applied sciences

MDPI

Article

FedRDS: Federated Learning on Non-IID Data via Regularization and Data Sharing

Yankai Lv [1,2,†], Haiyan Ding [1,2], Hao Wu [1,2,*,†], Yiji Zhao [3] and Lei Zhang [4]

[1] School of Information Science and Engineering, Yunnan University, Kunming 650091, China; yklv@mail.ynu.edu.cn (Y.L.); teidhy@163.com (H.D.)
[2] Key Lab of Intelligent Systems and Computing of Yunnan Province, Yunnan University, Kunming 650091, China
[3] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China; yijizhao@bjtu.edu.cn
[4] School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210024, China; leizhang@njnu.edu.cn
* Correspondence: haowu@ynu.edu.cn
† These authors contributed equally to this work.

Abstract: Federated learning (FL) is an emerging decentralized machine learning framework enabling private global model training by collaboratively leveraging local client data without transferring it centrally. Unlike traditional distributed optimization, FL trains the model at the local client and then aggregates it at the server. While this approach reduces communication costs, the local datasets of different clients are non-Independent and Identically Distributed (non-IID), which may make the local model inconsistent. The present study suggests a FL algorithm that leverages regularization and data sharing (FedRDS). The local loss function is adapted by introducing a regularization term in each round of training so that the local model will gradually move closer to the global model. However, when the client data distribution gap becomes large, adding regularization items will increase the degree of client drift. Based on this, we used a data-sharing method in which a portion of server data is taken out as a shared dataset during the initialization. We then evenly distributed these data to each client to mitigate the problem of client drift by reducing the difference in client data distribution. Analysis of experimental outcomes indicates that FedRDS surpasses some known FL methods in various image classification tasks, enhancing both communication efficacy and accuracy.

Keywords: federated learning; non-IID data; regularization; data sharing; machine learning

Citation: Lv, Y.; Ding, H.; Wu, H.; Zhao, Y.; Zhang, L. FedRDS: Federated Learning on Non-IID Data via Regularization and Data Sharing. *Appl. Sci.* **2023**, *13*, 12962. https://doi.org/10.3390/app132312962

Academic Editor: Jae Soo Yoo

Received: 20 October 2023
Revised: 19 November 2023
Accepted: 1 December 2023
Published: 4 December 2023

1. Introduction

1.1. Research Questions

With the development of Internet of Things (IoT) technology, smart devices such as cell phones and tablets are becoming more prevalent in people's lives. These portable devices collect many data, which can be used for training models and improving the efficiency of many applications. However, privacy concerns have made users increasingly reluctant to upload their sensitive data. As a result, Federated Learning (FL) [1,2] has already grown a decentralized learning framework and gained more attention in recent years. FL allows clients to download models from a server, train them locally with private data, and subsequently the server recycles these training parameters, which mitigates privacy vulnerabilities that may arise from directly uploading sensitive data. Among the earliest emerging FL frameworks is the Federated Average Algorithm (FedAvg) [3], which facilitates distributed model training for multiple users by employing cluster servers to aggregate locally trained models into a global model that converges over several iterations. However, FedAvg is less accurate when faced with non-Independent and Identically Distributed (non-IID) data. Clients have different preferences and sampling spaces, leading to

differences in data collection and resulting in data heterogeneity. The presence of heterogeneity among clients can cause variations in their objective functions and optimization directions, leading to drift in local updates and slower convergence speeds. This difference in data distribution is a crucial factor responsible for the inherent equilibrium, minimizing the loss of locally trained experience and reducing globally trained experience loss.

1.2. Aims

The conventional FL approach exhibits poor performance on data heterogeneity and lacks personalization for local tasks or datasets. These issues are further exacerbated via heterogeneous local data distribution, hampering clients' participation in the FL process. Personalized federated learning (such as customized loss function [4], multi-task learning [5,6], meta-learning [7,8], data augmentation [9], etc.) has become a new solution to these problems. Among them, regularization of local loss has been a popular personalized method in recent years. Overfitting often occurs in training models and regularization is a standard method to deal with this problem. The application of regularization in FL limits the influence of local updates, thus improving the convergence stability with the algorithm accuracy of the global training model. The traditional regularization method is easy to implement, but when there is a significant difference in the degree of dispersion of client data, the client drift will become more profound [10]. For example, FedProx [4] gradually approximates the local optimal target to the global optimal target by adding proximal terms. However, the presence of proximal terms hinders the local target from moving globally. During the training process, parameter bias will gradually accumulate and amplify, which is the main reason for the client drift phenomenon. The original intention of the data-sharing strategy [11,12] is to improve model accuracy by shortening the distribution separation of both local and global data. However, we found that while partitioning the shared dataset, each client has a portion of the same dataset, which reduces the data distribution gap between clients. Therefore, the combination of regularization and data sharing can shorten the untoward effect of regularization while retaining the advantages of regularization, greatly accelerating convergence efficiency and improving accuracy.

1.3. Objectives

Based on the findings above, We developed a FL algorithm based on regularization and data sharing (FedRDS) to alleviate client drift issues caused by skewed label distribution. FedRDS addresses these issues by using a combination of regularization and data-sharing strategies. Specifically, FedRDS incorporates a regularization term into the FedAvg local update process, allowing the local optimal targets to closely approximate the global optimal targets and minimize the impact of client drift on the global target. Additionally, FedRDS uses data sharing to relieve the unfavorable factors of heterogeneous data distribution among clients. The proposed method provides improved the performance over existing solutions by combining regularization and data sharing. Moreover, FedRDS employs the widely used Stochastic Gradient Descent (SGD) optimization technique, which enhances the overall stability of the optimization process in heterogeneous network environments.

Our work has made the following contributions:

- We design an FL algorithm FedRDS for non-IID data, which restricts the update of local models to approach the global model via regularization and data sharing gradually, as well as narrows the gap between client data distribution to relieve the negative influence of client drift.
- We conduct extensive experiments on the MNIST, Fashion-MNIST, SVHN, and CIFAR-10 datasets. From the experimental results, it can be seen that the FedRDS outperforms other algorithms in terms of efficiency.

The other parts of this paper are organized as follows: Section 2 elaborates on related work; Section 3 outlines the FedRDS algorithm and data-sharing principles; Section 4 consists of analyses based on experiments and concluding remarks; and Section 5 offers the conclusion.

2. Related Works

FL has become a wide-ranging topic across machine learning [13], cybersecurity [14], industrial IoT [15], medicine intelligence [16], etc. FedAvg performs weighted parameter averaging to update parameters from multiple clients, effectively reducing communication rounds and improving the efficiency of joint learning. However, the research in [11] indicates that the difference in client weight caused by non-IID data is the root cause of FedAvg's performance degradation. Some work attempts to reduce differences in client updates to accelerate convergence speed. FedProx [4] adds proximal regularization to the local model to overcome system heterogeneity, thus enhancing stability. Gao et al. [17] suggest that tackling the client drift problem can make the FL model converge faster. To reduce local drift on the global target, they propose the FedDC algorithm, which utilizes drift decoupling correction. Horvath et al. [18] proposed FedFa: the aggregation weights are assigned to each client at each round of global iteration based on the accuracy of client training and the number of times they are selected, as a way to reduce the aggregation caused by non-IID model performance degradation. Zhang et al. [19] proposed a stop-and-weight balancing method to balance the non-dropper updates in each iteration, which effectively controls the weight dispersion. FedBN suggests a novel aggregation approach that mitigates feature drift of non-IID data by setting the local BN out of sync with the global parameters [20]. Li et al. [21] proposed model contrastive learning (MOON) to enhance the algorithm performance by introducing a contrastive loss function to accelerate local update speed and improve the accuracy of the FL algorithm. Jeong et al. proposed a FAug approach based on FL to correct the non-IID training dataset and again reduced the communication overhead via knowledge distillation (FD), while also achieving a high accuracy rate [9].

