TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/31 pp112–123
DOI: 10.26599/TST.2023.9010156
Volume 30, Number 1, February 2025

Ensemble Knowledge Distillation for Federated Semi-Supervised Image Classification

Ertong Shang, Hui Liu*, Jingyang Zhang, Runqi Zhao, and Junzhao Du*

Abstract: Federated learning is an emerging privacy-preserving distributed learning paradigm, in which many clients collaboratively train a shared global model under the orchestration of a remote server. Most current works on federated learning have focused on fully supervised learning settings, assuming that all the data are annotated with ground-truth labels. However, this work considers a more realistic and challenging setting, Federated Semi-Supervised Learning (FSSL), where clients have a large amount of unlabeled data and only the server hosts a small number of labeled samples. How to reasonably utilize the server-side labeled data and the client-side unlabeled data is the core challenge in this setting. In this paper, we propose a new FSSL algorithm for image classification based on consistency regularization and ensemble knowledge distillation, called EKDFSSL. Our algorithm uses the global model as the teacher in consistency regularization methods to enhance both the accuracy and stability of client-side unsupervised learning on unlabeled data. Besides, we introduce an additional ensemble knowledge distillation loss to mitigate model overfitting during server-side retraining on labeled data. Extensive experiments on several image classification datasets show that our EKDFSSL outperforms current baseline methods.

Key words: federated learning; semi-supervised learning; federated semi-supervised learning; knowledge distillation

1 Introduction

Today, Artificial Intelligence (AI) offers great potential for data analysis and decision making. Traditional cloud-based intelligent services require collecting a large amount of data and training intelligent models in a centralized manner. However, with the explosive growth of the number of networked devices, the existing cloud computing capabilities can hardly afford to store and analyze the massive amount of data generated by these devices. For example, International Data Corporation (IDC) estimates there will be 55.7 billion connected Internet of Things (IoT) devices by 2025, generating almost 80 zettabytes (ZB) of data[1]. In addition, states across the world are strengthening laws to protect users' privacy and data security, such as the General Data Protection Regulation (GDPR)[2] and the California Consumer Privacy Act (CCPA)[3], which makes it almost impossible to collect large-scale raw data from different devices or organizations, further hindering the development of cloud-based intelligent

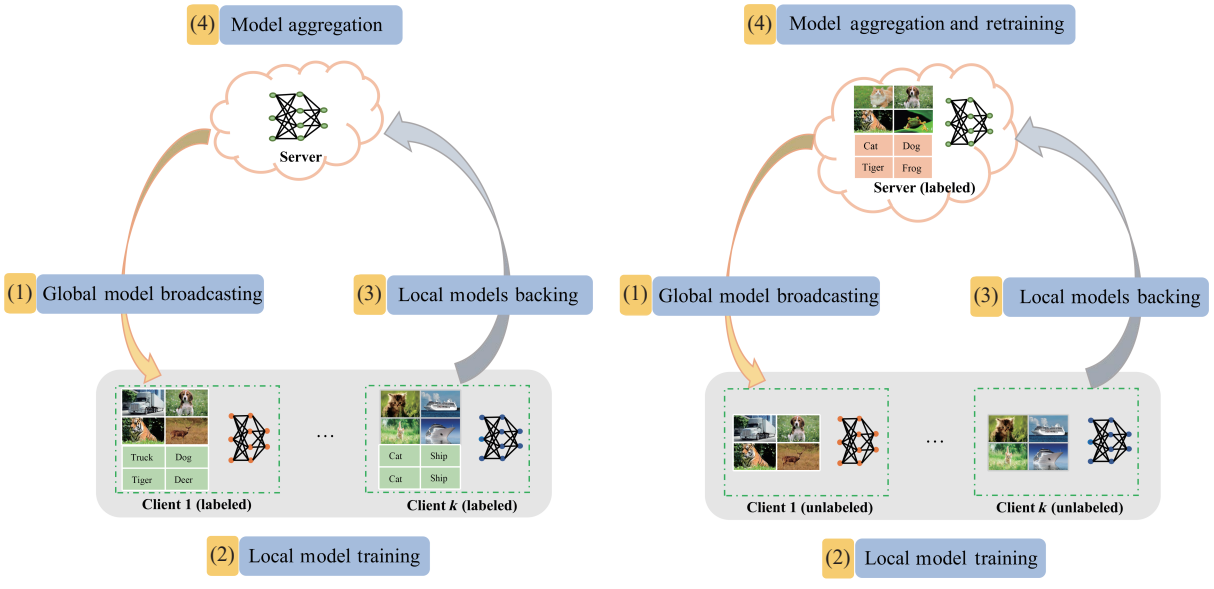
• Ertong Shang, Hui Liu, Jingyang Zhang, Runqi Zhao, and Junzhao Du are with School of Computer Science and Technology, Xidian University, Xi'an 710126, China. E-mail: shangertong@xidian.edu.cn; liuhui@xidian.edu.cn; 21031211391@stu.xidian.edu.cn; rqzhao@stu.xidian.edu.cn; dujz@xidian.edu.cn.
• Ertong Shang, Hui Liu, and Junzhao Du are also with Engineering Research Center of Blockchain Technology Application and Evaluation, Ministry of Education, Xi'an 710126, China, and Key Laboratory of Smart Human-Computer Interaction and Wearable Technology of Shaanxi Province, Xi'an 710126, China.
* To whom correspondence should be addressed.
Manuscript received: 2023-10-27; revised: 2023-12-10; accepted: 2023-12-19

applications.

Thanks to the growing computational and storage power on the mobile and IoT devices, storing data and executing computing process on the devices are becoming increasingly attractive, which motivates a new machine learning paradigm, Federated Learning (FL). FL provides a privacy-preserving distributed machine learning paradigm by keeping data locally. Recently, FL has received substantial interests from both academic and industrial researchers and has been widely applied in various applications, such as image processing[4], healthcare[5], and industrial engineering[6]. As shown in Fig. 1a, an FL system consists of a number of client devices that execute the training process and a central server that is responsible for parameter aggregation. Specifically, the following four steps are repeated in FL until the global model converges: (1) global parameter broadcasting: Active clients download the latest global model from the server; (2) local model training: Clients train client-side models with their local data; (3) local parameter uploading: Clients upload their updated local models to the server after local training; and (4) model aggregation: The server updates the global model by aggregating the received parameters.

