, it is difficult to train using such inappropriately high entropy logit in the simple aggregation method, and hence, the reduction of the entropy is required for training success.

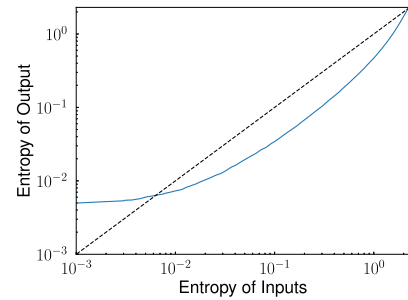
Another favorable consequence of reducing the entropy of global logits is to enhance the robustness against various attacks corrupting local logits and noising open data. In DS-FL, a malicious client can upload local logits that do not enhance or even harm model performance, which can occur by, for example, updating the local models over the dataset that is not labeled properly. In addition to such noisy label, open data can be noised; for example, the inadequate data is added to the open data. In these cases, similarly to under non-IID data distributions, the entropy of global logits yielded from the simple aggregation method averaging local logits becomes higher, leading to poorer model performances. This fact is verified in Section 4. Hence, reducing the entropy of the global logit is expected to enhance the robustness against such attacks.

3.2 Procedure for Reducing Entropy

The proposed ERA reduces the entropy of the global logits yielded from averaging uploaded local logits via the procedure depicted in Fig. 4a. To reduce the entropy of the global logit, we use the softmax function as an example.



(a) Procedure of proposed ERA.



(b) Entropy comparison between the input and output of the softmax function.

Fig. 4. Illustrative example of proposed ERA at $T = 0.1$ on three-class classification task.

Let $\hat{T}^{(\text{ERA})}$ denote the global logit yielded from ERA, which is an $N_L \times T^{\text{or}}$ matrix. Given the temperature of the softmax function T , the global logit generated by ERA is described as follows:

$$\hat{T}^{(\text{ERA})} = F_s \left(\frac{1}{K} \sum_{k=1}^K \hat{T}_k | T \right), \quad (13)$$

where $F_s(\cdot | T)$ denotes the softmax function with respect to temperature T . The softmax function $f_s(t | T) : \mathbb{R}^{N_L} \rightarrow \mathbb{R}^{N_L}$ is denoted as follows:

$$f_s(t | T) = \frac{1}{\sum_{n=1}^{N_L} e^{\frac{t_n}{T}}} e^{\frac{t}{T}}. \quad (14)$$

Moreover, $F_s(T | T)$ is denoted as follows:

$$F_s(T | T) = \{f_s(t_1 | T), \dots, f_s(t_{I_r} | T)\}. \quad (15)$$

The higher the temperature, the higher the entropy of the output of the softmax function, and viceversa.

For example, ERA sets lower temperature of $T = 0.1$, while original knowledge distillation (KD) [26] sets a higher temperature of $T = 20$. Hinton *et al.* [26] state that in KD, the scores in the logits can be interpreted as an inherent similarity between the corresponding label and the input samples. For example, one handwritten digit labeled as “7” might provide a score of 0.1 to the label “1” and that of 0.01 to the label “4.” Given that the physical meaning of this score is the probability that each sample pertains to the corresponding label, these scores can be interpreted as how much the given handwritten

digit is prone to be “1” and “4.” As the above bias of this likelihood (0.1 to “1” and 0.01 to “4”) may be yielded due to the similarity between “7” and “1,” the given logit scores are interpreted as the similarity between the corresponding label and the input samples. To transfer a similar structure, we trained the student model using the logit create by the softmax function using a high temperature; thus, emphasizing the non-highest scores. In contrast, particularly under non-IID data, the global logits of DS-FL is much more ambiguous than the predictions of a well-trained teacher model such as the model used in KD [26]. Moreover, the ambiguity may conceal the true classes each sample pertains, wherein the lower temperature is expected to be useful to balance the class information and a similar structure. As shown in Fig. 4b, at the temperature of $T = 0.1$, the entropy values outputted by the softmax function are generally lower than the input. Hence, as shown in Fig. 4a, the global logit yielded from ERA is sharper than that yielded via averaging the uploaded local logits. Noted that T of ERA discussed here differs from the temperature of the softmax function that activates the fully-connected layer on the output side of neural networks. The former temperature is specific to the training procedure and is set to a certain value lower than 1.0 (e.g., 0.1 in the experimental evaluation), whereas the latter temperature is set to 1.0 in the training and inference steps.

The ERA is compared with the baseline of only averaging uploaded local logit, which is named simple aggregation (SA). The resultant global logit yielded from SA $\hat{T}^{(SA)}$ is given as follows.

$$\hat{T}^{(SA)} = \frac{1}{K} \sum_{k=1}^K \hat{T}_k. \quad (16)$$

4 EXPERIMENTAL EVALUATION

4.1 Setup

Datasets. Four datasets were used for evaluation purposes, including image classification and text classification. For image classification, two major tasks, MNIST [33] and Fashion-MNIST [34], were used. MNIST [33] is a widely-used object classification dataset consisting of 60,000 training images and 10,000 testing images with 10 image classes. Fashion-MNIST [34] comprises 60,000 training images and 10,000 testing images of 10 different fashion products such as coats and sneakers. These datasets have been used in several machine learning studies. In addition to the image classification tasks, two major text classification tasks, Internet movie database (IMDb) review-sentiment and Reuters datasets, are used for evaluation purposes. The IMDb dataset was created in [35], which consists of 50,000 textual reviews of movies and divided into 25,000 training dataset and 25,000 testing dataset. The reviews are categorized into negative and positive sentiment class. The Reuters dataset¹ consists of 11,228 headline articles, divided into training and testing dataset in a ratio of 8:2. The headlines are categorized into 46 classes of topics such as “earn” and “trade.”