Xiong et al. [22] proposed 2DP-FL to enhance the performance of privacy issues in practical applications, adding noise to achieve differential privacy. Li et al. [23] emphasized research on fairness and robustness under non-IID and developed a scalable solver to achieve fairness and robustness. Orlandi et al. [24] used entropy calculation based on FedAvg and proposed the FedAvg-BE algorithm, which reduced the data processing workload. Another method, pFedMe [25], breaks down personal model optimization from global model learning via the Moreau envelope function, so pFedMe can update the global model as FedAvg while also optimizing personalized models in parallel based on the local data distribution of each client, thereby improving the model performance. Smith et al. [5] introduced multi-task learning in FL and proposed the MOCHA algorithm, which allows the local training models to maintain the same structure and use an alternate optimization algorithm [26] to resolve the optimal value. Table 1 lists the comparison of related methods.

Table 1. Comparison of related methods.

Method	Author	Year	Characteristics	Neural Networks	Personalization	Data Change	Effect
FedAvg	Mc et al. [3]	2017	SGD	CNN	N	N	Reduce communication rounds by 10 to 100×
FedProx	Li et al. [4]	2020	Proximal term	ANN	Y	N	Improved testing accuracy by 20%
FedDC	Gao et al. [17]	2022	Local drift variable	FCN/CNN	N	Y	Improve performance and accelerate convergence
FedFa	Horvath et al. [18]	2021	Ordered dropout	ResNet18/ CNN/RNN	Y	Y	Performance improvement

Table 1. *Cont.*

Method	Author	Year	Characteristics	Neural Networks	Personalization	Data Change	Effect
FedBN	Li et al. [20]	2021	Local batch normalization	AlexNet/CNN	Y	Y	Faster convergence rate
MOON	Li et al. [21]	2021	Contrastive learning	CNN/ ResNet-50	Y	N	-
FAug	Jeong et al. [9]	2021	Data augmentation	-	Y	Y	Low communication delay
FedCS	Nishio et al. [27]	2019	Resource constraints	CNN	N	N	Shorter training time
pFedMe	Dinh et al. [25]	2020	Moreau envelopes	MLR/DNN	Y	Y	Accelerate convergence
2DP-FL	Xiong et al. [22]	2022	Differential privacy	-	N	Y	-
Ditto	Li et al. [23]	2020	Multi-task learning	Linear SVM/CNN	Y	N	-
Energy-aware Dynamic	Sun et al. [28]	2019	Dynamic scheduling	MLP	N	Y	Improve accuracy by 9.8%
Flexible Sparse Method	Shi et al. [29]	2021	Sparsification	-	N	N	-
FedAvg-BE	Orlandi et al. [24]	2023	Border entropy evaluation	CNN	N	N	Reduce data execution time
EPPDA	Song et al. [30]	2023	Privacy-preserving data aggregation	-	Y	Y	Tolerate failing devices
FedPAD	Pei et al. [31]	2022	Network traffic detection	LSTM	Y	N	Better detection performance
DTEI	Yang et al. [32]	2023	Deep reinforcement learning	CNN	N	N	Avoid the straggler effect and clustering mechanism
FedCLIP	Lu et al. [33]	2023	Attention-based adapter	AlexNet	Y	N	Reduces computational and communication costs

Some works consider the impact of network and communication resources of clients. Nishio et al. [27] proposed the FedCS protocol to continuously add new clients during the training process, enabling them to deliver high-performance learning models in a very short time continuously. Wang et al. [34] considered computation heterogeneity as the bottleneck of FL and therefore used two near-optimal algorithms to schedule the IID vs. non-IID data workloads. The collaborative learning classification method in the IAS decision system plays an important role in resource management and performance optimization in heterogeneous network environments [35].

Moreover, several works have investigated the problems associated with skewed label distribution. In the realm of transfer learning, joint distribution adaptation (JDA) [36] is a technique that seeks to adjust marginal and conditional distributions in order to enhance the feature representations as follows: CO-ALignment (COAL) [37], a generalized domain adaptive framework for co-alignment, computes source prototypes by learning similarity-based classifiers and uses a minimum, maximum entropy algorithm, moving it to the target domain by using estimated target label distributions, training the classifier using class-balanced self-training methods, aligning label distributions in the context of feature shifting, and then merging feature and label distribution alignments into an end-to-end deep learning framework; Selective Adversarial Network (SAN) [38] is designed to

maintain consistency source and target data distribution. It performs well in facilitating the positive migration of relevant data while mitigating the negative migration of irrelevant data by utilizing an end-to-end framework. Training classifiers on datasets with long-tailed data and imbalanced class distributions can be a formidable challenge, as they are often biased towards the more frequent classes [39–41].

In contrast to these studies, we focus on the non-IID problem of label distribution skew in FL and propose FedRDS to compare the representations learned using different models.

3. Proposed Method

We have summarized the notations used in the proposed method and placed them in Appendix A.1.

3.1. Problem Formulation

Figure 1 displays a summarized outline of the FedRDS algorithm. At the beginning, the server chooses specific clients to join the FedRDS round and issues work request information (Step 1). Upon accepting the request, the client sends an acknowledgment to the server (Step 2). Subsequently, the server transmits the pre-trained model, the global model, and a shared data fraction α to the client (Step 3). Then, the client uses its private data and shared data fraction α for training (Step 4). Following the completion of local training, after the model is trained on the client, it will be uploaded to the server (Step 5). Finally, utilizing the FedRDS algorithm, the server aggregates the local models uploaded by the client to form a global model (Step 6). The server saves the model and the next FL round is opened.

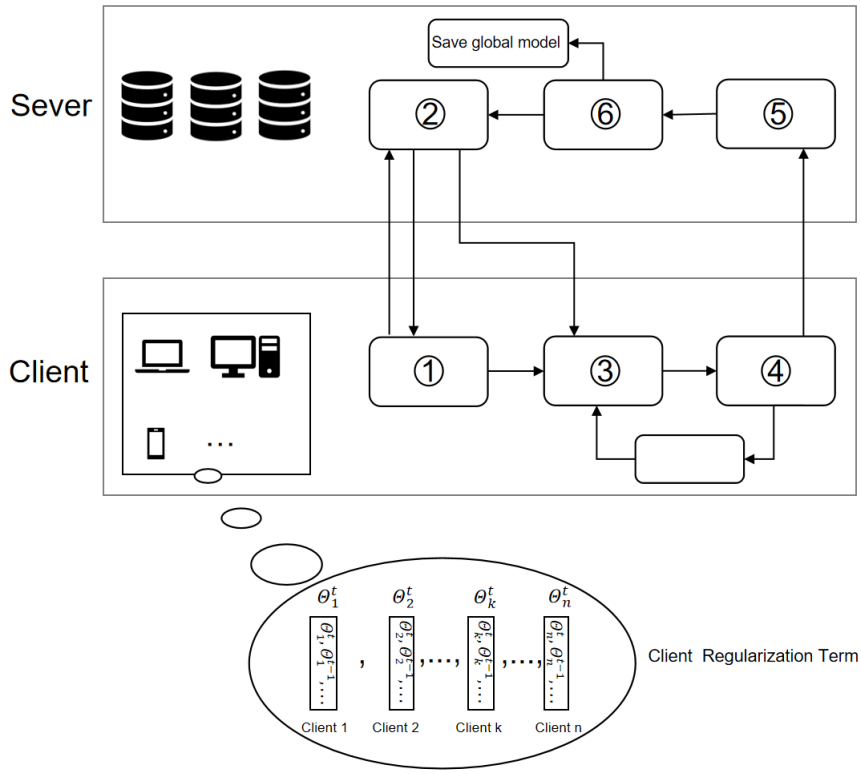


Figure 1. The proposed framework of FedRDS.