Most existing studies on FL focus on fully supervised settings, assuming that all the client-side data are annotated with ground-truth labels. In many realistic scenarios, however, client users may not have enough incentives or expertise to label their generated data. In contrast, the server, which is usually operated by the service provider, may possess a small amount of data labeled by domain experts. This data regime leads to a more realistic and challenging FL setting, namely federated semi-supervised learning, where users' devices collectively hold a massive amount of unlabeled samples, whereas the server holds a small labeled dataset. Although we can train an intelligent model based on only the labeled data, the model performance will be significantly limited by the size of the server's dataset. Thus, the core goal of FSSL is to enhance the model trained on the server by utilizing the massive unlabeled data at the client side.

How to reasonably utilize the server-side labeled data and the client-side unlabeled data is the core challenge in FSSL. In this paper, we explore this problem on the image classification task. Specifically, for client-side unsupervised learning, we use consistency regularization methods to take full use of unlabeled data. However, we find that the local model is prone to collapse due to the lack of correct supervision information. To address this problem, we adopt the latest received global model as the teacher to guide client-side consistency training. For server-side model aggregation, the server first aggregates the model updates received from clients to get a global unsupervised model, and then retrains the aggregated model using its labeled data. Nevertheless, retraining may lead to the global model overfitting on the limited labeled dataset and forgetting the information learned



(a) Federated learning (b) Federated semi-supervised learning

Fig. 1 Framework of federated learning and Federated Semi-Supervised Learning (FSSL).

by the client-side training. Thus we enforce the global model to retain what the local model learned by penalizing the differences between the predicted distribution of the local model ensemble and the global model, named ensemble knowledge distillation loss. Combining all the components, we propose a new FSSL algorithm for image classification based on consistency regularization and ensemble knowledge distillation, named EKDFSSL. Overall, our main contributions can be summarized as follows:

• We focus on the labels-at-server federated semi-supervised learning and analyze the challenges faced by client-side unsupervised learning and server-side supervised learning.

• We propose a novel algorithm, named EKDFSSL, to address these challenges. Specifically, we propose a global model guided consistency regularization to enhance both the accuracy and stability of client-side unsupervised learning on unlabeled data. In addition, we introduce an additional ensemble knowledge distillation loss to mitigate model overfitting during server-side retraining on labeled data.

• We experimentally evaluate the feasibility of our EKDFSSL for standard image classification tasks with four publicly available datasets. Experiment results show that EKDFSSL outperforms recent baseline methods. In addition, we perform sufficient ablation study to demonstrate the role of each component in our method, guaranteeing the interpretability of the method.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 formulates the problem and presents our EKDFSSL algorithm. Section 4 reports the evaluation of our approach. Finally, Section 5 concludes this paper.

2 Related Work

2.1 Federated learning

As a distributed model training paradigm used in edge computing, FL faces three inherent challenges: statistical heterogeneity, communication efficiency, and systems heterogeneity[7, 8]. Statistical heterogeneity arises due to varying local data distributions between different clients, leading to a local bias issue where models overfit to the biased local distribution, which can seriously impact the accuracy of the global model. There are two complementary approaches to solve this problem. On one hand, FedProx[9], MOON[10], and FedNTD[11] all improved local training by adding a regularization term. On the other hand, FedNova[12] and FedAvgM[13] aimed to improve the model aggregation phase. The limited bandwidth of client devices causes the communication cost to be a key bottleneck in the federated network. Konecny et al.[14] and Caldas et al.[15] reduced the amount of data traffic by using some communication compression techniques. Moreover, the communication, storage, and computational capabilities of each client in federated networks may differ due to variability in hardware resources, leading to some clients failing to complete local training within the prescribed time, which are called stragglers. To alleviate this problem, Xie et al.[16] proposed an asynchronous federated optimization algorithm, providing staleness tolerance for stragglers. Nishio and Yonetani[17] explored active device sampling policies based on systems resources. In summary, although there have been a lot of researches devoted to addressing the different challenges in FL, most of them have focused on fully supervised settings, assuming that all samples are labeled with artificial annotations, which is often unrealistic for real-world applications.

2.2 Federated semi-supervised learning

Federated semi-supervised learning considers a practical problem of deficiency of labels in FL and has been studied in various real-world applications such as COVID-19 medical segmentation[18], travel mode identification[19], and human activity recognition[20]. Federated semi-supervised learning can be divided into two categories based on where the labeled data are located: (1) Labels-at-server in which clients own only unlabeled data and a few labeled data are available at the server[21−25]; and (2) labels-at-clients in which each client contains partially labeled data while the server has no data[21−24, 26]. In this paper, we focus on the labels-at-server FSSL setting and introduce some of the related work in this setup. FedMatch[21] is the first study in FSSL, which introduced parameter decomposition for disjoint learning between the supervised learning on the server side and the unsupervised learning on the client side. In addition, it proposed an inter-client consistency loss to enhance local training, improving the performance upon naive combinations of FL and SSL approaches. FedRGD[22] demonstrated that the large gradient diversity of local models is a critical issue that affects the global model

performance and proposed to use group normalization and a novel grouping-based model averaging method to deal with this problem. FedMix[23, 24] proposed a novel model parameter mixing strategy to aggregate the server-side supervised model, client-side unsupervised models, and the latest global model. Recently, GDST[25] proposed a global distillation loss that utilizes outputs of global model for local unlabeled samples as supervision information to enhance client-side self-training. In addition, it used the server-side labeled data to fine-tune the aggregated global model. It can be observed that the above works either improve client-side training or modify the parameter aggregation mechanism, while this paper aims to improve both client-side training and server-side global model optimization.

