This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Communication-Efficient Federated Data Augmentation on Non-IID Data

Hui Wen[1], Yue Wu[1], Jingjing Li[12], Hancong Duan[1]
[1] University of Electronic Science and Technology of China
[2] Institute of Electronic and Information Engineering of UESTC in Guangdong
lijin117@yeah.net

Abstract

Federated learning (FL) is an attractive distributed machine learning framework due to the property of privacy preservation. The implementation of FL encounters the challenge of the Non-Independent and Identically Distributed (Non-IID) data across devices. This work focuses on mitigating the impact of Non-IID datasets in wireless communications. To achieve this goal, we propose a generative models-based federated data augmentation strategy (FedDA) with privacy preservation and communication efficiency. In FedDA, the Conditional AutoEncoder (CVAE) is adopted to generate the missing samples on Non-IID datasets. The Knowledge Distillation Mechanism is introduced to achieve Federated learning, through which knowledge is shared, rather than model parameters or gradients. The knowledge is designed based on the hidden-layer features to reduce the communication overhead and protect raw data privacy. Meanwhile, to generate cross-class samples that are easy to classify, the latent variables in CVAE are constrained and the attention mechanism is introduced. Extensive experiments are conducted on Fashion-MNIST datasets and CIFAR-10 with different data distributions. The results show that FedDA can improve the model accuracy by up to 8% *while reducing the communication overhead by up to* $2\times$*, compared to classic baselines with highly Non-IID data.*

1. Introduction

Federated learning (FL) [10,18,31] has been proposed as an attractive distributed machine learning framework with privacy preservation and has been applied to many real-world applications, e.g., smart healthcare [17], automated industrial processes [23], and vehicular services [20]. In FL, instead of having to share private raw data, devices keep their data locally and only share information of the locally trained model. As a distributed system [2], FL faces the challenge of Non-Independent and Identically Distributed (Non-IID) data due to different local environments and characteristics of devices [16, 29]. In wireless communications, the limited link resource is another challenge [14, 33].

To handle Non-IID data, some approaches [18, 22, 27] have shared parameters based on the weights or gradients of target models. Unfortunately, due to limited bandwidth in wireless communications, it is difficult to share so many parameters between devices and the server [5]. Other approaches [7, 24, 31, 36] have concentrated on creating a central dataset in the server, through which local missing data on devices can be supplemented. The data in the central dataset is composed of local raw samples uploaded by each device [7, 24, 36], or is generated by generators trained on local raw samples [31]. While sharing raw data or generator parameters are good solutions for the challenge of Non-IID data, they compromise the privacy of the raw data.

Inspired by the research [7, 22, 31, 36], we design a federated data augmentation strategy, FedDA, to address the Non-IID challenges in wireless communication. The purpose of FedDA is to create an IID foundation in each device with the cooperation of the central server. The key characteristics of FedDA are concerned with efficient communication between devices and the server, and the protection of data privacy.

To achieve data augmentation, the generative model [4, 25] is leveraged in each device. Considering the storage space and processing capacity of the wireless device, Conditional AutoEncoder (CVAE) [25] is chosen as the generator. CVAE in each device is trained under FL settings to utilize the data from all devices. To implement Federated learning, the knowledge distillation mechanism is introduced. In this mechanism, one neural network can learn some useful information through the knowledge from other neural networks [1, 6, 15, 30]. In FedDA, the knowledge is designed based on the typical hidden-layer features for each class and shared between devices and the server. To extract general knowledge for each category the attention mechanism [35] is introduced. Since knowledge takes up fewer bits compared to model weights or gradients, the sharing of knowledge consumes fewer communication resources. Without sharing generators' weights or generated samples,

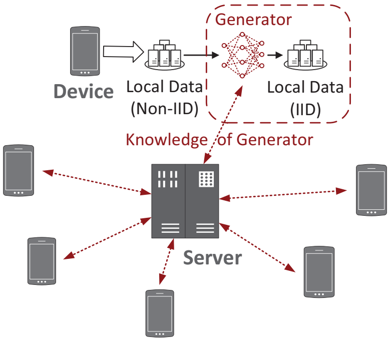

Figure 1. The overall procedure of FedDA. The dotted bidirectional arrows represent the upload and download process between devices and the server. The hollow arrow indicates the expansion of the details.

the privacy of the raw data is protected. In addition, to generate cross-class samples that can be easily distinguished, the CVAE is constrained by the sampling of latent variables.