Given the compact space $\mathcal{X}$, the class space $\mathcal{Y} = \{i | i = 1 : I\}$. The data points $\{x, y\}$ follow the distribution p over $\mathcal{X} \times \mathcal{Y}$. The function $f : \mathcal{X} \to \mathcal{S}$ maps x to the probabilistic simplex $\mathcal{S}$, in which $\mathcal{S} = \{z | \sum_{i=1}^{I} z_i = 1, z_i \geq 0, \forall i \in [I]\}$ with f_i denoting the probability of

the *i*th class. f parameterized on the hypothesis category Θ. In FL, our goal is usually to optimize the following functions:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \mathcal{L}(\Theta) = \sum_{k=1}^{K} \frac{n_k}{n} l_k(\Theta), \tag{1}$$

where K is the number of total clients, n_k is the number of samples on client k, n is the total number of samples on all clients, also known as $\sum_k \frac{n_k}{n} = 1$, where l_k is the loss function of client k, and Θ is the set of global model parameters. In this paper, we consider $\mathcal{L}(\Theta)$ as non-convex.

In traditional FL algorithms, all participating clients perform the same amount of local work. This approach would be appropriate if the data were evenly distributed across each client. In many cases, however, each client may also differ in storage, computation, and communication capabilities due to differences in hardware, power, and network connectivity. At this point, the client training by FL creates the client drift problem (shown in Figure 2, which leads to the client subproblem optimal solution being inconsistent with the global objective [42]. Although many previous studies solved the non-IID data problem by adding regularization, they did not discuss regularization terms when the data distribution gap is enormous.

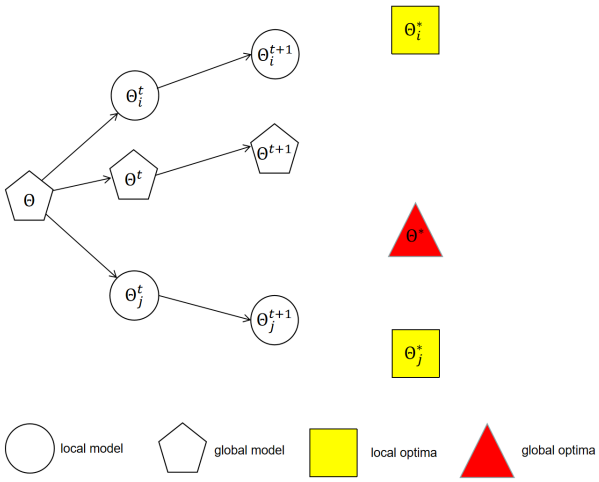


Figure 2. Client drift under the non-IID setting.

3.2. Federal Regularization

To obtain more benefits from the computational resources of the edge devices and increase the amount of light computation to reduce the performance divergence between different local clients caused by non-IID, the dynamic regularization strategy is employed to enable the local loss function to converge to the global model gradually. During the training process, the selected client $k \in S_r$ uses the untrained local model Θ and the global model Θ^t sent from the server to regularize the loss function. Formally, we solve the following optimization problem:

$$\Theta_k^{t+1} = \underset{\Theta}{\operatorname{argmin}}\, l_k(\Theta) + \frac{\sigma}{2} \|\Theta - \Theta^t\|^2 \tag{2}$$

Take the first derivative of Equation (2) to obtain the following equation:

$$\nabla l_k(\Theta_k^{t+1}) + \sigma(\Theta_k^t - \Theta^t) = 0. \tag{3}$$

In the t-th round of training, the value of hyperparameter σ in the client regularization term $\frac{\sigma}{2}\|\Theta - \Theta^t\|^2$ is calculated based on the similarity between the initial local model Θ_k^t of client k and the global model Θ^t sent by the server, as shown in the Equation (4):

$$\sigma_k^t = exp(sim(\Theta_k^t, \Theta^t)) \tag{4}$$

Among them, $sim(\Theta_k^t, \Theta^t)$ represents the cosine similarity between Θ_k^t and Θ^t.
The global objective function needs to satisfy the following formula,

$$\nabla\mathcal{L}(\Theta) \triangleq \frac{1}{K}\sum_{k=1}^{K}\nabla l_k(\Theta) = \sum_{k=1}^{K}\mathbb{E}_{(\mathrm{x},y)}\nabla\mathcal{L}(\Theta;(\mathrm{x},y)) = 0. \tag{5}$$

We assume that Θ_k^* is the optimal solution of the local loss function l_k satisfying $\nabla l_k(\Theta_k^*) = 0$, Θ^* is the optimal solution of $\mathcal{L}(\Theta)$ in Equation (1), and $\Theta^* = \frac{1}{K}\sum_{k=1}^{K}\Theta_k^*$ in FL. But, in the non-IID setting with data distribution $\mathrm{p}_i(\mathrm{x},y) \neq \mathrm{p}_j(\mathrm{x},y)$, there is $\Theta_i^* \neq \Theta_j^*$, which leads to $\Theta_k^* \neq \Theta^*$. Different client updates towards the local subproblem optimal solution Θ_k^*, causing Θ^* to be inconsistent with the theoretical value, which leads to a decrease in global model accuracy and slow convergence. In Equation (2), we incorporate a regularization term into the initial loss function to slightly modify the local loss function and create a dynamic link between the global and local models that gradually aligns the local target with the global target, thereby minimizing the influence of the client drift.

3.3. Data Sharing

Although the inclusion of regular terms in the calculation of weights can alleviate the issue of client drift resulting from non-IID data in a heterogeneous network environment to some extent, the parameter bias gradually accumulates and amplifies during the training of the model using FL, and this is a crucial factor that contributes to the algorithm's decreased accuracy and slower convergence rate. It has been proven that shortening the dissimilarity of the local and global data distribution allows the local target to gradually approach the global target, quantified via the EMD (Earth Mover's Distance) [11]. The same conclusion holds even when regularization is applied in parallel. Therefore, we can introduce additional strategies to further improve the efficiency and effectiveness of FL algorithms under non-IID conditions.

To this end, we instantiate $l_k(\Theta)$ with the cross-entropy loss for further explanation:

$$l_k(\Theta) = \sum_{i=1}^{I}\mathrm{p}(y=i)\mathbb{E}_{\mathrm{x}|y=i}[\log f_i(\mathrm{x},y)]. \tag{6}$$

If the generalization error is ignored and the total training loss is directly optimized, the problem turns:

$$\underset{\Theta}{\operatorname{argmin}}\sum_{i=1}^{I}\mathrm{p}(y=i)\mathbb{E}_{\mathrm{x}|y=i}[\log f_i(\mathrm{x},\Theta)]. \tag{7}$$

Using the SGD optimization algorithm to calculate the value of Θ, the weight of the central server after the t-th update will be:

$$\begin{aligned}\Theta^t = \Theta^{t-1} &- \eta\sum_{i=1}^{I}\mathrm{p}(y=i)\nabla_\Theta\mathbb{E}_{\mathrm{x}|y=i}[\log f_i(\mathrm{x},\Theta^{t-1})] \\ &- \sigma(\Theta^t - \Theta^{t-1}).\end{aligned} \tag{8}$$