2.3 Knowledge distillation

Knowledge distillation refers to the transfer of knowledge from one model to another, which is originally used for model compression[27, 28]. The teacher-student architecture is a basic structure to form the knowledge transfer. Recently, knowledge distillation is widely used in FL to solve various challenges. For data heterogeneity, FedNTD[11] and FedGKD[29] took the global model as the teacher to guide local training, forcing the local model to retain global knowledge, thereby alleviating the issue of local bias. Orthogonal to them, FedFTG[30], FedBKD[31], and FedBiKD[32] fine-tuned the aggregated model using knowledge distillation on the server side, so that knowledge can be transferred from the local model to the teacher model more efficiently, improving the performance of the global model. For expensive communication, FD[33] exchanged logits as knowledge instead of model parameters between servers and clients, significantly reducing the amount of data transferred in the federated network. FedKD[34] trained a small model and a large model simultaneously on the client side, which learn and distill knowledge from each other. Only the small model is shared by different clients for collaborative learning, effectively reducing the communication cost. CMFL[35] improved communication efficiency by identifying and excluding irrelevant client-side updates. For model heterogeneity, FedMD[36] and FedDF[37] used knowledge distillation instead of parameter averaging to update the global model, enabling clients to learn collectively on heterogeneous models. Different from previous studies, in this article, we employ knowledge distillation in federated semi-supervised learning to prevent the server-side global model from overfitting on the limited labeled data.

3 Proposed Method

In this section, we first describe the problem setup of federated semi-supervised learning with some notations. Then, we present our method, covering the local training on the unlabeled client side and the aggregation as well as fine-tuning of global model on the server side.

3.1 Problem setup

In this work, we consider the labels-at-server FSSL with a fully-labeled server and M fully-unlabeled clients. Formally, the server has a labeled dataset $D_{\mathrm{s}} = \{(x_i, y_i)\}_{i=1}^{n_{\mathrm{s}}}$, where x_i denotes the feature vector (i.e., image), y_i denotes its corresponding label (i.e., image class), and n_{s} denotes the number of samples available at the server, respectively. Meanwhile, each client m owns an unlabeled dataset $D_m = \{u_i\}_{i=1}^{n_m}$, for $m \in \{1, 2, \ldots, M\}$, where u_i denotes the unlabeled feature vector, and n_m denotes the number of samples available at the client m. The number of labeled training samples at the server is considered to be much smaller than the total number of unlabeled training samples at the clients, i.e., $n_{\mathrm{s}} \ll \sum_{m=1}^{M} n_m$. Like universal federated framework, the goal of FSSL is also to learn a shared global model, and the local datasets, including the server-side labeled dataset, cannot be exchanged or shared across the federated network.

3.2 Federated semi-supervised learning with consistency regularization and ensemble knowledge distillation

To make efficient use of unlabeled data on the client side and labeled data on the server side, we propose a novel FSSL algorithm with consistency regularization and ensemble knowledge distillation, EKDFSSL. As shown in Fig. 1b, similar to traditional supervised FL, EKDFSSL also involves four steps in each FL round. In addition to aggregating the global model, however, the server also participates in training by updating the aggregated global model using its labeled dataset. Algorithm 1 summarizes the training process of EKDFSSL. Next, we will give the details of the proposed EKDFSSL method.

Algorithm 1 EKDFSSl algorithm

Input: server-side labeled dataset D_s, private unlabeled dataset D_m for each client, $m = 1,2,\ldots,M$, the number of communication rounds R, the number of sampling clients A, the number of local epochs E, the learning rate η, and the weight of knowledge distillation loss α.

Output: the final global model w^{R+1}

Server executes:

Initialize the global model w^0

for each round $r = 1,2,\ldots,R$ **do**

$M^r \leftarrow$ random sample of A clients

Broadcast global model w^r to M^r

for each client $m \in M^r$ **do**

$w_m^r \leftarrow$ ClientUpdate (D_m, w^r)

end for

$\alpha = \frac{r}{R}$

Model aggregation: $w_s^r \leftarrow \sum_{m \in M^r} \frac{|D_m|}{\sum_{m \in M^r} |D_m|} w_m^r$

$w^{r+1} \leftarrow$ ServerUpdate $(D_s, \alpha, w_s^r, w_1^r, \ldots, w_A^r)$

end for

ClientUpdate (D_m, w^r):

$w_g, w_m \leftarrow w^r$

for each local epoch $e = 1,2,\ldots,E$ **do**

for each mini-batch (u_b) of D_m **do**

$\mathcal{L}_{CR} = H(f(w; \mathcal{A}(u_b)), f(w_g; \alpha(u_b)))$ ▷ Eq. (1)

$W_m \leftarrow W_m - \eta \cdot \nabla \mathcal{L}_{cr}$

end for

end for

Return w_m

ServerUpdate $(D_s, \alpha, w_s, w_1, \ldots, w_A)$:

// Fine-tune the aggregated model with the server's labeled data

for each local epoch $e = 1,2,\ldots,E$ **do**

for each mini-batch (x_b, y_b) of D_s **do**

$\mathcal{L}_{CE} = H(f(w_s; x_b), y_b)$ ▷ Eq. (2)

$\bar{y}_b = \frac{1}{A} \sum_{i=1}^{A} f(w_i; x_b)$ ▷ Eq. (3)

$\mathcal{L}_{KD} = \mathrm{KL}(f(w_s; x_b), \bar{y}_b)$ ▷ Eq. (4)

$w_s \leftarrow w_s - \eta \cdot \nabla(\mathcal{L}_{CE} + \alpha \mathcal{L}_{KD})$

end for

end for

Return w_s

3.2.1 Consistency regularization for unlabeled client training

Consistency regularization, one of the most commonly used approaches in the semi-supervised learning paradigm, is based on the smoothness assumption that points close to each other in the input space are more likely to share the same label. This category of methods makes use of unlabeled data by enforcing model predictions to be invariant under different augmentations of the same data point. The basic structure of consistency regularization based methods is the teacher-student structure, where the teacher model is used to generate target predictions, while the student model learns by producing similar predictions on perturbed samples. In the existing SSL methods, the teacher and the student can be the same model or different models. For example, the self-ensembling method, Π model[38], and Virtual Adversarial Training (VAT)[39] generated targets using the currently being trained model, while mean teacher[40] used the Exponential Moving Average (EMA) weight of past student models to produce a more accurate teacher model.