The procedure of FedDA in wireless communications is briefly described below. An initialized CVAE in each device is trained by the local Non-IID dataset with the constrained mean value of the hidden variables. Meanwhile, attention-based knowledge is calculated and utilized to restrain the federated training process. The main procedure of FedDA is demonstrated in Fig. 1.

The main contributions are summarized as follows:

- We propose a Federated data augmentation strategy, FedDA to address the Non-IID challenges in FL. The FedDA can provide an IID foundation for target tasks in each device based on the local Non-IID dataset. The separation of data augmentation from the target task addresses the Non-IID data challenge while helping the target task to achieve higher accuracy.
- Federated data augmentation strategy is achieved by generative models with privacy-preserving. CVAEs are utilized as generative models to fit the characteristics of wireless devices. Meanwhile, to generate cross-class samples that can be easily distinguished, the latent variables of CVAE are constrained and the attention mechanism is introduced.
- Knowledge distillation mechanism is employed to achieve Federated data augmentation in wireless communications. The knowledge is designed based on the per-label typical hidden-layer features to reduce the communication overhead and protect raw data privacy.
- The experiments show that FedDA can improve the model accuracy by up to 8% while reducing the communication overhead by up to $2\times$, compared to classic baselines with highly Non-IID data.

2. Related Work

Some research has focused on mitigating the impact of Non-IID datasets in wireless communications. Sattler *et al.* [22] have designed a sparse ternary compression FL framework for low-bitwidth communication with Non-IID data. Rothchild *et al.* [21] have compressed model gradients by a count sketch to reduce shared bits in each global round and address the Non-IID data challenge. Vahidian *et al.* [27] have proposed a model pruning approach to reduce the shared bits in each global and mitigate the accuracy degradation caused by Non-IID data. Zhao *et al.* [37] have derived a tractable upper bound to reduce the impact of non-IID data and designed a joint optimization algorithm to keep the balance between the model accuracy and the cost. Li *et al.* [13] have designed a Lottery Ticket hypothesis-based FL, in which, lottery ticket networks are designed for learning and communication.

Some research has concentrated on data augmentation for Non-IID datasets. Zhao *et al.* [36] have proposed a data-sharing strategy by creating a cloud dataset, which is a collection of raw samples uploaded by each device. The cloud dataset is downloaded by all the edge devices to supplement missing local data. Jeong *et al.* [7] have built the cloud dataset, as research [36]. Thereafter, a global generator is trained on that cloud dataset and downloaded by each device to augment the local dataset. Duan *et al.* [3] have proposed a Self-balancing framework, in which each client needs to send its label distribution information to the server. Wen *et al.* [31] have designed a generator-sharing strategy, in which, the weights of generators are shared between devices and the server. This work has paid attention to defending against Byzantine devices' attacks in classification tasks. Shin *et al.* [24] have proposed an XOR-based one-shot FL framework. The core idea is to collect other devices' encoded data samples, which are decoded only using each device's own data samples. Yoon *et al.* [34] have designed a mean augmented method by exchanging the averaged batch local data with the server.

3. FedDA

This section includes two sections: (1) We describe the system modeling and introduce our optimization problem. (2) We present the details of FedDA. The local training process in devices is defined as the local iteration. The communication iteration between devices and the server with upload and download processes is defined as the global iteration. In each global iteration, the knowledge uploaded by devices is defined as the local knowledge, and the knowledge downloaded from the server is defined as the global knowledge. The local training iteration for the generator in each device is defined as the local iteration. The main notations in sections 3, 4 are summarized in Table 1.