Generally, there are K clients, and a separate SGD optimization is performed locally at each client. Correspondingly, the weights of client k after the t-th round of updates will be:

$$\Theta_k^t = \Theta_k^{t-1} - \eta \sum_{i=1}^{I} \mathrm{p}(y=i) \nabla_\Theta \mathbb{E}_{\mathrm{x}|y=i}[\log f_i(\mathrm{x}, \Theta_k^{t-1})] - \sigma(\Theta_k^t - \Theta^{t-1}). \tag{9}$$

Suppose that synchronization is performed every T optimization round at the central server; thus, the global model weight after T synchronizations will be:

$$\Theta_f^{mT} = \sum_{k=1}^{K} \frac{n_k}{n} \Theta_k^{mT}. \tag{10}$$

Denote p_k as the data distribution of client k and suppose that $\nabla_\Theta E_{\mathrm{x}|y=i} \log f_i(\mathrm{x}, \Theta)$ is $\lambda - Lipschitz$ for each class $i \in [I]$, where the extent of client drift (a.k.a the weight variance) after m-th synchronization for each class i follows the inequality (Equation (10)) every T steps, according to [11]:

$$\begin{aligned} \|\Theta_f^{mT} - \Theta_c^{mT}\| \leq & \sum_{k=1}^{K} \frac{n_k (a_k)^T}{n} \underbrace{\|\Theta_f^{(m-1)T} - \Theta_c^{(m-1)T}\|}_{(1)} \\ & + \eta \sum_{k=1}^{K} n_k \underbrace{\sum_{i=1}^{I} \|\mathrm{p}_k(y=i) - \mathrm{p}(y=i)\|}_{(2)} \\ & + \sum_{j=1}^{T-1} (a_k)^j g_{max}(\Theta_c^{mT-1-k}) \end{aligned} \tag{11}$$

where

$$\begin{aligned} & g_{max}(\Theta) = max_{i=1}^{I} \|\nabla_\Theta \mathbb{E}_{\mathrm{x}|y=i} \log f_i(\mathrm{x}, \Theta)\|, \text{and} \\ & a_k = 1 + \sigma + \eta \sum_{i=1}^{I} \mathrm{p}_k(y=i) \lambda_{\mathrm{x}|y=i}. \end{aligned} \tag{12}$$

The weight variance after the m-th synchronization is as follows: (1) the weight variance after the $(m-1)$-th synchronization; and (2) the weight variance of the probability distance of the client k data distribution relative to the actual joint distribution.

Let all clients be initialized from the same weights, and the client drift will be mainly attributed to the component (1) and the coefficient a_k. The component (1) measures the EMD between the data distribution of client k and the collective distribution on the central server [11]. Consequently, the regularization coefficient σ plays a significant role in a_k. The larger the EMD and σ, the greater the degree of client drift. Hence, properly reducing EMD can also influence the performance of FL algorithms, even given the optimal value of σ.

We attempted to introduce the data-sharing strategy to minimize the magnitude of EMD values. Therefore, during the initialization phase for FL, we take a portion β of the global data D as the shared dataset $D'$and then distribute a portion of D' to each participating client. Each client has a portion of the same data $\alpha\beta D$, which narrows the data distribution gap between each client. Subsequently, every client employs its data and shared data to train the localized model.

Figure 3 illustrates the diagrammatic representation of data sharing. Data sharing can reduce the EMD of the clients, thus significantly improving the loss of accuracy of the FL algorithm caused by non-IID data. Additionally, we have provided the selection criteria for α and β in Appendix A.2. To help readers understand the details of FedRDS, Algorithm 1 showcases the fundamental steps of our approach, which we present below.

Algorithm 1: FedRDS

Input: K, T, σ, α, β, p_k, $k \in S_r$;
Output: T-round global model Θ^T;
Initialize global model and global variables of loss function;
Remove part β from the global data as a shared dataset;
Send part α of the shared dataset to the client;
for *t=1:T* **do**
  Select the subset of clients to be involved: $S_r \in S$ with a probability of p_k;
  The server sends Θ^t to all selected clients;
  for $k \in S_r$ **do**
    $\Theta_k^{t+1} = l_k(\Theta) + \frac{\sigma}{2}\|\Theta - \Theta^t\|^2$;
    Sends Θ^t to the server;
  end
  Server aggregation Θ as $\Theta^{t+1} = \frac{1}{k}\sum_{k \in S_r} \Theta_k^{t+1}$;
end
Return Θ^T

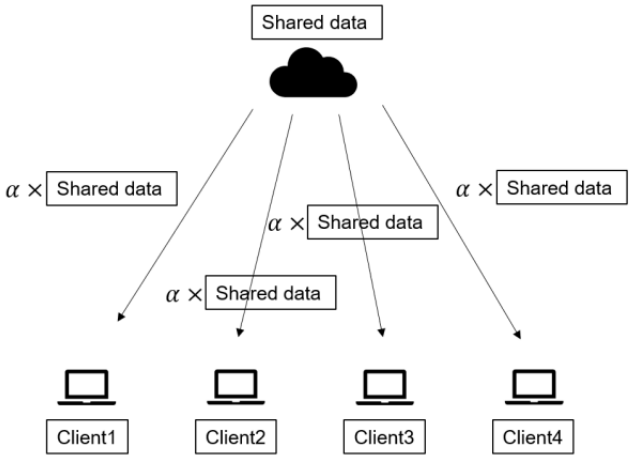


Figure 3. Data-sharing strategy.

4. Experiments and Analysis

4.1. Datasets and Settings

The datasets for the experiments include **MNIST**, **Fashion-MNIST**, **SVHN**, and **CIFAR-10**, and the network models in the experiments are all convolutional neural networks (CNNs).

MNIST: Consists of grayscale images of handwritten digits classified into 10 categories. Consisting of 60,000 training samples and 10,000 test samples, the dataset features images that are 28×28 pixels in size. The neural network model for this dataset comprises two convolutional layers, each with a kernel size of 5×5, and the first layer has 32 channels. In contrast, the second layer has 64 channels, both featuring a max-pooling layer of 2×2. Additionally, it consists of a fully connected layer that employs 512 neurons, which are activated using the rectified linear unit (ReLU) mechanism, followed by a softmax output layer.

Fashion-MNIST: an image dataset replacing the MNIST handwritten dataset, consisting of 10 classes of clothing items with (28×28) pixel grayscale images. The Fashion-MNIST dataset is divided into training and test sets in the same manner as MNIST, comprising 60,000 training samples and 10,000 test samples. Each image in this dataset measures 28×28. The neural network architecture for this dataset incorporates two convolutional layers and one fully connected layer.

SVHN: Derived from Google Street View house numbers and consists of images comprising Arabic numerals '0–9'. It necessitates minimal data pre-processing and is comparable to the MNIST dataset, but it contains a more extensive collection of labeled data. The training set comprises 73,257 digits, while the test set comprises 26,032 digits and an additional 531,131 digits. We used the same neural network model as CIFAR-10 for this dataset.

CIFAR-10: (32 × 32) 3-channel pixel RGB images belonging to 10 categories of objects, with each image measuring 32 × 32. The dataset comprises five batches allocated for training and one batch for testing, each containing 10,000 images. For the test batch, 1000 images have been randomly drawn from each category. The remaining 50,000 images have been evenly distributed among the five training batches, resulting in around 5000 images per category. Due to the breakdown of the remaining 50,000 images into five training batches, the number of images per category may be different across batches. This leads to the possibility of some categories having more images than others. This dataset's neural network comprises two convolutional layers, two fully connected layers, and a linear transformation layer.

For dividing the dataset, we used the method in [3]. For IID data, we shuffle the data and divide it into 100 clients with 600 shards per client. For non-IID data, we partitioned the training and test data into 200 shards based on their labels and randomly allocated two shards to each client.