As described in Section 3.1, labeled and unlabeled data are disjoint in the labels-at-server FSSL, meaning that no labeled data can provide direct supervision information for local training at the client side. In this disjoint data setting, using the local model being trained as the teacher model may lead to model collapse due to the lack of correct supervision information, resulting in performance of the trained model deteriorating sharply. To meet this challenge, we adopt the latest received global model as the teacher model for client-side consistency training. Besides, recent work demonstrated that the quality of data augmentations plays a crucial role in consistency regularization[41]. It substitutes simple noising operations with advanced data augmentation methods, such as RandAugment, which combines various augmentation transformations uniformly sampled from all image processing transformations in the Python Image Library (PIL). Therefore, we also use the efficient data augmentations described in Ref. [41]. As a result, the consistency regularization loss used in client-side unsupervised learning in our EKDFSSL is as follows:

$$\mathcal{L}_{CR} = H(f(w; \mathcal{A}(u)), f(w_g; \alpha(u))) \tag{1}$$

where $f(w;x)$ denotes the output probability distribution of the model w for input x, and $H(p,q)$ denotes cross-entropy between two distributions. $\alpha(\cdot)$ represents the weakly-augmentation operations such as flipping, shifting, and croping, and $\mathcal{A}(\cdot)$ represents the strongly-augmentation operations such as RandAugment. w_g and w represent the parameters of

the global model and the currently local model, respectively.

3.2.2 Ensemble knowledge distillation

Different from vanilla FL, there is a small amount of labeled data on the server in the FSSL setup. In our EKDFSSL, these labeled data are used to fine-tune the aggregated unsupervised model to improve the performance of the global model. For image classification tasks, cross-entropy is a widely used loss function, which measures the difference between the predicted probability distributions of the model and the true label. Its formal definition is as follows:

$$\mathcal{L}_{\mathrm{CE}} = H(f(w;x),y) = -\sum_{i=1}^{C} y^{(i)} \log(f(w;x)^{(i)}) \tag{2}$$

where y is the one-hot label for x, and $y^{(i)}$ represents the i-th element of y. $f(w;x)$ is the softmax probability output of the model w for input x, $f(w;x)^{(i)}$ stands for the prediction probability on class i, and C is the total number of classes to identify, respectively.

In real-world applications, the number of labeled samples on the server is often very limited due to the high cost of manual labeling. In this case, using only the above classification loss function may cause the global model to overfit on the server-side data, reducing the generalization ability of the global model. To mitigate this problem, we introduce an additional adaptive knowledge distillation loss, which views all local models uploaded as the teacher ensemble and the aggregated global model as the student, enforcing the global model to retain what the local model learned by penalizing the differences between the predicted distribution of teacher and student models. As described in Algorithm 1 and Fig. 2, after aggregating the client-side unsupervised models, we fine-tune the aggregated model w_s using the following three steps: (1) Each client model w_i outputs its own predicted probability $f(w_i;x)$ for the server-side dataset D_s; (2) these probability outputs are averaged as ensemble knowledge; and (3) the server uses both cross-entropy classification loss and the proposed ensemble knowledge distillation loss with labeled data to fine-tune the model w_s. Specifically, the ensemble knowledge distillation loss is calculated by the Kullback–Leibler Divergence (KL Divergence) between the average predicted outputs of client models and the outputs of the global model, which is formally defined as follows:

$$\bar{y} = \frac{1}{A}\sum_{i=1}^{A} f(w_i;x) \tag{3}$$

$$\mathcal{L}_{\mathrm{KD}} = \mathrm{KL}(f(w_s;x),\bar{y}) = -\sum_{i=1}^{C} \bar{y}^{(i)} \log\left(\frac{f(w_s;x)^{(i)}}{\bar{y}^{(i)}}\right) \tag{4}$$

In summary, the loss function used on the server side is composed of two parts. The first part is a typical classification loss for the classifier retraining. The second part is our proposed ensemble knowledge distillation loss, which avoids overfitting the global model on limited labeled data. In addition, we

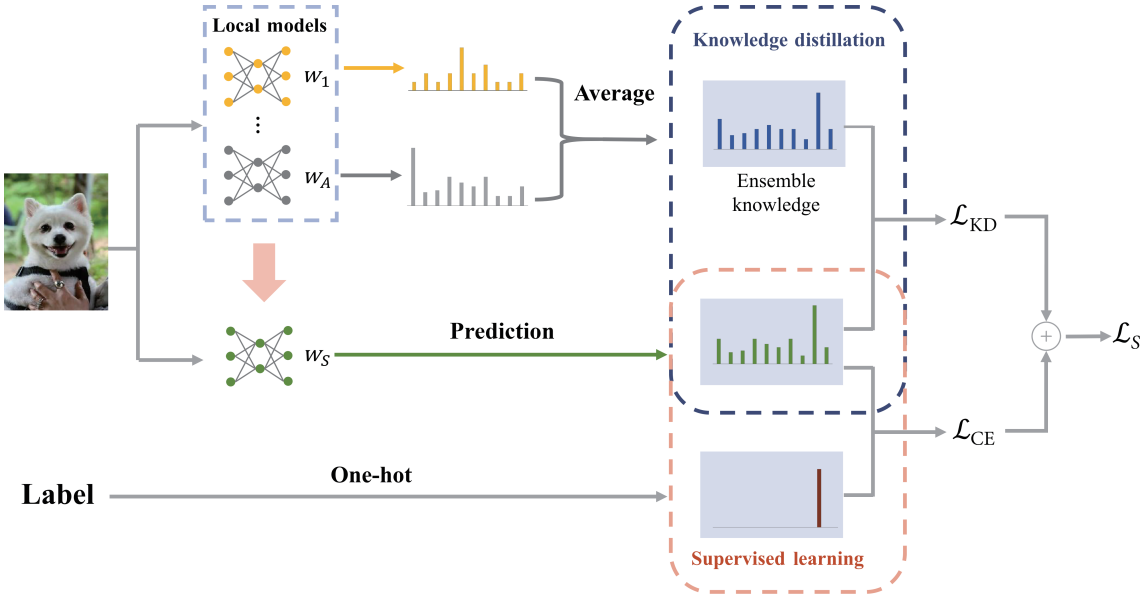


Fig. 2 Training process on the server side.