1. The Reuters dataset we used was Keras [36] revised subset of Reuters-21578 corpus [37], [37] is freely available for experimentation purposes from <http://www.daviddlewis.com/resources/testcollections/~reuters21578/>.

Pre-Processing of the Sentences. We used different neural network architecture and different preprocessing method for IMDb and Reuters datasets. For the IMDb dataset, we only considered the top 20k words and the first 200 words of each movie review, following Keras [36] tutorial. According to the frequency appearance, each word is converted to an integer. Moreover, following the word order in the sentences, each sentence is converted to a sequence of integers. For Reuters dataset, we only considered the top 10k words and each word was converted to a integer, as IMDb dataset. In the sentences categorization task, such as Reuters dataset, the type of words are more useful than the word order. Thus, we employed the Bag-of-Words method, which is often used as a preprocessing for the Reuters dataset [38], i.e., the headline is converted to a binary vector, indicating the sentence composition of words.

Data Partitions. The data distribution over the clients was determined based on [4]. For the image classification tasks, we fixed the number of clients K to 100. Subsequently, the dataset was divided into the unlabeled open dataset and the labeled private dataset. Let denote that the open dataset consists of I^o images, and the private dataset consists of I^p image-label pairs, where $I^o + I^p \leq 60,000$. This study considered two ways of partitioning the private dataset over the clients: IID datasets and non-IID datasets. The private dataset was shuffled and partitioned into K portions for the clients to obtain the IID datasets. Thus, each client had I^p/K pairs of images and labels. To generate non-IID datasets, we sorted the private dataset by its classification label and divided it into $2K$ shards of size $I^p/2K$, among which two shards are assigned to each client. In this study, the non-IID data distribution among the clients followed the pioneer study [4], which considered more severe non-IID data than that in [6]. Hence, the differences in the data distribution are the reason for the difference in the test accuracy of the FD between this study and [6]. In the evaluation, we did not adapt any data augmentation, such as rotation or flipping of an image.

For the text classification task, we fixed the number of clients K to 10. Subsequently, the dataset was divided into the unlabeled open dataset and the labeled private dataset. For IMDb dataset, (I^o, I^p) was (10,000, 15,000) and for Reuters dataset, (I^o, I^p) was (3,982, 5,000). To generate the non-IID partitioned datasets for IMDb, we divided the dataset so that, for all clients, the ratio of the number of positive labeled sentences to that of negative was 9:1 or 1:9. In consequence, some clients had 150 positive labeled sentences and 1,350 negative labeled sentences, and the other had 1,350 positive labeled sentences and 150 negative labeled sentences. To generate non-IID partitioned datasets for Reuters, we sorted the private dataset by its classification label and divided it into K shards of size I^p/K , among which one shard was assigned to each client.

Our evaluation aimed to evaluate FL frameworks under the same clients’ privacy level. Thus, we avoided to compare with other baselines sharing any clients’ labeled data, such as FD+FAug [6].

ML Model. For the image classification tasks, we examined two ML models designed for either MNIST or Fashion-MNIST dataset. Specifically, the model for MNIST was a convolutional neural network model that consisted of two

5×5 convolution layers (32 and 64 output channels, each of which was activated by batch normalization and ReLU, followed by 2×2 max pooling) and two fully-connected layers (512 units with ReLU activation and another 10 units activated by softmax). For Fashion-MNIST, the model consisted of six 3×3 convolution layers (32, 32, 64, 64, 128, and 128 channels, each of them activated by ReLU and batch normalized. Every two of them followed by 2×2 max pooling) and by three entirely connected layers (382 and 192 units with ReLU activation and another 10 units activated by softmax). For the text classification tasks, we examined two ML models designed for either IMDB or Reuters dataset. The IMDB dataset is semantic classification, where the word order in the sentences is useful. Thus, for the IMDB dataset, we employed long short-term memory (LSTM), which can learn the time-dependent relationships among inputs and outputs. Specifically, the model for IMDB consisted of a simple LSTM model, which followed Keras [36] tutorial, consisting of an embedding layer (output dimension of each word was 32), a LSTM layer (32 nodes), and a fully-connected layer (2 unit was activated by softmax). For Reuters dataset, we employed a simple multi-layer perceptron (text-DNN). Specifically, the model was a three layer perceptron (512 and 128 units activated by ReLU and batch normalized and another 46 units activated by softmax). The ML models resulted in 583,242 model parameters (2.3 megabytes in a 32-bit float) for the MNIST dataset, 2,760,228 model parameters (11.2 megabytes in a 32-bit float) for the Fashion-MNIST dataset, 646,338 model parameters (2.6 megabytes in a 32-bit float) for the IMDB dataset, and 5,194,670 (20.8 megabytes in a 32-bit float) model parameters for Reuters dataset.

Training Hyperparameters. When the models were updated and distilled, the optimizer, mini-batch size, the number of epochs in each round, and training rate were selected as stochastic gradient descent, 100, 5, and 0.1, respectively. For text classification tasks, they are selected as Adam, 128, 5, and 0.001, respectively. The temperature of the softmax function T in the DS-FL with ERA was set to 0.1. As shown in Section 3.2, the temperature of the softmax function that activates the fully-connected layer on the output side of neural networks is set to 1.0 in the training and inference steps. The amount of unlabeled data used in each round, i.e., the size of \mathbf{o}_r , was 1,000.

Attack Settings. To evaluate the attack robustness of ERA, we considered attacks where malicious clients corrupted local logits. Specifically, we considered noisy labels, noisy data, and model poisoning attacks. The robustness evaluations of this section used image classification task described above.