For the edge computing scenario, many data nodes often have the problem of non-IID data and this paper simulates the non-IID in the scenario. For the actual scenario simulated on the four datasets, we set the learning rate (η), the number of local epochs (E), batch size (B), the fraction of shared data (β), the regularization coefficient (σ), the fraction of shared data per client (α), and the fraction of clients (C) as shown in Table 2. In order to assess the effectiveness of the algorithms, the experiments were run on a Windows computer with NVIDIA GeForce RTX 3070 Laptop GPU and 16 GB of memory.

Table 2. Experimental Settings.

Dataset	η	B	C	β	α	E
MNIST	0.001	8	0.1	0.1	0.5	10
Fashion-MNIST	0.0001	14	0.1	0.1	0.5	10
SVHN	0.1	500	0.1	0.1	0.5	10
CIFAR-10	0.01	10	0.1	0.1	0.5	10

4.2. Baselines

We compare FedRDS with the following three state-of-the-art models: **FedAVG** [3] randomly selects a portion of the clients, samples their parameters to compute local model updates, and aggregates these updates to generate a global model update. The non-sampled clients then substitute their local models for the updated global model. **FedProx** [4] allows the local models with incomplete training; namely, all local models do not need to be finely tuned. The loss of the local model is constrained by adding the proximal term. **MOON** [21] introduces a loss function that utilizes contrastive learning to incentivize improving the accuracy of local models, ultimately enhancing the performance of the global model.

4.3. Comparison of Average Accuracy

Figure 4 presents the four models' comparative experimental results on the MNIST, Fashion MNIST, SVHN, and CIFAR-10 datasets (FedAvg, FedProx, MOON, and FedRDS). The figure displays the accuracy trend and training loss under non-IID nodes.

In Figure 4a, after 126 rounds, the FedRDS algorithm gradually converges and its model accuracy reaches 98.23%. Figure 4b shows that the accuracy of FedRDS differs from other methods during the initial phase of model training and reaches a smooth accuracy curve after 165 rounds. The accuracy of the model can ultimately reach 88.74%. In Figure 4c,d, the FedRDS accuracy curve showed significant fluctuations in the early stages,

and before 36 rounds, the loss curve experienced severe tremors. However, the two curves gradually stabilized as the number of communication rounds increased. The accuracy of FedRDS after 200 rounds of communication reached 84.95%, MOON achieved an 83.98% accuracy, FedProx achieved an 83.32% accuracy, and FedAvg achieved an 82.62% accuracy, while FedRDS achieved an accuracy of 55.91% for CIFAR-10 and FedAvg only achieved 53.03% (2.88% lower than FedRDS). FedProx and MOON showed moderate accuracy and some convergence in all four datasets. However, their performance is relatively weak in non-IID situations and further optimization and improvement are needed.

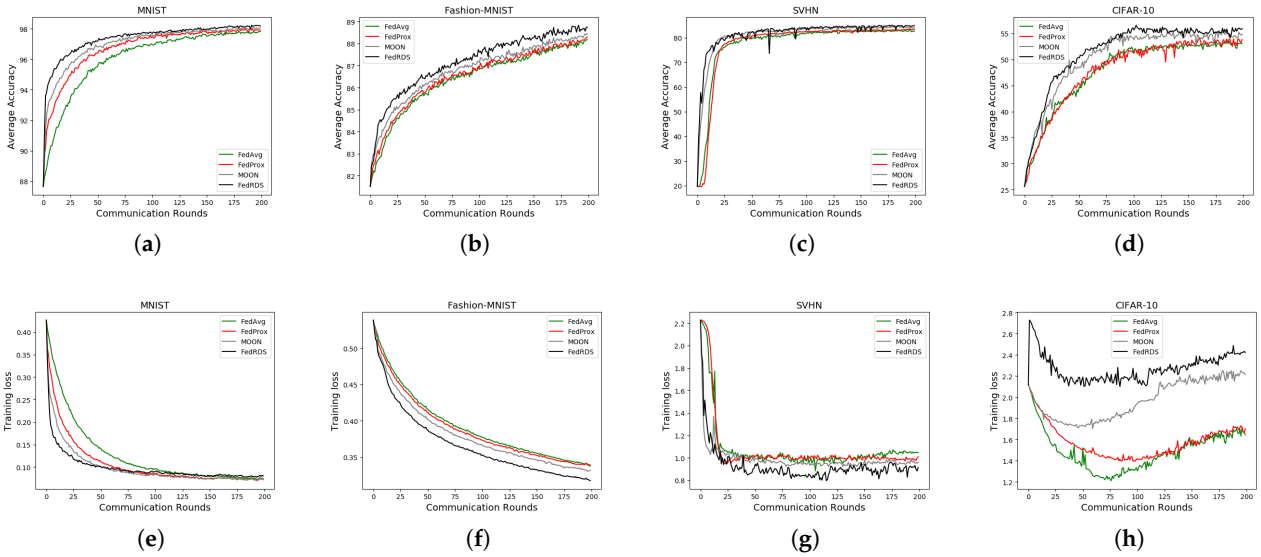


Figure 4. Average accuracy and training loss in the non-IID scenario. (**a**) Average accuracy on MNIST. (**b**) Average accuracy on Fashion-MNIST. (**c**) Average accuracy on SVHN. (**d**) Average accuracy on CIFAR-10. (**e**) Training loss on MNIST. (**f**) Training loss on Fashion-MNIST. (**g**) Training loss on SVHN. (**h**) Training loss on CIFAR-10.

The data in Figure 4 indicates that FedRDS has faster convergence and higher model accuracy compared to other baselines in most non-IID situations. This is due to the use of dynamically balanced weights and allocated FedRDS algorithm node weights being reasonable, which improves the model's performance. The FedRDS algorithm gradually converges after training because the FedRDS algorithm avoids the overfitting of local models, thus improving the generalization ability of models. The accuracy curve in the experiment fluctuates more because the data-sharing strategy promotes data flow between all clients and allows them to use the existing data better. The data-sharing strategy aids in minimizing the gap between local and global data and provides more categories of training data. The loss curve indicates that most algorithms have a smooth decreasing curve, and the loss curve of FedRDS decreases faster and lower than other algorithms. This result indicates that FedRDS has better training efficiency when processing non-IID data and can use data from each client more effectively. Therefore, compared to FedAvg, FedProx, and MOON, FedRDS has faster and higher accuracy improvements. We provide a detailed comparison of the accuracy of four methods in Table 3.

Table 3. Performance comparisons with standard FL algorithms. Columns correspond to three scenes. The average accuracy of the 200 rounds is reported.

	Method			
Dataset	**FedAvg**	**FedProx**	**MOON**	**FedRDS**
MNIST	97.87%	97.94%	98.07%	**98.23**%
Fashion-MNIST	88.18%	88.29%	88.45%	**88.74**%
SVHN	82.62%	83.32%	83.98%	**84.95**%
CIFAR-10	53.03%	53.85%	54.88%	**55.91**%

The best results are marked in **bold**.

4.4. Effect of the Number of Local Epochs

Furthermore, we examined how varying numbers of local epochs affected the model's accuracy, with local epochs ranging from 10, 20, 40, to 80, as shown in Figure 5. When E = 10, the client executes fewer local epochs and is forced to stop before full training, thus affecting the accuracy of all methods. However, FedRDS performs exceptionally well on the CIFAR-10 dataset, achieving higher accuracy than other methods. This is because the optimized local loss function can use the global model as additional knowledge and is robust to insufficient local periods. As E gradually increases, the accuracy of most methods has improved due to more thorough training on local datasets. When E = 20, FedRDS gradually widens the gap with other methods. However, on MNIST, the accuracy of all methods is very close because MNIST has fewer data categories and can train quickly regardless of the size of E. FedRDS can rely on a small number of local epochs E to dynamically correct the local model via the regularization method to achieve a faster model performance. Therefore, increasing E has little benefit for local models, and the increase in accuracy tends to be gradual. When E = 40, FedRDS shows performance degradation on the Fashion MNIST and CIFAR-10 datasets. Excessive local epochs result in the overfitting of local models, exacerbating the issue of client drift caused by non-IID data. Therefore, selecting an appropriate number of local epochs can help the regularization method perform better. When E = 80, the accuracy of FedRDS on the SVHN dataset is even lower than that of MOON. However, other methods maintain a trend of continuously improving accuracy, as they rely on more local epochs to achieve the same accuracy.