introduce an adaptive coefficient α for progressively increasing the weight of the knowledge distillation loss during the training process. Specifically, α is calculated by $\alpha = r/R$, where r and R represent the current round and the number of total communication rounds, respectively. The final loss function for the server-side model retraining is

$$\mathcal{L}_S = \mathcal{L}_{\mathrm{CE}} + \alpha \mathcal{L}_{\mathrm{KD}} \tag{5}$$

4 Experiment

4.1 Experiment setup

Dataset and data partitioning. To verify the effectiveness of our proposed method, we conduct a comprehensive experiment on four popular image classification datasets, Fashion-MNIST[42], SVHN[43], CIFAR-10[44], and CIFAR-100[44]. The first three datasets contain 10 image classes each and the last one contains 100 classes. Specifically, Fashion-MNIST comprises of 28 × 28 grayscale images of 70 000 fashion products from 10 categories. SVHN is obtained from house numbers in Google Street View images, consisting of 73 257 training samples as well as 26 032 testing samples. CIFAR-10 and CIFAR-100 consist of 50 000 training images and 10 000 testing images with 10 and 100 image classes, including people, animals, flowers, insects, and so on. To simulate the FSSL setting, we first sample a small number (N_s) of images from the training set as labeled data on the server side, while the remaining images are allocated to clients as unlabeled samples. For unlabeled data partitioning, we consider both Independent and Identically Distributed (IID) and Non-Independent and Identically Distributed (Non-IID) settings.

- **IID**: Each client is randomly assigned the same number of images, which means that all clients are subject to the same distribution as the entire dataset.
- **Non-IID**: Following existing works in Refs. [10, 11], we use a Dirichlet distribution Dir (1.0) to generate the non-IID data partition in clients, where the numbers of classes and samples at each client differ from each other.

Implementation detail. We employ a 13-layer convolutional neural network widely used in semi-supervised learning[39, 40] as the backbone model for the first three datasets, and Wide-ResNet-28-2[45] as the backbone model for CIFAR-100. The total number of clients M and the number of clients selected in each communication round A are 100 and 10, respectively. The number of training rounds R is set to 500, local training epoch E is 5, and mini-batch size B is 30. The number of the labeled data at the server N_s is 500 for the first three datasets, and is 5000 for CIFAR-100. We use Stochastic Gradient Descent (SGD) to optimize the global model as well as local models and use a cosine decay as the scheduler of learning rate. The initial value of the learning rate is set as 0.01 and the local momentum is 0.9.

4.2 Baseline method

To fairly validate the proposed EKDFSSL method, we use the following baselines.

- Server-only refers to training with only server-side labeled data, which is considered as the lower bound for any effective FSSL algorithm.
- FedMatch[21] adopted parameter decomposition to separate supervised learning on the server side and unsupervised learning on the client side, while also introducing an inter-client consistency loss to enhance local training.
- FedRGD[22] aimed to reduce the gradient diversity among local models by using group normalization and grouping-based model averaging.
- FedMix[23, 24] proposed a novel model parameter mixing strategy, which aggregates the server-side supervised model, client-side unsupervised model, and the latest previous round of global models to obtain a new global model.
- GDST[25] utilized self-training and the proposed global distillation loss for local training, and retrained the aggregated model with server-side labeled data. The global distillation loss utilizes the outputs of global model in local unlabeled data as supervision information to enhance client-side training.

4.3 Overall result

We compare our proposed method with other baseline approaches on the four datasets described above. In each communication round, the global model is tested on the test set and the highest test accuracy is reported for all methods. As shown in Table 1, our EKDFSSL not only significantly outperforms the server-only baseline, but also outperforms its competitors in all settings, which demonstrates the validity of our proposed EKDFSSL algorithm. For example, on the CIFAR-10 dataset, EKDFSSL has an averaged accuracy advantage of 21.59% compared to server-only and an averaged accuracy advantage of 9.30% over the

Table 1 Highest accuracy of our method and the other baseline methods on Fashion-MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets.

Method	Highest accuracy (%)							
	Fashion-MNIST		SVHN		CIFAR-10		CIFAR-100	
	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
Server-only	83.53	83.53	70.26	70.26	54.42	54.42	46.45	46.45
FedMatch	81.62	81.69	77.49	77.20	52.73	53.07	44.91	44.97
FedRGD	83.41	83.61	73.66	72.08	48.76	48.60	46.97	46.78
FedMix	82.67	83.44	78.16	77.11	57.45	56.99	48.23	48.74
GDST	83.61	83.31	89.30	89.19	67.05	66.35	49.51	49.09
EKDFSSL	**87.21**	**87.15**	**90.72**	**90.26**	**77.21**	**74.80**	**52.39**	**51.47**

best baseline method, GDST. The results are similar on the CIFAR-100 dataset, which is more difficult to classify.

Data heterogeneity is a common phenomenon in FL, which usually causes global model performance degradation. To prove the effectiveness of our method on heterogeneous data, we conducted experiments in both IID and non-IID data distributions. As shown in Table 1, the performance gap of our EKDFSSL between the two data distributions is very small. Specifically, with the exception of the CIFAR-10 dataset, our approach has a performance gap of less than 1.00% between IID and non-IID data. In other words, the heterogeneous data distribution between clients does not seriously affect our approach, proving the practicality of our approach in real-world scenarios.