Noisy Labels. First, in the noisy label attack, a particular client's images pertaining to a certain class were labeled as another class to corrupt the local logit. We assumed that all clients had the same degree of noisy labeled datasets, i.e., mistakenly labeled private data. This was regarded as a situation where all clients can be considered as attackers. Thus, we evaluated DS-FL and FL in a worst-case scenario, which we believed was sufficiently worthwhile to understand the attack robustness in practical and severe situations. More precisely, consider a number of noising class C , where each client independently selects C classes as source classes $S = \{S_1, \dots, S_C\}$ and another C classes as false

classes $F = \{F_1, \dots, F_C\}$. Subsequently, all the images pertaining to the source class $S_c \in S$ are mistakenly classified to the corresponding false class $F_c \in F$. Assuming 10 objective classes, $C I^p/10$ images of the private dataset were mistakenly labeled. In this evaluation, we used MNIST dataset, considering IID data distribution.

Noisy Data. Second, we evaluated the robustness of ERA in a noisy data attack, where a malicious client adds noisy semantic data into the open dataset. In more detail, consider training a handwritten digit classifier, where the private datasets and the test dataset, testing our methods, are the MNIST dataset. In this experiment, we assumed non-IID distribution and added I^n Fashion-MNIST images to I^o MNIST open dataset, i.e., $I^o + I^n$ images were used for the unlabeled open dataset. We fixed I^o to 20,000 and experimented with I^n .

Model Poisoning. To evaluate robustness against model poisoning attack, we conducted the experiment with multiple malicious clients, where the number of malicious clients is denoted as m . In the evaluation, we fixed the total number of the clients K to 100. In this analysis, we assumed that the malicious clients performed a model poisoning attack [39], which aimed to replace the global model with an arbitrarily model and introduced a backdoor to the global model. In this attack, the malicious clients aimed to replace the global model w_g with a malicious model w_x performing a malicious client selected backdoor task, as in (17).

$$w_x = \frac{1}{K} \sum_{k=1}^K w_k. \quad (17)$$

As the training process progress, all the clients' model w_k converge to w_g . Then, we obtain

$$w_x = \frac{K-m}{K} w_g + \frac{m}{K} w_M, \quad (18)$$

where w_M is the model parameter uploaded by the malicious clients. Therefore, the malicious clients upload w_M to replace the global model as follow:

$$w_M = \frac{1}{m} \{K w_x - (K-m) w_g\}. \quad (19)$$

A single shot attack of the malicious clients replaces the global model with the malicious model, and the backdoor survives for long rounds without any attacks. We simply extend this attack to DS-FL; the malicious clients send the logit made by w_x , while never updating the model w_x . The model poisoning attack was designed to attack the FL and not the DS-FL. Thus, this evaluation could not be fair for FL and DS-FL; however, the evaluation reveals the toleration of DS-FL to one of the most powerful attacks designed for FL.

We assumed that the main task was the MNIST task and the backdoor task was the Fashion-MNIST task. The data distribution over benignant clients was assumed IID. The malicious clients' intention was to classify the images of handwritten digits and fashion products of the global model (e.g., classify images of digit "0" and "T-shirt" to class "0" and images of digit "3" and "Dress" to class "3"). The malicious clients included the model trained using the entire MNIST training dataset and Fashion-MNIST training

TABLE 1
Comparison of Communication Cost Per Round
in the Image Classification Tasks

Method	MNIST (smaller model)	Fashion-MNIST (larger model)
Benchmark 1: FL	236.1 MB	1.1 GB
Benchmark 2: FD	40.4 kB	40.4 kB
Proposed: DS-FL	4.0 MB	4.0 MB

dataset (containing 120,000 images and corresponding labels). The malicious clients performed the model poisoning attack once every five rounds.

4.2 Results

Communication Cost Per Round. The communication cost per round with FL, FD, and DS-FL are calculated and listed in Tables 1 and 2 for image classification and text classification tasks, respectively. As seen in the tables, the communication costs of the proposed DS-FL and the FD benchmark is smaller than that of the FL benchmark and do not depend on the model sizes. The reason for this result is that the payload size of the logits uploaded in the DS-FL and the FD benchmark is smaller than that in the ML model parameter used in the FL benchmark and does not depend on the number of model parameters. Furthermore, the communication cost of the FD benchmark is 100 times smaller than that of DS-FL. The reason is that the number of logits uploaded by a DS-FL client is more than that of the FD. In the FD benchmark, the clients upload local logits on a per-classy basis, while in the DS-FL, they upload local logits on a per-sample basis in the unlabeled dataset. However, the proposed DS-FL exhibits a higher classification accuracy than that of the FD benchmark, which is verified in the following results.

Although the evaluations are conducted without any missing clients per round assuming stable and sufficient communication qualities, the results further implies that DS-FL is more robust to limited communication qualities causing such missing clients. This is because of the smaller payload size of DS-FL than FL. More concretely, the smaller payload size in DS-FL allows more clients to complete the uploading of the training results, i.e., model parameters and logits in FL and DS-FL, respectively, even under the limited communication quality. Hence, even if some clients are possibly missing for the communication round, the number of the missing clients in DS-FL is smaller than that in FL. In this sense, we believe that these results sufficiently contribute to solving the FL problem of limited communications in view of the case for the missing clients.