Table 1. Description of main notations

Notation	Description
D	Dataset
B	Batch size
x	Raw data sample
$y, y = (1, \ldots, Y)$	Label of a sample
$r, r = (1, \ldots, Y), (r \neq y)$	Label of a sample
$h, h = (1, \ldots, H)$	Index of a sample in y class
NI	Non-IID
IID	IID
$\tilde{x}$	Generated sample
$*$	Server
$n, n = (1, \ldots, N)$	Index of a device
$m, m = (1, \ldots, N), m \neq n$	Index of a device
F	Neural network function
w	Model weights
η	Local learning rate
$\mathcal{L}$	Loss function
k	Knowledge
$u, u = (0, \ldots, U)$	Index of the global iteration
$v, v = (0, \ldots, V)$	Index of the local iteration
G	Generator
Enc	Encoder of CVAE
Dec	Decoder of CVAE
z	Latent variables
μ	Mean value of latent variables
σ	Standard deviation of latent variables
ϕ	Weights of encoder
θ	Weights of decoder
A	Attention Score
KD	Process of knowledge distillation
λ	Weight of KD loss function
D_{KL}	Distance of KL-divergence
Q	Number of imbalanced classes
IR	Imbalance-ratio

3.1. Problem Statement

We assume that there is a server and N devices in the wireless network. Each device is connected directly to the server, and there is no connection between devices. Each device hosts local data processing units and the same initialized generative networks. The server collects and processes the information uploaded from the devices and returns the processed data to the devices.

In each device, there is a local and labeled dataset $D := (x^h, y)_{h=1}^{H}, y = (1, \ldots, Y)$. These data can be described by the joint probability between features x and labels y. Statistical models in each device draw examples $(x, y) \sim P(x, y)$ from the local data distribution. Compared with a dataset in centralized machine learning approaches with joint distribution $P_*(x, y)$, there are N datasets in N devices with distribution $P_n(x, y), n = (1, \ldots, N)$. To protect data privacy, FL cannot assemble datasets from individual devices into a centralized dataset D_* in the server. Therefore, we are committed to obtaining a dataset in each device with a data distribution $\hat{P}_n(x, y)$ similar to that of a centralised dataset $P_*(x, y)$.

When IID data is referenced, the distribution of datasets across devices is the same ($P_n(x, y) = P_m(x, y), n = (1, \ldots, N), m = (1, \ldots, N), m \neq n$). In wireless communications, considering the scenario of Non-IID data, the data tend to deviate from being identically distributed across devices, $P_n(x, y) \neq P_m(x, y)$. To analyze the data distribution, we rewrite $P_n(x, y)$ as $P_n(x|y)P_n(y)$. We focus on the label distribution skew, where the marginal distributions $P_n(y)$ and $P_m(y)$ may vary across devices, even if $P_n(x|y) = P_m(x|y)$ is the same.

For FedDA, the Non-IID dataset D^{NI} is acted as the input in each device. We are committed to adjusting the joint distribution in devices to reduce the differences between $P_n(x, y)$, $P_m(x, y)$ and the hypothetical $P_*(x, y)$. Therefore, the generator is introduced to build an IID dataset D^{IID} in each device. The data distribution can be calculated as $P(x, y) = P(y|x)P(x)$. Through the trained generators, the marginal distributions $P_n(x)$ and $P_m(x)$ are expected to be as equal as possible. Compared with generators of vanilla VAEs, the CVAEs are employed to adjust the conditional probabilities so as to make $P_n(x|y)$ and $P_m(x|y)$ as equal as possible.

With the above analysis, the design goals can be described in the following aspects.

- **Privacy**. To protect data privacy in each device, it is essential to design a framework that keeps each device's data locally without sharing. Meanwhile, we are dedicated to designing a federated training scheme for generators that do not share model weights and generated samples.
- **Low communication overhead**. The total communication overhead of FL relates to the shared bits in each global iteration and the number of global iterations. We are devoted to minimizing the total communication overhead by reducing the shared bits in each iteration and decreasing the number of iterations.
- **High Quality**. FedDA needs to ensure that the quality of the generated samples is within the desired range on Non-IID data.

3.2. Federated Data Augmentation Strategy

The FedDA addresses the Non-IID data challenge by building an IID data foundation. The generative models are applied to achieve data augmentation. To prevent the privacy of local data, some noise samples, initialized

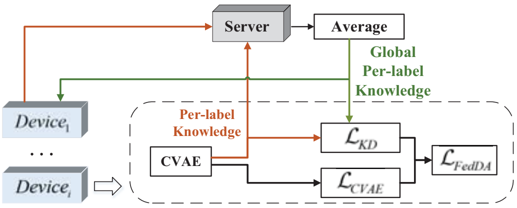

Figure 2. The training process of the knowledge distillation in FedDA. The server and two devices are selected as examples. The hollow arrow indicates the expansion of the details. The ↑ represents the upload and the ↓ represents the download.

by Laplace noise [32], are mixed into each device's local dataset.