Another obvious phenomenon is that the performance of the first three methods (FedMatch, FedRGD, and FedMix) is significantly lower than that of the latter two methods (GDST and EKDFSSL), and even lower than that of server-only in some cases. The main difference between these two types of methods is that the first three methods directly aggregate the server-side supervised model and client-side unsupervised models to get the global model, while the latter two methods first aggregate the client-side local models and then fine-tune the aggregated global model using server-side labeled data. This result demonstrates the importance of fine-tuning the global model using server-side labeled data.

4.4 Analysis and ablation study

In general, the amount of labeled data significantly affects the performance of semi-supervised learning methods. Therefore, we first compare the performance of different algorithms with different numbers of labeled samples. In addition, the main contributions of our EKDFSSL include the use of the global model as the teacher in consistency regularization methods and the introduction of additional knowledge distillation losses when fine-tuning the global model. Then, we will analyze the role of these two components separately.

Varying the size of labeled data. We explore the impact of the amount of labeled data held by the server on the model performance. As shown in Table 2, our EKDFSSL not only beats its competitors in all settings, but the performance advantage increases as the labeled data size N_s decreases. Specifically, when $N_s = 1000$, EKDFSSL has an averaged accuracy advantage of 13.82% compared to server-only and an averaged accuracy advantage of 3.19% over the best baseline

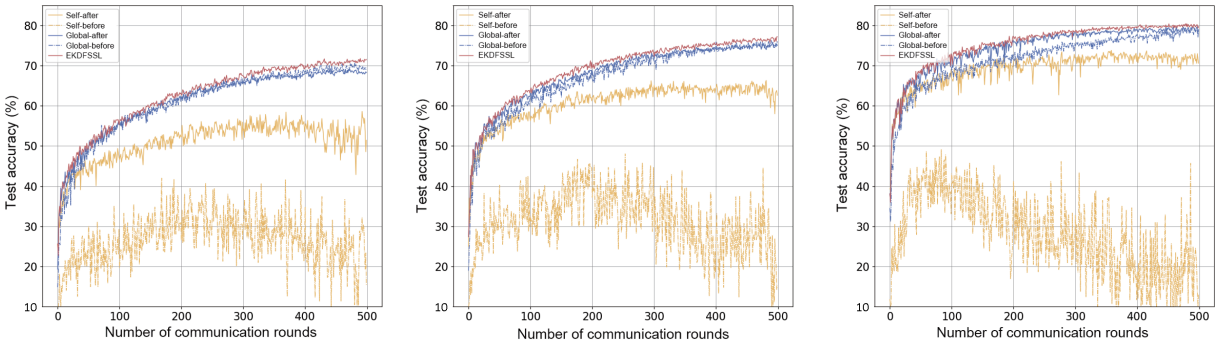
Table 2 Highest accuracy of our method and the other baseline methods on CIFAR-10 dataset with different numbers of labeled samples.

Method	Highest accuracy (%)					
	$N_s = 250$		$N_s = 500$		$N_s = 1000$	
	IID	Non-IID	IID	Non-IID	IID	Non-IID
Server-only	44.13	44.13	54.42	54.42	66.02	66.02
FedMatch	39.85	40.37	52.73	53.07	68.20	68.22
FedRGD	43.22	43.22	48.76	48.60	60.60	60.42
FedMix	39.15	40.14	57.45	56.99	69.03	68.65
GDST	54.02	53.80	67.05	66.35	76.34	76.95
EKDFSSL	**71.79**	**68.12**	**77.21**	**74.80**	**80.45**	**79.22**

method, GDST. When N_s is reduced to 250, EKDFSSL not only outperforms server-only by a large margin, but also has an averaged accuracy advantage of 16.05% over GDST. This is because when the number of labeled samples is large, the impact of local-side unsupervised training and parameter aggregation on the global model performance is relatively small, since using only labeled samples can train a good model. Conversely, when a small amount of labeled data are insufficient to train a good global model, local-side unsupervised training and parameter aggregation play a crucial role. In other words, this proves the advantages of our proposed EKDFSSL algorithm in unsupervised training and parameter aggregation, as well as the effectiveness of our method in the case of a small number of labeled samples.

Global-teacher vs. self-teacher. As described in Section 3.2.1, consistency regularization methods are based on the teacher-student structure. In client-side local training, the teacher can be either the latest global model received from the server (denoted as global-teacher) or the local model currently being trained (denoted as self-teacher). This study experimentally proves that local training with self-teacher leads to model collapse. Specifically, we record the test accuracy of the aggregated model (i.e., before server-side retraining) and the global model (i.e., after server-side retraining) in each training round. As shown in Fig. 3, global-teacher significantly outperforms self-teacher in terms of global model performance. In addition, in self-teacher mode, regardless of the value of N_s, the accuracy of the aggregated model is poor and fluctuates greatly, which indicates that the consistency training with self-teacher leads to collapses of local models. In contrast, using the global model as the teacher can provide reliable supervision information for local training, so the performance of the aggregated model has steadily improved in global-teacher mode.