Accuracy Improvement Per Communication Cost in Training. Figs. 5a and 5b show the accuracy as a function of the cumulative communication cost for MNIST and Fashion-MNIST, respectively, with $\{I^p, I^o\} = \{20000, 20000\}$ and non-IID datasets. The cumulative communication cost of DS-FL includes an initial cost to distribute the unlabeled data to the clients in addition to the per round cost, while that of other baselines did not include the initial cost. The initial cost is described as ComU@I in Table 3. In both the FL and DS-FL, as the training processes progress, i.e., the cumulative communication costs to share models or logits increase,

TABLE 2
Comparison of Communication Cost Per Round
for Text Classification Tasks

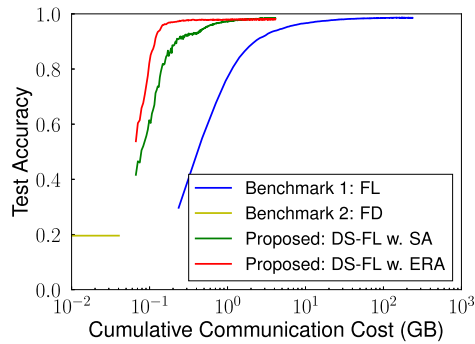
Method	IMDb (LSTM)	Reuters (text-DNN)
Benchmark 1: FL	28.6 MB	228.8 MB
Benchmark 2: FD	176 B	93 kB
Proposed: DS-FL	88 kB	2.0 MB

and the accuracy improves. DS-FL performance is evaluated using the global model, which is trained on the server as described in Section 2.3.2. Meanwhile, the FD accuracy remains approximately 20 percent, which is almost similar to that of a single client. In both tasks: MNIST and Fashion-MNIST, the proposed DS-FL outperforms the FL (benchmark 1) in terms of cumulative communication cost while achieving comparable accuracy. As shown in Table 1, the results could be due to the communication cost per round of the DS-FL that is lower than that of FL. If the aggregation methods in DS-FL are compared, the proposed ERA obtains almost the same accuracy as that of the SA baseline, while the cumulative communication cost to the convergence of ERA is smaller than that of SA due to the acceleration effect of ERA. Based on these results, we can conclude that the DS-FL with the proposed ERA reduces the communication costs substantially while achieving similar performance to the FL benchmark, i.e., ERA accelerates the convergence speed.

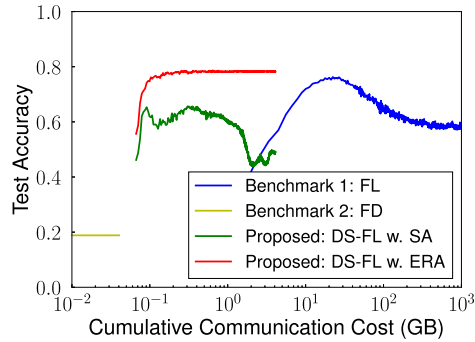
Figs. 5c and 5d show the accuracy as a function of the cumulative communication cost for IMDb and Reuters. The results show a similar trend to computer vision tasks, i.e., MNIST and Fashion-MNIST tasks. In both the FL and DS-FL, as the training processes progress, i.e., the cumulative communication costs to share models or logits increases, and the accuracy improves. In both tasks, IMDb and Reuters, the proposed DS-FL outperforms the FL (benchmark 1) in terms of cumulative communication cost while achieving comparable accuracy. In contrast, the accuracy of FD is much lower than that of FL, i.e., 23.3 and 39.0 percent lower for IMDb and Reuters, respectively. If the aggregation methods in DS-FL are compared, the proposed ERA obtains almost the same accuracy as that of the SA baseline, while the cumulative communication cost to the convergence of ERA is smaller than that of SA, due to the acceleration effect of ERA.

Moreover, these results are also verified in Table 3. The table lists the cumulative communication costs required to achieve a test classification accuracy of $x\%$, and the highest testing accuracy among the training process, referred to as ComU@ $x\%$, and Top-Accuracy, respectively. In Table 3, the DS-FL with ERA achieves lower ComU@ $x\%$ than that using DS-FL with SA and FL for all cases. For example, regarding Fashion-MNIST, DS-FL with ERA achieves 99.0 percent lower ComU@65% and 99.4 percent lower ComU@75% than FL. Moreover, regarding Reuters, DS-FL with ERA achieves 99.4 percent lower ComU@65% and higher Top-Accuracy than FL.

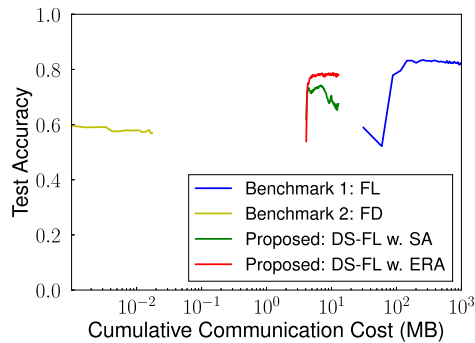
Test Accuracy Comparison. Table 3 lists the highest accuracy for all the training rounds denoted as Top-Accuracy.



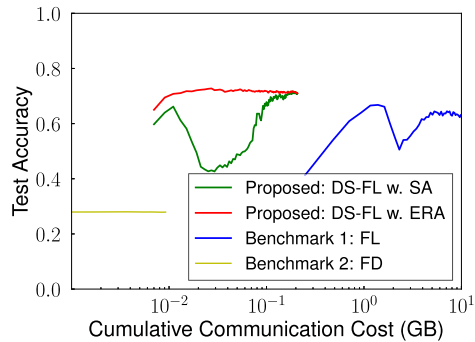
(a) MNIST



(b) Fashion-MNIST



(c) IMDb



(d) Reuters

Fig. 5. Test accuracy versus cumulative communication costs under non-IID data. For MNIST and Fashion-MNIST, the numbers of samples in the local and open unlabeled dataset are 20,000 and 20,000, respectively.