Generative models in FedDA are chosen based on the characteristics of wireless communications. Considering the insufficient computing power and storage spaces in common devices, VAE-based generators [9] are more suitable than GAN-based generators [4]. As the images generated by the baseline VAE have no labelling information, the generated samples cannot supplement the Non-IID dataset by class. Therefore, Conditional variational autoencoder (CVAE) [25], the variant of VAE with label information, is introduced due to the ability to generate samples with classification information.

CVAE is a directed probabilistic graphical model of the marginal likelihood for data samples [25]. In CVAE, an encoder and a decoder are two components. The encoder is to infer the posterior $q(\mathbf{z}|\mathbf{x},\mathbf{y})$, where $\mathbf{z}$ is the latent variables, $\mathbf{x}$ is the input data sample, and y is employed to denote "condition", which contains the label information, and encoded as one-hot vectors. After modeling the posterior, the decoder models the conditional likelihood of $\mathbf{x}$ with the latent variables $\mathbf{z}$ and the "condition" by $p(\mathbf{x}|\mathbf{z},\mathbf{y})$. The decoder outputs the generated samples with label information. The loss function of CVAE is:

$$\mathcal{L}_{CVAE} = \\ -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{y})) + E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}(\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})), \tag{1}$$

which represents the variational lower bound of marginal likelihood $\log p(\mathbf{x}|\mathbf{z},\mathbf{y})$.

There are two constraints during the training process of CVAE: (1) The sampling process for latent variables is constrained. (2) The data reconstruction is regularized by knowledge distillation in the FL way.

3.2.1 Sampling of Latent Variables

For CVAE, the mean value of latent variables for each class is constrained so as to cross-class samples that can be easily distinguished. The mean value of latent variables for each class is assigned a unique value to distinguish the peaks of the Gaussian curves from the different categories. The latent distribution is constrain by predefining the prior $p(\mathbf{x}|\mathbf{z},\mathbf{y})$ to Gaussian $\mathcal{N}(\boldsymbol{\mu},\sigma^2)$, where the value of $\boldsymbol{\mu}$ satisfies the 3σ Rule. Notably, the mean value for the same class is set to be the same across devices.

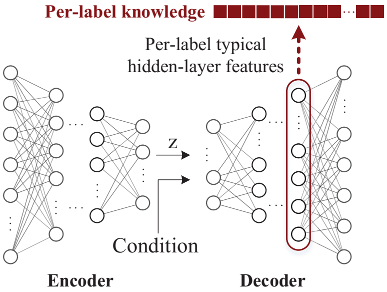

Figure 3. The shared information of FedDA in each global iteration. Each device shares the local knowledge based on the typical hidden-layer features of each class.

3.2.2 Reconstruction Process of VAE

To adjust the conditional probability $P(x|y)$ across devices and leverage sample information from all devices, the CVAE is regularized in a federated manner. The Knowledge Distillation Mechanism [6,30] is introduced to minimize the difference between the local knowledge $\mathbf{k}_{G,n}$ from devices and the global knowledge $\mathbf{k}_{G,*}$ from the server.

The distance between the local and global knowledge is measured by the Kullback-Leibler divergence and reduced through the following loss function:

$$\mathcal{L}_{KD} = \sum_{y=1}^{Y} D_{KL}(\mathbf{k}^y_{G,n}, \mathbf{k}^y_{G,*}). \tag{2}$$

Therefore, the loss function of CVAE is written as:

$$\mathcal{L}_{FedDA} = \min_{\theta,\phi}(\mathcal{L}_{CVAE} + \lambda\mathcal{L}_{KD}), \tag{3}$$

where the parameter λ is related to the proportion of the two parts of the loss function. The training process for CVAE is shown in Fig. 2.