EKDFSSL without EKD. In this section, we explore the effects of our proposed EKD loss in the global-teacher model. By comparing global-before and global-after in Fig. 3, we can see that when N_s is large, server-side fine-tuning can improve the performance of the global model, which verifies the necessity of retraining on the server side. However, when N_s decreases to 250, server-side fine-tuning even damages the performance. This means that using only the classification loss function may cause the global model to overfit on the server-side limited data, reducing the generalization ability of the global model. To address this problem, we propose an additional ensemble knowledge distillation, enforcing the global model to retain what the local models learned. As shown in Fig. 3, server-side retraining with additional ensemble global distillation can improve the performance of the aggregated model in all settings. In addition, as indicated in Table 3, using both EKD loss and cross-entropy loss also achieves better performance than using only cross-entropy loss, and the performance improvement increases as N_s decreases. For example, when $N_s = 1000$, the test accuracy is improved by an average of 0.79%. As N_s decreases to 250, the test accuracy is improved by an average of 2.12%. In summary, these results demonstrate not only the necessity for fine-tuning on the server side, but also the



(a) $N_s = 250$ (b) $N_s = 500$ (c) $N_s = 1000$

Fig. 3 Performance comparison on CIFAR-10 dataset between our EKDFSSL, global-teacher based consistency regularization, and self-teacher based consistency regularization. The naming convention in the figures is as follows: teacher mode-before or after retraining. For example, self-after denotes that the local model is used as the teacher model in consistency regularization and records the accuracy of the global model after retraining; self-before denotes that the local model is used as the teacher model in consistency regularization and records the accuracy of the aggregated model before retraining.

Table 3 Highest test accuracy of our EKDFSSL without EKD on CIFAR-10 dataset with different numbers of labeled samples.

N_s	Setting	Highest accuracy (%)	
		EKDFSSL	Without EKD
250	IID	71.79	69.13
	Non-IID	68.12	66.55
500	IID	77.21	75.65
	Non-IID	74.80	74.59
1000	IID	80.45	79.67
	Non-IID	79.22	78.43

importance of an effective retraining mechanism.

5 Conclusion

In this paper, we focus on the labels-at-server FSSL setup and propose a novel method EKDFSSL that considers both unsupervised learning on the client side and supervised retraining on the server side. Specifically, training on client-side unlabeled data may lead to model collapse due to the lack of reliable supervision information. To solve this problem, we propose a global model guided consistency regularization method, which adopts the latest received global model as the teacher model for client-side consistency training, enhancing both the accuracy and stability of local models. After model aggregation, retraining on server-side labeled data may cause the global model overfitting, especially when the number of labeled data is very limited. We introduce an additional ensemble knowledge distillation loss, which enforces the global model to retain what the local models learned, reducing the overfitting and improving the generalization ability of the global model. By combining all the components, our method outperforms current baseline methods on the classic image classification datasets. In future work, we can improve local training methods to make more efficient use of unlabeled data on the client side. In addition, we plan to investigate a more challenging FSSL setting where the data distribution mismatches between unlabeled clients and the labeled server.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. 62032017 and 62272368), Key Talent Project of Xidian University (No. QTZX24004), Innovation Capability Support Program of Shaanxi (No. 2023-CX-TD-08), Shaanxi Qinchuangyuan "Scientists + Engineers" Team (No.2023KXJ-040), and Science and Technology Program of Xi'an (No. 23KGDW0005-2022).

References

[1] J. Hojlo, Future of industry ecosystems: Shared data and insights, https://blogs.idc.com/2021/01/06/future-of-industryecosystems-shared-data-and-insights, 2021.

[2] P. Regulation, General data protection regulation, https://eur-lex.europa.eu/legalcontent/EN/TXT/PDF/?uri=CELEX:32016R0679, 2016.

[3] E. Illman and P. Temple, California consumer privacy act (CCPA), https://oag.ca.gov/privacy/ccpa, 2018.

[4] F. A. KhoKhar, J. H. Shah, M. A. Khan, M. Sharif, U. Tariq, and S. Kadry, A review on federated learning towards image processing, *Comput. Electr. Eng.*, vol. 99, p. 107818, 2022.

[5] R. S. Antunes, C. A. da Costa, A. Küderle, I. A. Yari, and B. Eskofier, Federated learning for healthcare: Systematic review and architecture proposal, *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–23, 2022.

[6] P. Boobalan, S. P. Ramu, Q. V. Pham, K. Dev, S. Pandya, P. K. R. Maddikunta, T. R. Gadekallu, and T. Huynh-The, Fusion of Federated Learning and Industrial Internet of Things: A survey, *Comput. Netw.*, vol. 212, p. 109048, 2022.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, *Found. Trends Mach. Learn.*, vol. 14, nos. 1&2, pp. 1–210, 2021.

[8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, Federated learning: Challenges, methods, and future directions, *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, Federated optimization in heterogeneous networks, in *Proc. 3rd Machine Learning and Systems*, Austin, TX, USA, 2020, pp. 429–450.

[10] Q. Li, B. He, and D. Song, Model-contrastive federated learning, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 10713–10722.

[11] G. Lee, M. Jeong, Y. Shin, S. Bae, and S. Y. Yun, Preservation of the global knowledge by not true distillation in federated learning, in *Proc. 36th Int. Conf. Neural Information Processing Systems*, New Orleans, LA, USA, 2022, pp. 38461–38474.

[12] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, Tackling the objective inconsistency problem in heterogeneous federated optimization, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Virtual, 2020, pp. 7611–7623.

[13] T. M. H. Hsu, H. Qi, and M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, arXiv preprint arXiv: 1909.06335, 2019.

[14] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, Federated learning: Strategies for improving communication efficiency, arXiv preprint arXiv: 1610.05492, 2016.