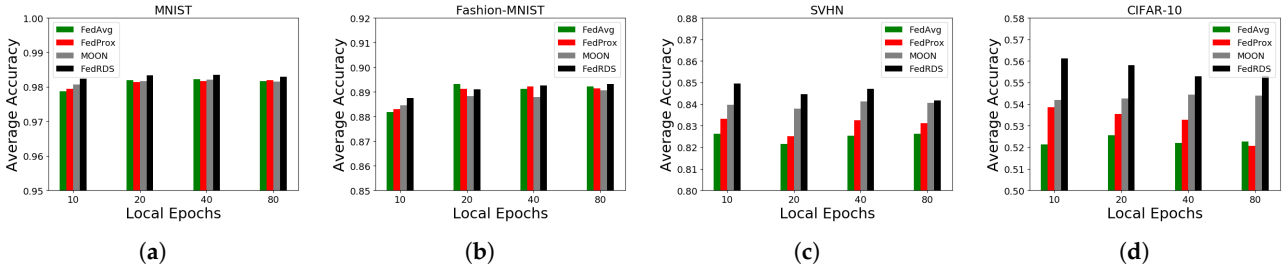


Figure 5. Effect of different numbers of local epochs E on the average accuracy. (**a**) Average accuracy on MNIST. (**b**) Average accuracy on Fashion-MNIST. (**c**) Average accuracy on SVHN. (**d**) Average accuracy on CIFAR-10.

The above indicates that FedRDS can perform better than other methods with relatively sufficient local training. FedAvg, FedProx, and MOON require more local epochs than FedRDS for the same accuracy. In the actual application scenario of FL, the computing performance and connection stability of the client are uncertain and more local epochs mean longer connection times and more excellent dropout opportunities, which is a considerable challenge for large FL networks. FedRDS does not require too many local epochs, which is more advantageous in extreme cases of insufficient local epochs and can address dropout issues in FL due to network bandwidth and other reasons. However, FedRDS may also suffer from overfitting caused by too many local epochs. Hence, selecting the proper number of local epochs is crucial in FL.

4.5. Comparison of Communication Efficiency

In our experimental setup, for MNIST, Fashion-MNIST, and SVHN, we established the baseline accuracy using FedAvg's performance after 100 training rounds. Table 4 displays the count of communication rounds necessary for all methods with default parameters to reach the benchmark accuracy. As seen from the table, FedRDS demands notably fewer communication iterations when compared to FedAvg and FedProx, indicating that regularization methods can offer efficient communication besides improving the generalization performance. Furthermore, in most cases, FedProx follows FedAvg throughout the training process. We used a small μ parameter ($\mu = 0.01$), so the proximal term in FedProx seems to have a minor effect on the training procedure. The findings illustrated in Figure 4 indicate that FedRDS converges faster than the other two methods in the initial stage due to its higher stability with non-IID data, enabling it to reach the benchmark accuracy sooner and ensure efficient communication.

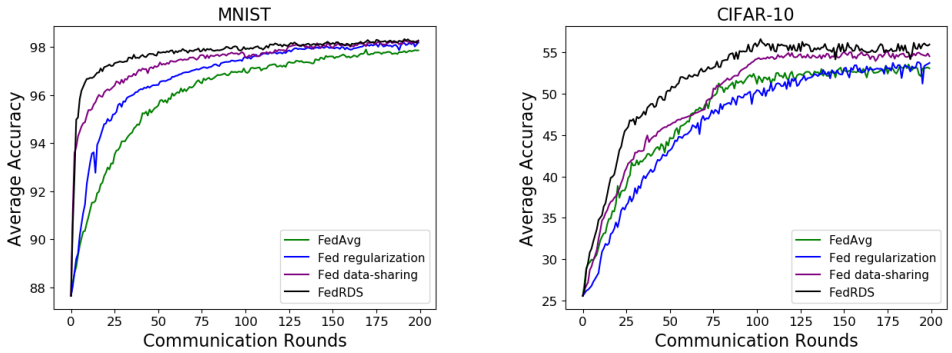
The FedRDS algorithm has proven to be highly efficient in communication across all the datasets used in our experiments. It is worth highlighting that the algorithm demonstrated remarkable efficiency when dealing with the difficult-to-classify and semantically complex SVHN dataset, achieving a 2.78-fold enhancement. Incorporating a larger shared dataset size for CIFAR-10 proved beneficial in reducing the gap between the client and server data distributions. It facilitated faster target accuracy attainment during the initial training stages of FedRDS. Moreover, dynamically adjusting the shared dataset size as per actual requirements is possible without any additional communication costs, and this strategy only needs to be executed once during initialization.

Table 4. Communication efficiency comparison under the default setting.

FL Method	Dataset							
	MNIST		**Fashion-MNIST**		**SVHN**		**CIFAR-10**	
	Rounds	***Speedup***	***Rounds***	***Speedup***	***Rounds***	***Speedup***	***Rounds***	***Speedup***
FedAvg	100	1×	100	1×	100	1×	100	1×
FedProx	74	1.35×	95	1.05×	79	1.26×	92	1.09×
MOON	61	1.64×	83	1.2×	42	2.38×	64	1.56×
FedRDS	40	2.5×	67	1.49×	36	2.78×	57	1.75×

4.6. Ablation Experiment

Figure 6 demonstrated that using either the regularization term alone or data sharing was more effective than FedAvg, and both approaches mitigated the non-IID problem while improving model accuracy. Combining dynamic regularization and data sharing can give the FL algorithm the best of both worlds. Also, the results confirm our speculation that both dynamic regularization and data-sharing strategies can mutually promote each other.



(**a**) MNIST (**b**) CIFAR-10

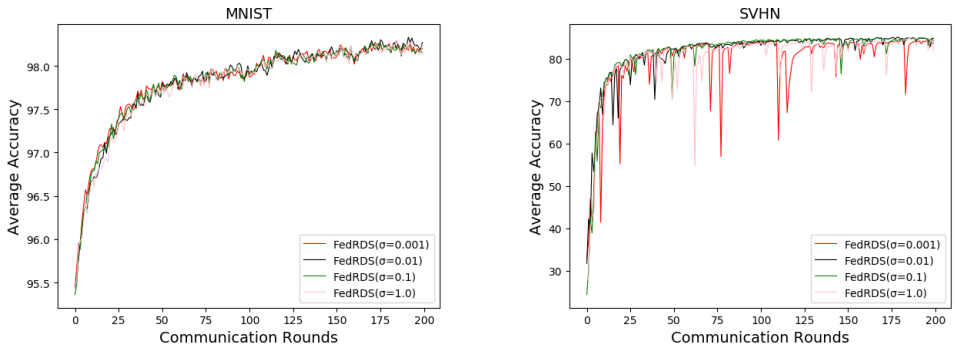
Figure 6. Ablation Experiment.