The table indicates that the DS-FL with ERA achieves similar or superior Top-Accuracy compared to that of FL and DS-FL with SA. Comparing DS-FL with ERA to the FL

TABLE 3
Comparison of Communication Cost and Top-Accuracy With the Size of the Open Dataset I^o

(a) MNIST

Method	I^o	ComU@I (GB)	ComU@95% (GB)	ComU@97% (GB)	Top-Acc (%)
Single Client	-	-	-	-	19.4
FL	-	-	5.67	12.5	98.7
FD	-	-	-	-	19.6
w.SA w.ERA	5,000	0.016	0.44 0.09	0.8 -	97.6 96.9
w.SA w.ERA	10,000	0.031	0.46 0.11	0.84 0.18	98.1 97.5
w.SA w.ERA	20,000	0.063	0.52 0.14	0.87 0.2	98.6 98.1
w.SA w.ERA	40,000	0.13	0.59 0.2	0.93 0.25	98.7 98.5

(b) Fashion-MNIST

Method	I^o	ComU@I (GB)	ComU@65% (GB)	ComU@75% (GB)	Top-Acc (%)
Single Client	-	-	-	-	18.6
FL	-	-	6.71	15.6	76.3
FD	-	-	-	-	18.9
w.SA w.ERA	5,000	0.016	0.07 0.03	- 0.05	73.5 77.5
w.SA w.ERA	10,000	0.031	0.22 0.04	- 0.06	68.7 77.1
w.SA w.ERA	20,000	0.063	0.09 0.07	- 0.10	65.6 78.7
w.SA w.ERA	40,000	0.13	0.33 0.14	- 0.17	65.3 79.0

(c) IMDb

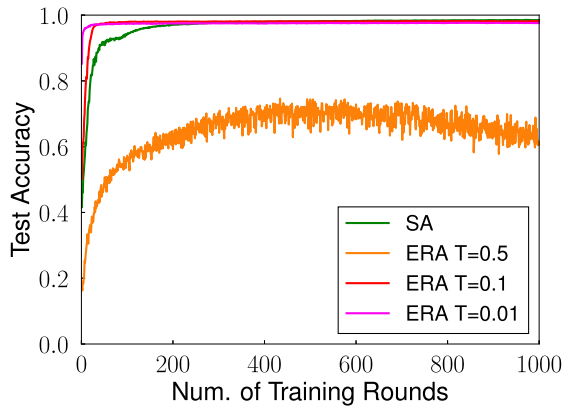
Method	I^o	ComU@I (MB)	ComU@70% (MB)	ComU@78% (MB)	Top-Acc (%)
Single Client	-	-	-	-	50.0
FL	-	-	87.9	116.3	83.4
FD	-	-	-	-	60.1
w.SA w.ERA	10,000	4.0	4.3 4.2	- 5.1	74.2 78.7

(d) Reuters

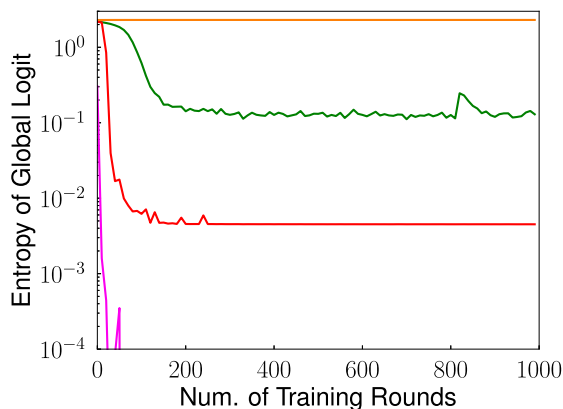
Method	I^o	ComU@I (MB)	ComU@65% (MB)	ComU@70% (MB)	Top-Acc (%)
Single Client	-	-	-	-	16.2
FL	-	-	1163.4	-	66.8
FD	-	-	-	-	28.0
w.SA w.ERA	5,000	6.3	11.0 7.0	124.4 11.0	71.7 72.8

ComU@I indicates the initial communication cost to distribute open dataset to all the clients for DS-FL. ComU@ x indicates the cumulative communication cost required to achieve a testing classification accuracy of x . Top-Accuracy indicates the highest testing accuracy among the training process. Single client indicates the accuracy without any collaboration between the clients.

benchmark in the MNIST task, the Top-Accuracy of DS-FL with ERA reaches up to 98.5 percent when $I^o = 40,000$, which is only 0.2 percent lower than that of FL. In the



(a) Accuracy



(b) Entropy of Global logit

Fig. 6. Test accuracy and entropy of global logit versus training rounds with temperature of ERA, using non-IID and MNIST dataset.

Fashion-MNIST and Reuters tasks, the Top-Accuracy of the DS-FL with ERA is higher than that of FL for all cases. In the IMDb task, the Top-Accuracy of the DS-FL with ERA is 4.7 percent lower than that of FL, whereas it is 18.6 percent higher than that of FD. Hence, we can again conclude that DS-FL with ERA achieves similar test performance to the FL benchmark while drastically reducing the communication costs. Comparing the Top-Accuracy of the proposed ERA and SA baseline, SA achieves higher Top-Accuracy relative to ERA for the MNIST task even though the difference becomes smaller as the number of samples in the unlabeled open dataset increases. Meanwhile, regarding the Fashion-MNIST task, the proposed ERA achieves higher Top-Accuracy than SA, where the difference ranges from 4.0 to 13.7 percent. Hence, recall that Fashion-MNIST is a more complicated task than MNIST [34]. In the IMDb and Reuters tasks, the proposed ERA achieves higher accuracy than SA. These results provide insight into the importance of reducing the entropy global logits, particularly in more complicated tasks, to enhance the DS-FL model performance.