[15] S. Caldas, J. Konecny, H. B. McMahan, and A. Talwalkar, Expanding the reach of federated learning by reducing client resource requirements, arXiv preprint arXiv: 1812.07210, 2018.
[16] C. Xie, S. Koyejo, and I. Gupta, Asynchronous federated optimization, arXiv preprint arXiv: 1903.03934, 2019.
[17] T. Nishio and R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in *Proc. IEEE Int. Conf. Communications* (*ICC*), Shanghai, China, 2019, pp. 1–7.
[18] D. Yang, Z. Xu, W. Li, A. Myronenko, H. R. Roth, S. Harmon, S. Xu, B. Turkbey, E. Turkbey, X. Wang, et al., Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan, *Med. Image Anal.*, vol. 70, p. 101992, 2021.
[19] Y. Zhu, Y. Liu, J. J. Q. Yu, and X. Yuan, Semi-supervised federated learning for travel mode identification from GPS trajectories, *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 3, pp. 2380–2391, 2022.
[20] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, Semi-supervised federated learning for activity recognition, arXiv preprint arXiv: 2011.00851, 2020.
[21] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, Federated semi-supervised learning with inter-client consistency & disjoint learning, arXiv preprint arXiv: 2006.12097v3, 2020.
[22] Z. Zhang, Y. Yang, Z. Yao, Y. Yan, J. E. Gonzalez, K. Ramchandran, and M. W. Mahoney, Improving semi-supervised federated learning by reducing the gradient diversity of models, in *Proc. IEEE Int. Conf. Big Data* (*Big Data*), Orlando, FL, USA, 2021, pp. 1214–1225.
[23] Z. Zhang, S. Ma, J. Nie, Y. Wu, Q. Yan, X. Xu, and D. Niyato, Semi-supervised federated learning with non-IID data: Algorithm and system design, in *Proc. IEEE 23rd Int. Conf. High Performance Computing & Communications; 7th Int. Conf. Data Science & Systems; 19th Int. Conf. Smart City; 7th Int. Conf. Dependability in Sensor, Cloud & Big Data Systems & Application* (*HPCC/DSS/SmartCity/DependSys*), Haikou, China, 2021, pp. 157–164.
[24] Z. Zhang, S. Ma, Z. Yang, Z. Xiong, J. Kang, Y. Wu, K. Zhang, and D. Niyato, Robust semi-supervised federated learning for images automatic recognition in Internet of drones, *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5733–5746, 2023.
[25] X. Liu, L. Zhu, S. T. Xia, Y. Jiang, and X. Yang, GDST: Global distillation self-training for semi-supervised federated learning, in *Proc. IEEE Global Communications Conf*. (*GLOBECOM*), Madrid, Spain, 2021, pp. 1–6.
[26] L. Che, Z. Long, J. Wang, Y. Wang, H. Xiao, and F. Ma, Fedtrinet: A pseudo labeling method with three players for federated semi-supervised learning, in *Proc. 7th IEEE Int. Conf. Big Data*, Huizhou, China, 2021, pp. 715–724.
[27] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, Model compression, in *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, 2006, pp. 535–541.
[28] G. Hinton, O. Vinyals, and J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv: 1503.02531, 2015.
[29] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, Z. Xu, and L. Sun, Local-global knowledge distillation in heterogeneous federated learning with non-IID data, arXiv preprint arXiv: 2107.00051, 2021.
[30] L. Zhang, L. Shen, L. Ding, D. Tao, and L. Y. Duan, Fine-tuning global model via data-free knowledge distillation for non-IID federated learning, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 10174–10183.
[31] P. Qi, X. Zhou, Y. Ding, Z. Zhang, S. Zheng, and Z. Li, FedBKD: Heterogenous federated learning via bidirectional knowledge distillation for modulation classification in IoT-edge system, *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 1, pp. 189–204, 2023.
[32] E. Shang, H. Liu, Z. Yang, J. Du, and Y. Ge, FedBiKD: Federated bidirectional knowledge distillation for distracted driving detection, *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11643–11654, 2023.
[33] 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, arXiv preprint arXiv: 1811.11479, 2018.
[34] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, Communication-efficient federated learning via knowledge distillation, *Nat. Commun.*, vol. 13, no. 1, p. 2032, 2022.
[35] L. Wang, W. Wang, and B. Li, CMFL: Mitigating communication overhead for federated learning, in *Proc. IEEE 39th Int. Conf. Distributed Computing Systems* (*ICDCS*), Dallas, TX, USA, 2019, pp. 954–964.
[36] D. Li and J. Wang, FedMD: Heterogenous federated learning via model distillation, arXiv preprint arXiv: 1910.03581, 2019.
[37] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, Ensemble distillation for robust model fusion in federated learning, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, pp. 2351–2363.
[38] S. Laine and T. Aila, Temporal ensembling for semisupervised learning, arXiv preprint arXiv: 1610.02242, 2016.
[39] T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, 2019.
[40] A. Tarvainen and H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 1195–1204.
[41] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, Unsupervised data augmentation for consistency training, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Virtual, 2020, pp. 6256–6268.
[42] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv: 1708.07747, 2017.
[43] N. Yuval, Reading digits in natural images with

unsupervised feature learning, in *Proc. 25th NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Granada, Spain, 2011, pp. 1–9.

[44] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, Technical report, University of Toronto, Toronto, Canada, 2009.

[45] S. Zagoruyko and N. Komodakis, Wide residual networks, arXiv preprint arXiv: 1605.07146, 2016.

Ertong Shang received the BS degree from Xidian University, Xi'an, Shaanxi, China, in 2017. He is currently pursuing the PhD degree at Xidian University, China. His research interests include federated learning, edge AI, mobile computing, and semi-supervised learning.

Jingyang Zhang received the BS degree from Xidian University, Xi'an, Shaanxi, China, in 2021. He is currently pursuing the MS degree at School of Computer Science and Technology, Xidian University, China. His research interests include federated learning and semi-supervised learning.

Junzhao Du received the BS, MS, and PhD degrees from Xidian University, China, in 1997, 2000, and 2008, respectively. He is currently a professor and PhD advisor at Xidian University, China. His research interests include edge AI, mobile computing, cloud computing, and IoT systems. He is the member of ACM/IEEE, senior member of CCF, and vice secretary of ACM Xi'an Chapter.

Hui Liu received the BS, MS, and PhD degrees from Xidian University, China, in 1998, 2003, and 2011, respectively. She is currently an associate professor at School of Computer Science and Technology, Xidian University, China. Her research interests include big data analysis, task scheduling, and mobile computing. She is the member of ACC, IEEE, and CCF.

Runqi Zhao received the BS degree from Xidian University, Xi'an, Shaanxi, China, in 2023. He is currently pursuing the MS degree at School of Computer Science and Technology, Xidian University, China. His research interests include unsupervised learning and semi-supervised learning.