We further investigate the specific performance of Fed regularization and Fed data sharing. The results show that the accuracy of Fed regularization in Figure 6b is low in the early training period due to the high computational complexity of introducing regularization terms and the more complex data classes of CIFAR-10 compared to MNIST. But, as the model continues to learn, Fed regularization eventually outperforms the benchmark model FedAvg. Fed data sharing is achieved by sharing server-side data rather than client-side data. Fed data sharing performs better than Fed regularization, which indicates that narrowing the data distribution gap significantly impacts the experimental results more than regular terms. Nevertheless, if a is set too large, it will make the distribution of client-side data too similar to that of server-side data, which elevates the possibility of compromising privacy. The experimental results show that using the regularization term alone or data sharing can significantly alleviate the non-IID problem and boost the model's robustness. However, when updating the weights, the combined approach of combining regularization terms and data sharing makes the local loss function more complex, and FedRDS takes longer to converge.

As a regularization and data sharing strategy for FL, FedRDS offers a unique solution to some challenges that arise when dealing with non-IID data. By combining a regularization term with the data-sharing strategy, the algorithm effectively mitigates the issue of uneven model distribution and overfitting, resulting in better model generalization. The results of the experiments indicate that FedRDS outperforms the benchmark models. Our findings suggest that dynamic regularization and the data-sharing strategy have significant practical applications for FL-related problems, including reducing the effects of overfitting and heterogeneous model distributions as well as improving generalization and accuracy in non-IID data settings. Therefore, dynamic regularization and data sharing is of great significance and value and has broad application prospects for solving problems in FL networks, which will help improve the efficiency, accuracy, and robustness of various FL applications in such diverse fields as healthcare, finance, and manufacturing.

4.7. Effect of the Regularization Coefficient

How to choose the proper regularization coefficient σ is a crucial problem. If the σ is too large, it will make the client and global model too similar and the fitting of local data insufficient, increasing the degree of client drift. Choosing σ as a relatively small value may result in insufficient constraints on local updates. Therefore, we explored the regularization coefficient on the MNIST and SVHN datasets. The impact on the algorithm's accuracy is shown in Figure 7. Throughout all trials, we fine tune the optimal variable σ from a restricted pool of candidates set {0.001, 0.01, 0.1, 1.0}. When σ = 0.001, the regularization term has insufficient generalization ability to the full according model, so the algorithm's accuracy is relatively low. With the increase in σ, the advantage of the regular term gradually becomes noticeable and the accuracy reaches the highest at σ = 0.01. But, as the σ value further increases, the accuracy begins to decrease. The reason for this is that if σ is too large, it can worsen the problem of client drift, which will negatively impact the model's overall performance. While increasing σ can improve the model's generalization ability, this benefit is not significant enough to offset the adverse effects of client drift. Therefore, the experimental results confirm our previous conjecture that the optimal σ values for both datasets in Figure 7 are 0.01.



(**a**) MNIST (**b**) SVHN

Figure 7. Coefficient Experiment.

5. Conclusions

With edge devices' growing computing power and storage capacity, the risk of data privacy breaches also increases. FL is expected to become a prominent framework in the future. Even with the non-IID characteristic of the data, FL may be unstable and less efficient during the training period. We suggest the FedRDS algorithm as a solution for dealing with the client drift problems caused by non-IID in federated networks. The algorithm utilizes a regularization term when constructing the local loss function to gradually align the local model with the global model and relies on the data-sharing strategy to diminish the gap between local and global data distributions, thus significantly improving accuracy. Extensive experiments were conducted on different datasets and the results demonstrate the feasibility of our solution in handling non-IID data and the advantages of FedRDS over some existing state-of-the-art methods concerning accuracy and communication effectiveness.

6. Future Work

Our future efforts will focus on enhancing the theoretical research and practical application of the FedRDS algorithm, particularly by optimizing its performance for use in real-world scenarios and improving its overall efficiency.

Author Contributions: Methodology, Y.L.; Validation, H.D., Y.Z. and L.Z.; Formal analysis, H.W.; Data curation, Y.L.; Writing – original draft, Y.L.; Writing – review & editing, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (62362069), partially supported by the Yunnan Provincial Foundation for Leaders of Disciplines in Science and Technology (202005AC160005), and Yunnan High-Level Talent Training Support Plan: Young Top Talent Special Project (YNWR-QNBJ-2019-188).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://yann.lecun.com/exdb/mnist/, https://github.com/zalandoresearch/fashion-mnist, http://ufldl.stanford.edu/housenumbers/, http://www.cs.toronto.edu/~kriz/cifar.html (accessed on 19 October 2023).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study.

Appendix A

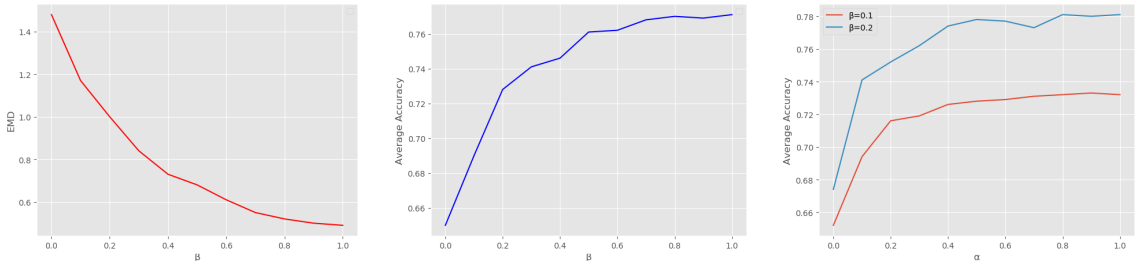
Appendix A.1

Table A1. Notations.

Notations	Description
$\Theta^*, \Theta(\Theta_k), \Theta^t(\Theta_k^t)$	Global or Local model parameters (at iteration t)
$i, j, t, k, \mathrm{p}, c, f$	Indices for clients, iteration, client, Probability distribution, client of SGD, client of FedRDS
$\mathcal{X}, \mathcal{Y}, \mathcal{S}, K$	Compact space, Class space, Probabilistic simplex, Total number of clients
S, S_r	Collection of clients, Subset of clients
D, D'	The global data, The shared data
$m, n(n_k)$	synchronization, The amount of data (at client k)
T, I	The number of steps before synchronization, Total number of classes
$\mathbb{E}$	Expectation
$\mathcal{L}, l_k$	Global loss function, Local loss function (at client k)
$f(\mathrm{x}, y)$	The neural network based function to map the input x $\in \mathcal{X}$ to the output $y \in \mathcal{Y}$
α, β, η	The fraction of shared data per client, The fraction of shared data, The learning rate
B, C, E	Batch size, The fraction of cilents, The number of local epochs
σ, μ	The regularization coefficient of FedRDS, The regularization coefficient of FedProx

Appendix A.2

We conducted relevant experiments on CIFAR-10, as shown in Figure A1. In Figure A1b, as β increases, the average accuracy can reach 78.82%, and when b = 10%, the accuracy can reach 74.22%. In the experiment, we found that as β gradually increases, the time required to train the model increases. Therefore, the size of β should be selected according to actual needs. In Figure A1c, we selected two specific β values of 10% and 20%, respectively. We can see that the accuracy of α increases rapidly from 0 to 10% in the initial stage, but after increasing to 50%, the accuracy tended to flatten out. Therefore, we should select α appropriately, which can minimize the size of the data distributed to clients as much as possible.



(**a**) (**b**) (**c**)

Figure A1. (**a**) EMD vs. β, (**b**) average accuracy vs. β, and (**c**) average accuracy vs. the distribution function α.

References

1. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 12:1–12:19. [CrossRef]
2. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
4. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
5. Smith, V.; Chiang, C.; Sanjabi, M.; Talwalkar, A. Federated Multi-Task Learning. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 4424–4434.