Knowledge should be set up to suit the FL settings and wireless communication environment. The local knowledge $\mathbf{k}_{G,n}$ is designed based on the hidden-layers of the decoder to protect privacy. Per-label typical hidden-layer features from the fully connected layer before the deconvolutional layer are exploited as knowledge. Since the knowledge takes up fewer bits compared to model weights or gradients, sharing knowledge consumes fewer communication

Algorithm 1 Federated Generation

```
Require: The value of $z$ is set to be the same across devices.
procedure SERVER EXECUTES:
    for $u$-th global iteration do:
        for label $y = 1 \dots Y$ do
            for device $n = 1 \dots N$ in parallel do
                $k_{G,n}^{y,u+1} \leftarrow$ DeviceUpdate($k_{G,*}^{y,u}$)
            $k_{G,*}^{y,u+1} \leftarrow \frac{1}{N} \sum_{n=1}^{N} k_{G,n}^{y,u+1}$
function DEVICEUPDATE($k_{G,*}^{y}$)
    for $v$-th local iteration do: $B \leftarrow D^{NI}$
        for sample $x \in B$ do
            $z \leftarrow Enc(x, y) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma)$
            $\tilde{x} \leftarrow Dec(z, y)$
            $k_{G,n}^{y} \leftarrow A^{y}$
            $cnt_{G,n}^{y} \leftarrow cnt_{G,n}^{y} + 1$
    for label $y = 1 \dots Y$ do
        $k_{G,n}^{y} \leftarrow F_{G,n}^{y} / cnt_{G,n}^{y}$
        calculate $\mathcal{L}_{KD}$ using Eq.(2) based on $k_{G,n}^{y}$ and $k_{G,*}^{y}$
    $g \leftarrow \nabla(\mathcal{L}_{CVAE}(\mathbf{x}, \mathbf{c}; \theta, \phi) + \mathcal{L}_{KD})$
    $\phi, \theta \leftarrow$ Update parameters using gradients $g$
    return $k_{G,n}^{y}(y = 1, \dots, Y)$ to the server
```

resources. The shared information between devices and the server is shown in Fig. 3.

To optimize the shared knowledge, the attention score [28] is introduced. The attention score is a measurement, introduced from the attention mechanism, which attempts to capture the correlations among different information within the same sequence. In FedDA, the attention score is applied to signify which information the decoder is most concerned with.

The formula of attention score can be described as mapping a "query" ($\mathbf{Q}$) and "key" ($\mathbf{K}$) to an output. The definitions of $\mathbf{Q}$, $\mathbf{K}$ are as follows: $\mathbf{Q} = \mathbf{XW^Q}, \mathbf{K} = \mathbf{XW^K}$. The character $\mathbf{X}$ is an input. Parameters concerning $\mathbf{W^Q}$ and $\mathbf{W^K}$ are trained. The matrix of outputs is calculated as Eq. (4):

$$\tilde{\mathbf{A}} = softmax(\frac{f(\mathbf{Q}, \mathbf{K})}{\sqrt{\mathbf{d_K}}}). \tag{4}$$

The $\tilde{\mathbf{A}}$ is defined as the attention score to explore the relationship of similarity between $\mathbf{Q}$ and $\mathbf{K}$. The character of $\mathbf{d_k}$ is the vector to control variance. As the attention score is acted as the knowledge, the attention score is simplified to a nonparametric version. The element values of $\mathbf{d_K}$ are set to 1. The both characters of "query" ($\mathbf{Q}$) and "key" ($\mathbf{K}$) represent the vectors, defined by features $\mathbf{Z}$ from the fully connected layer before the deconvolutional layer: $\mathbf{Q} = \mathbf{Z}, \mathbf{K} = \mathbf{Z}$.

To reduce the communication overhead, the attention score is calculated by the Hadamard product [8]. Hadamard products were used to extract the significant and general information in each category. Meanwhile, the communication between devices and the server is benefited from the Hadamard products with few parameters.