Both FD+FAug [6], which is an advanced method of FD, and DS-FL with ERA deal with non-IID data distribution. Moreover, and FD+FAug outperforms FD. However, FD+FAug requires clients to upload a part of their labeled data, which does not satisfy the intentions of comparing FL

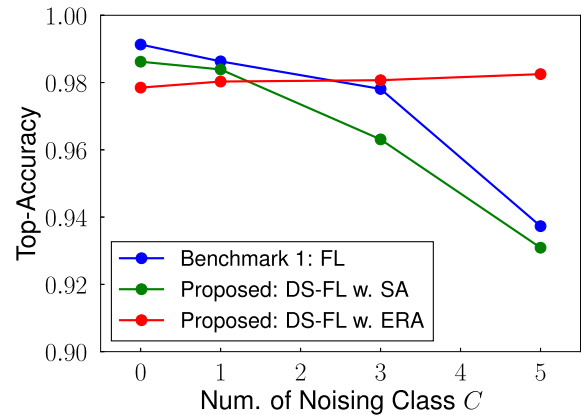


Fig. 7. Impact of noisy labels. Top-Accuracy as a function of the number of noising classes C . We use MNIST dataset with $I^p : I^o = 40,000 : 20,000$. The data distribution over client was IID.

frameworks under the same clients' privacy level. Thus, comparing DS-FL to other approaches that let clients share their raw data, such as FD+FAug and [16] is beyond the scope of this study.

Effect of Temperature T on Entropy Reduction Aggregation. To evaluate the effect of the temperature on the ERA, we evaluated the performance of ERA according to various T using the MNIST dataset considering non-IID data. Figs. 6a and 6b show the test accuracy and entropy of global logit as functions of the training round. When $T = 0.5$, the entropy is larger than that of SA, and the training is slower than when considering SA and ERA with smaller T . For $T = 0.1$ and 0.01 , the entropy is lower than that of SA, and the training is faster than that when considering SA. Thus, we note that ERA with a low T accelerates training.

Attack Robustness of Entropy Reduction Aggregation for Noisy Labels. Fig. 7 shows the Top-Accuracy for MNIST as a function of the number of noising classes C . All the clients hold noisy labeled and properly labeled data samples with a ratio $C : 10 - C$. Regarding the DS-FL with SA and FL, Fig. 7 shows that as the noised classes increase, the Top-Accuracy decreases. However, the DS-FL with ERA maintains the Top-Accuracy, when the noised-classes increase. This indicates that the DS-FL with ERA is more robust to IID noising than the FL. The following section presents an analysis of the global logit entropy to explain the robustness.

Attack Robustness of Entropy Reduction Aggregation for Noisy Open Dataset. Fig. 8 shows the Top-Accuracy for MNIST as a function of the number of noised samples in the open unlabeled dataset. First, the FL is unaffected by the noisy open data because FL does not use open data. Overall, as the number of noisy datasets in the open dataset I^n increases, the Top-Accuracy of DS-FL decreases. From the perspective of decreasing the Top-Accuracy with a particular number of noisy open datasets from that with $I^n = 0$ (i.e., the open dataset includes any noisy image) comparing the proposed ERA and the SA baseline, the decrease in Top-Accuracy of ERA is smaller than that of SA. Hence, we can conclude that the proposed ERA is more robust against a noisy open dataset than the SA baseline. This result is because the proposed ERA alleviates the increase in the entropy of global logits due to the noisy unlabeled data relative to the SA baseline, as shown in the following section.

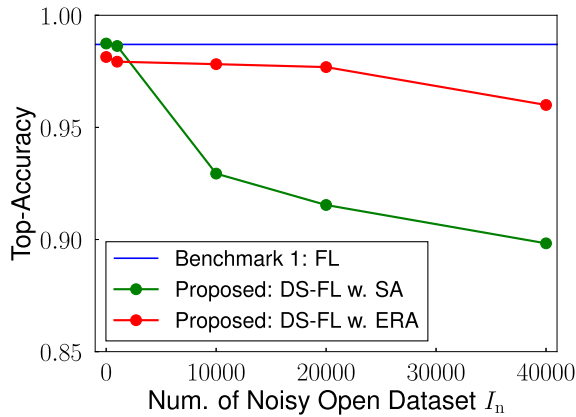


Fig. 8. Impact of noisy open dataset. Top-Accuracy as a function of the number of noisy open datasets I_n . Using MNIST dataset as clean open dataset and Fashion-MNIST dataset as noisy open dataset. The size of the clean open dataset was fixed to 20,000. The data distribution over client was non-IID.

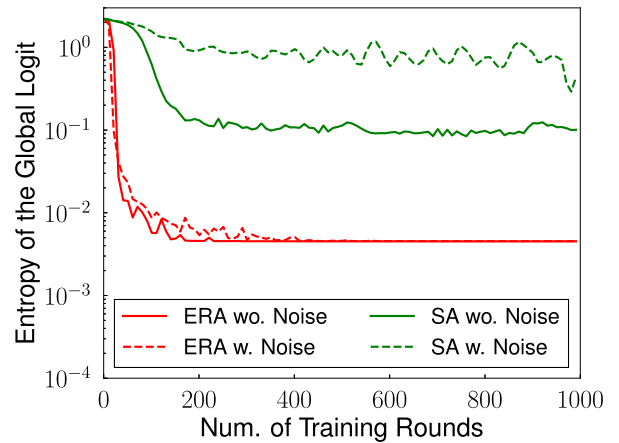
Entropy Analysis Under Noisy Data Attack. To explain the difference between the proposed ERA and SA baseline, we show in Fig. 9 the entropy of the global logits when the datasets include or not noises. When the dataset includes noises, both SA and ERA entropies become larger relative to that without noises. Meanwhile, the difference in ERA is smaller than that of SA. In the SA baseline, the high entropy target vectors are used to train each client’s model, making the SA process vulnerable to the noisy open dataset. Simultaneously, the proposed ERA alleviates the increase in the entropy of global logits, leading to high model performance.