6. Mills, J.; Hu, J.; Min, G. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 630–641. [CrossRef]
7. Fallah, A.; Mokhtari, A.; Ozdaglar, A.E. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, Online, 6–12 December 2020.
8. Yang, L.; Huang, J.; Lin, W.; Cao, J. Personalized Federated Learning on Non-IID Data via Group-Based Meta-Learning. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–20. [CrossRef]
9. Jeong, E.; Oh, S.; Park, J.; Kim, H.; Bennis, M.; Kim, S. Hiding in the Crowd: Federated Data Augmentation for On-Device Learning. *IEEE Intell. Syst.* **2021**, *36*, 80–87. [CrossRef]
10. Tan, A.Z.; Yu, H.; Cui, L.; Yang, Q. Towards personalized federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 9587–9603. [CrossRef] [PubMed]
11. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated Learning with Non-IID Data. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; pp. 1–13.
12. Shao, J.; Sun, Y.; Li, S.; Zhang, J. DReS-FL: Dropout-Resilient Secure Federated Learning for Non-IID Clients via Secret Data Sharing. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 10533–10545.
13. Korkmaz, C.; Kocas, H.E.; Uysal, A.; Masry, A.; Ozkasap, O.; Akgun, B. Chain fl: Decentralized federated machine learning via blockchain. In Proceedings of the 2020 Second International Conference on Blockchain Computing and Applications (BCCA), Antalya, Turkey, 2–5 November 2020; pp. 140–146.
14. Alazab, M.; Priya, R.M.S.; Parimala, M.; Maddikunta, P.K.R.; Gadekallu, T.R.; Pham, Q. Federated Learning for Cybersecurity: Concepts, Challenges, and Future Directions. *IEEE Trans. Ind. Inform.* **2022**, *18*, 3501–3509. [CrossRef]
15. Zhou, J.; Lu, Q.; Dai, W.; Herrera-Viedma, E. Guest Editorial: Federated Learning for Industrial IoT in Industry 4.0. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8438–8441. [CrossRef]
16. Dayan, I.; Roth, H.R.; Zhong, A.; Harouni, A.; Gentili, A.; Abidin, A.Z.; Liu, A.; Costa, A.B.; Wood, B.J.; Tsai, C.S.; et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nat. Med.* **2021**, *27*, 1735–1743. [CrossRef] [PubMed]
17. Gao, L.; Fu, H.; Li, L.; Chen, Y.; Xu, M.; Xu, C. FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10102–10111.
18. Horváth, S.; Laskaridis, S.; Almeida, M.; Leontiadis, I.; Venieris, S.I.; Lane, N.D. FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, Online, 6–14 December 2021; pp. 12876–12889.
19. Zhang, Y.; Duan, L.; Cheung, N. Accelerating Federated Learning on Non-IID Data Against Stragglers. In Proceedings of the IEEE International Conference on Sensing, Communication, and Networking, Online , 20–23 September 2022; pp. 43–48.
20. Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; Dou, Q. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In Proceedings of the 9th International Conference on Learning Representations, Online, 3–7 May 2021.
21. Li, Q.; He, B.; Song, D. Model-contrastive federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10713–10722.
22. Xiong, Z.; Cai, Z.; Takabi, D.; Li, W. Privacy Threat and Defense for Federated Learning With Non-i.i.d. Data in AIoT. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1310–1321. [CrossRef]
23. Li, T.; Hu, S.; Beirami, A.; Smith, V. Ditto: Fair and Robust Federated Learning Through Personalization. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020.
24. Orlandi, F.C.; Dos Anjos, J.C.S.; Leithardt, V.R.Q.; De Paz Santana, J.F.; Geyer, C.F.R. Entropy to Mitigate Non-IID Data Problem on Federated Learning for the Edge Intelligence Environment. *IEEE Access* **2023**, *11*, 78845–78857. [CrossRef]
25. Dinh, C.T.; Tran, N.H.; Nguyen, T.D. Personalized Federated Learning with Moreau Envelopes. *arXiv* **2020**, arXiv:abs/2006.08848.
26. Bezdek, J.C.; Hathaway, R.J. Some Notes on Alternating Optimization. In Proceedings of the Advances in Soft Computing—AFSS 2002: 2002 AFSS International Conference on Fuzzy Systems, Calcutta, India, 3–6 February 2002.
27. Nishio, T.; Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In Proceedings of the 2019 IEEE International Conference on Communications, Changchun, China, 11–13 August 2019; pp. 1–7.
28. Sun, Y.; Zhou, S.; Gündüz, D. Energy-Aware Analog Aggregation for Federated Learning with Redundant Data. In Proceedings of the ICC 2020—IEEE International Conference on Communications (ICC), Online, 7–11 June 2019; pp. 1–7.
29. Shi, D.; Li, L.; Chen, R.; Prakash, P.; Pan, M.; Fang, Y. Toward Energy-Efficient Federated Learning Over 5G+ Mobile Devices. *IEEE Wirel. Commun.* **2021**, *29*, 44–51. [CrossRef]
30. Song, J.; Wang, W.; Gadekallu, T.R.; Cao, J.; Liu, Y. EPPDA: An Efficient Privacy-Preserving Data Aggregation Federated Learning Scheme. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 3047–3057. [CrossRef]
31. Pei, J.; Zhong, K.; Jan, M.A.; Li, J. Personalized federated learning framework for network traffic anomaly detection. *Comput. Netw.* **2022**, *209*, 108906. [CrossRef]
32. Yang, W.; Xiang, W.; Yang, Y.; Cheng, P. Optimizing Federated Learning With Deep Reinforcement Learning for Digital Twin Empowered Industrial IoT. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1884–1893. [CrossRef]

33. Lu, W.; Hu, X.; Wang, J.; Xie, X. FedCLIP: Fast Generalization and Personalization for CLIP in Federated Learning. *IEEE Data Eng. Bull.* **2023**, *46*, 52–66.
34. Wang, C.; Yang, Y.; Zhou, P. Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 394–410. [CrossRef]
35. Anjos, J.C.S.D.; Matteussi, K.J.; Orlandi, F.C.; Barbosa, J.L.V.; Silva, J.S.; Bittencourt, L.F.; Geyer, C.F.R. A Survey on Collaborative Learning for Intelligent Autonomous Systems. *ACM Comput. Surv.* **2023**, *56*, 1–37. [CrossRef]
36. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer Feature Learning with Joint Distribution Adaptation. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, 1–8 December 2013; pp. 2200–2207.
37. Tan, S.; Peng, X.; Saenko, K. Generalized Domain Adaptation with Covariate and Label Shift CO-ALignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7165–7174.
38. Cao, Z.; Long, M.; Wang, J.; Jordan, M.I. Partial Transfer Learning With Selective Adversarial Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2724–2732.
39. Cao, K.; Wei, C.; Gaidon, A.; Aréchiga, N.; Ma, T. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 1565–1576.
40. Hayat, M.; Khan, S.; Zamir, S.W.; Shen, J.; Shao, L. Gaussian Affinity for Max-Margin Class Imbalanced Learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic fof Korea, 27 October–2 November 2019; pp. 6468–6478. [CrossRef]
41. Jamal, M.A.; Brown, M.; Yang, M.; Wang, L.; Gong, B. Rethinking Class-Balanced Methods for Long-Tailed Visual Recognition From a Domain Adaptation Perspective. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7607–7616.
42. Acar, D.A.E.; Zhao, Y.; Navarro, R.M.; Mattina, M.; Whatmough, P.N.; Saligrama, V. Federated Learning Based on Dynamic Regularization. In Proceedings of the 9th International Conference on Learning Representations, Online, 3–7 May 2021.