Attack Robustness of Distillation-Based Semi-Supervised Federated Learning for Model Poisoning. In Table 4, the malicious clients achieved their objective in FL, while the attack failed in DS-FL with SA and ERA. Note that the objective of the malicious clients was to replace the global model with the model achieving high-test accuracy on both the main and the backdoor tasks. Table 4 shows the test accuracy of the global model after 100 rounds for the main (MNIST) and backdoor tasks (Fashion-MNIST) by the DS-FL and FL, for the number of the malicious clients of 1, 10 and, 50. In FL, for every case, the global model achieved high-test accuracy on both the main and the backdoor tasks. This result implies

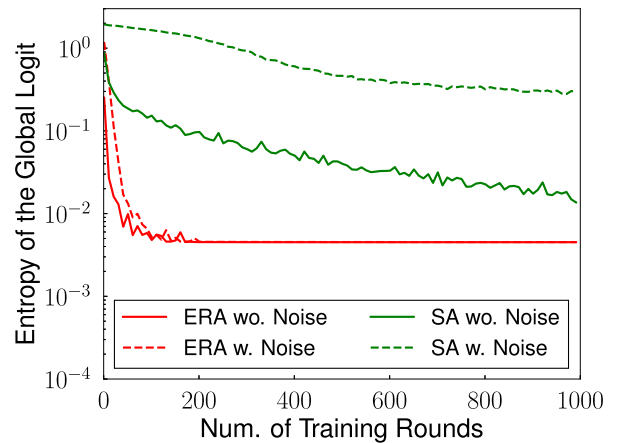
TABLE 4
Impact of Model Poisoning Attack

Num. malicious clients	Method	Accuracy of main task%	Accuracy of backdoor task%
1	FL	98.9	90.4
	DS-FL w. SA	97.5	9.6
	DS-FL w. ERA	97.9	8.7
10	FL	98.9	90.4
	DS-FL w. SA	98.5	9.9
	DS-FL w. ERA	98.1	9.1
50	FL	98.9	90.6
	DS-FL w. SA	98.8	7.9
	DS-FL w. ERA	98.7	10.0

Test accuracy implies the classification accuracy after 100 rounds. The total number of malicious clients and benignant clients is 100.



(a) Entropy of the global logits vs training round with or without noises in the open dataset. The open dataset consists of 20,000 benignant images or 20,000 benignant images and 40,000 noisy images.



(b) Entropy of the global logits vs training round with or without noisy label in clients dataset. Each client hold 200 noisy labeled dataset and 200 properly labeled dataset.

Fig. 9. Entropy of the global logits with or without noises dataset.

that the objective of the malicious clients was achieved. In contrast, for every case, in DS-FL with SA and with ERA, the test accuracy of the global model on the main task was as high as that in FL, while the accuracy on the backdoor task was much lower than that in FL. This result implies the failure of the attack. The reason is that DS-FL asks clients to transmit only logit but not ML model parameters, which prevents the malicious clients from the attack that corrupts the uploaded model parameters.

5 CONCLUSION

We proposed a cooperative learning method, named DS-FL, designed to be scalable according to model sizes in terms of communication efficiency while achieving similar accuracy to benchmarks FL algorithms. The fundamental idea of the proposed DS-FL was the model output exchange for an unlabeled open dataset. Additionally, we proposed a logit aggregation method for the DS-FL, which aimed to accelerate the training process and enhance robustness under the non-IID data. The simulations showed that the proposed DS-FL method outperformed the benchmark method FL in

terms of communication cost and robustness while achieving similar or superior accuracy to that of the FL. Moreover, the experimental results showed that the DS-FL with ER was more communication efficient and robust than the DS-FL with SA. To explain the performance of the proposed methods, we analyzed the experimental results from the perspective of entropy. Additionally, the impact of the open dataset volume was evaluated.

The future works will include developing the logit aggregation method, considering the individual device characteristics. For example, enhancing the impact of the logits uploaded by the reliable or high-performance client with respect to the global logit. However, how to evaluate the reliability of the clients, and how to control the impact of the uploaded logit are unknown. Another interesting direction is leveraging the logits had uploaded in the past round. In this work, the server and the clients used the logits uploaded at the current round. However, it might be useful to note that the logits had been uploaded in the past rounds. Moreover, another direction of future work is to design an FL framework performing under non-IID data distributions and unbalanced and massively distributed data while achieving communication costs scalability. To evaluate the FL framework under unbalanced and massively distributed data, the benchmarking framework for FL, LEAF [40], will be helpful.

ACKNOWLEDGMENTS

This work was supported in part by the JSPS KAKENHI under Grants JP17H03266 and JP18K13757, in part by the JST PRESTO under Grant JPMJPR2035, and in part by the KDDI Foundation.

REFERENCES

- [1] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1, pp. 1–121, Mar. 2021.
- [2] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.
- [3] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 22, no. 3, pp. 2031–2063, Third Quarter 2020.
- [4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. yArcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
- [5] T. Yang *et al.*, "Applied federated learning: Improving google keyboard query suggestions," 2018, *arXiv: 1812.02903*.
- [6] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under Non-IID private data," in *Proc. Conf. Neural Inf. Process. Syst., 2nd Workshop Mach. Learn. Phone Other Consum. Devices*, Nov. 2018, pp. 1–6.
- [7] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis, and S. L. Kim, "Mix2FLD: Downlink federated learning after uplink federated distillation with two-way mixup," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2211–2215, Oct. 2020.
- [8] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE 30th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2019, pp. 1–6.
- [9] J.-H. Ahn, O. Simeone, and J. Kang, "Cooperative learning via federated distillation over fading channels," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2020, pp. 8856–8860.
- [10] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," 2019, *arXiv: 1912.11279*.
- [11] S. Itahara, T. Nishio, M. Morikura, and K. Yamamoto, "A study for knowledge distillation based semi-supervised federated learning with low communication cost," in *Proc. RISING*, Nov. 2019, p. 1.
- [12] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, Sep. 2020.
- [13] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2017, *arXiv:1604.00981*.
- [14] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–7.
- [15] S. Dhakal, S. Prakash, Y. Yona, S. Talwar, and N. Himayat, "Coded federated learning," in *Proc. IEEE Globecom Workshops*, 2019, pp. 1–6.
- [16] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with Non-IID data," 2018, *arXiv:1806.00582*.
- [17] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1–10.
- [18] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," in *Proc. Conf. Neural Inf. Process. Syst.* 2016, pp. 1–5.
- [19] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 560–569.
- [20] S. Itahara, T. Nishio, M. Morikura, and K. Yamamoto, "Lottery hypothesis based unsupervised pre-training for model compression in federated learning," in *Proc. IEEE 92nd Veh. Technol. Conf.*, 2020, pp. 1–5.
- [21] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5050–5060.
- [22] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.
- [23] Z. Xianglong, F. Anmin, W. Huaqun, Z. Chunyi, and C. Zhenzhu, "A privacy-preserving and verifiable federated learning scheme," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
- [24] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. Dahl, and G. Hinton, "Large scale distributed neural network training through online distillation," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.
- [25] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4320–4328.
- [26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Representation Learn. Workshop*, 2014, pp. 1–9.
- [27] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–16.
- [28] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–34.
- [29] Y. Kang, Y. Liu, and T. Chen, "FedMVT: Semi-supervised vertical federated learning with multiview training," 2020, *arXiv: 2008.10838*.
- [30] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "A survey towards federated semi-supervised learning," 2020, *arXiv: 2002.11545*.
- [31] Z. Zhang, Z. Yao, Y. Yang, Y. Yan, J. E. Gonzalez, and M. W. Mahoney, "Benchmarking semi-supervised federated learning," 2020, *arXiv: 2008.11364*.
- [32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [34] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv: 1708.07747*.

- [35] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, pp. 142–150.
- [36] F. Chollet et al., "Keras," 2015. [Online]. Available: <https://keras.io>
- [37] D. D. Lewis et al., "Reuters-21578," 2004. [Online]. Available: <http://www.daviddlewis.com/resources/testcollections/reuters21578>
- [38] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, Apr. 2018.
- [39] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.
- [40] S. Caldas et al., "Leaf: A benchmark for federated settings," 2018, *arXiv: 1812.01097*.



Sohei Itahara (Student Member, IEEE) received the BE degree in electrical and electronic engineering from Kyoto University, in 2020. He is currently working toward the MI degree with the Graduate School of Informatics, Kyoto University.



Takayuki Nishio (Senior Member, IEEE) received the BE degree in electrical and electronic engineering and the master's and PhD degrees in informatics from Kyoto University, in 2010, 2012, and 2013, respectively. Since 2020, he has been an associate professor at the School of Engineering, Tokyo Institute of Technology, Japan. From 2013 to 2020, he was an assistant professor with the Graduate School of Informatics, Kyoto University. From 2016 to 2017, he was a visiting researcher with Wireless Information

Network Laboratory, Rutgers University, United States. His current research interests include machine learning-based network control, machine learning in wireless networks, and heterogeneous resource management.



Yusuke Koda (Member, IEEE) received the BE degree in electrical and electronic engineering from Kyoto University, in 2016, and the ME degree with the Graduate School of Informatics from Kyoto University, in 2018. He is currently working toward the PhD degree at the Graduate School of Informatics from Kyoto University. In 2019, he visited the Centre for Wireless Communications, University of Oulu, Finland to conduct collaborative research. He was the recipient of the Nokia Foundation Centennial Scholarship in

2019 and the VTS Japan Young Researcher's Encouragement Award in 2017. He is a member of the IEICE.



Masahiro Morikura (Member, IEEE) received the BE, ME, and PhD degrees from Kyoto University, Kyoto, Japan, in 1979, 1981 and 1991, respectively, all in electronic engineering. He joined NTT in 1981, where he was engaged in the research and development of TDMA equipment for satellite communications. From 1988 to 1989, he was with the communications Research Centre, Canada, as a guest scientist. From 1997 to 2002, he was active in standardization of the IEEE802.11a based wireless LAN. He was recipient of the

Paper Award and the Achievement Award from the IEICE, in 2000 and 2006, respectively, the Education, Culture, Sports, Science and Technology Minister Award, in 2007, and the Medal of Honor with Purple Ribbon from Japan's Cabinet Office, in 2015. He is currently a professor at the Graduate School of Informatics, Kyoto University. He is a Fellow of the IEICE.



Koji Yamamoto (Senior Member, IEEE) received the BE degree in electrical and electronic engineering from Kyoto University, in 2002, and the ME and PhD degrees in informatics from Kyoto University, in 2004 and 2005, respectively. Since 2005, he has been at the Graduate School of Informatics, Kyoto University, where he is currently an associate professor. From 2008 to 2009, he was a visiting researcher with Wireless, KTH, Royal Institute of Technology, Sweden. His research interests include radio resource management and applications of game theory. Since 2017, he has been an

editor of the *IEEE Wireless Communications Letters* and the track co-chairs of the APCC 2017 and the CCNC 2018. He was the recipient of the PIMRC 2004 Best Student Paper Award in 2004, the Ericsson Young Scientist Award in 2006, the Young Researcher's Award, the Paper Award, the SUEMATSU-Yasuharu Award from the IEICE of Japan in 2008, 2011, and 2016, respectively, and the IEEE Kansai Section GOLD Award in 2012. From 2004 to 2005, he was a research fellow of the Japan Society for the Promotion of Science.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**