The calculation of the Hadamard product-based attention score is as follows. Firstly, the unnormalized attention score $\hat{\mathbf{A}}^y$ with "condition" y is computed by the Hadamard product $\hat{\mathbf{A}}^y = [\hat{a_i}^y] = \mathbf{z}^y \circ \mathbf{z}^y \in \mathbb{R}^{1 \times H}$, where the $\mathbf{z}$ is the vector of typical feature with $1 \times H$ dimensions with "condition" y. Then, the attention score is normalized by the softmax function to make it comparable between different devices. The normalized form $\mathbf{A}^y = [a_i^y] \in \mathbb{R}^{1 \times H}$ is as follows:

$$a_i^y = \frac{e^{a_i^y}}{\sum\limits_{j=1}^{H} e^{a_j^y}} = \frac{e^{z_i^y \circ z_i^y}}{\sum\limits_{j=1}^{H} e^{z_j^y \circ z_j^y}}. \tag{5}$$

Finally, the attention scores with condition y is averaged: $\mathbf{A}^y = \frac{1}{H^y} \sum_{h=1}^{H^y} \mathbf{A}_h$. The mini-batch is set to unify the number of input samples.

The local knowledge $\mathbf{k}_{G,n}^y$ is defined by per-label averaged normalized attention score $\mathbf{A}^y$. The global knowledge $\mathbf{k}_{G,*}^y$ from the server is calculated by averaging the local knowledge from participating devices: $\mathbf{k}_{G,*}^y = \frac{1}{N} \sum_{n=1}^{N} \mathbf{k}_{G,n}^y$.

3.2.3 Main Procedure of FedDA

The main procedure of FedDA is shown in the following process. A global generator, CVAE, is initialized. In each device, the global CVAE is acquired and treated as the local generator. The mean value of the hidden variables for each class is assigned a unique number which is set to be the same across devices. From the first round of local iteration, the local generator is trained by the local data, and the local knowledge $\mathbf{k}_{G,n}$ based on the attention score is calculated. After the v-th local iteration, generators start the first global iteration in a federated manner. In devices, the local knowledge is uploaded to the server. The index of the local iteration is recounted from 0-th, and the index of the global iteration is increased by one. In the server, the global knowledge $\mathbf{k}_{G,*}$ is calculated and downloaded by each device. In each device, the global knowledge act as guidance in each local iteration to update the local generator. After the v-th local iteration, the local knowledge is uploaded to the server for the next global iteration. The algorithm for the main federated training procedure of the generator in FedDA is shown in **Algorithm 1**.

Table 2. Average accuracy of CNN-4 on Fashion-MNIST.

Dataset		Standalone	FedAvg	XorMixFL	FAug-Standalone	**FedDA**-Standalone	**FedDA**-FedAvg
Class: 5	$IR = 3$	$82.76_{\pm 0.51}$	$88.32_{\pm 0.78}$	$91.29_{\pm 0.47}$	$90.87_{\pm 0.54}$	$90.62_{\pm 0.61}$	$91.55_{\pm 0.68}$
	$IR = 6$	$80.35_{\pm 0.68}$	$87.16_{\pm 0.81}$	$90.86_{\pm 0.59}$	$89.48_{\pm 0.73}$	$90.31_{\pm 0.69}$	$90.27_{\pm 0.69}$
	$IR = 12$	$77.74_{\pm 0.96}$	$85.75_{\pm 0.72}$	$89.47_{\pm 0.63}$	$89.05_{\pm 0.78}$	$88.26_{\pm 0.61}$	$90.16_{\pm 0.77}$
Class: 8	$IR = 3$	$81.16_{\pm 0.65}$	$86.91_{\pm 0.86}$	$89.08_{\pm 0.62}$	$87.22_{\pm 0.59}$	$88.35_{\pm 0.51}$	$89.48_{\pm 0.62}$
	$IR = 6$	$77.48_{\pm 0.62}$	$85.26_{\pm 0.93}$	$87.96_{\pm 0.66}$	$85.16_{\pm 0.70}$	$87.58_{\pm 0.62}$	$88.96_{\pm 0.66}$
	$IR = 12$	$74.92_{\pm 0.89}$	$82.35_{\pm 1.03}$	$84.52_{\pm 0.73}$	$84.75_{\pm 0.84}$	$85.12_{\pm 0.71}$	$87.52_{\pm 0.73}$
Class: 9	$IR = 3$	$62.34_{\pm 1.84}$	$85.12_{\pm 0.71}$	$86.26_{\pm 0.65}$	$84.75_{\pm 0.84}$	$85.37_{\pm 0.92}$	$87.03_{\pm 0.67}$
	$IR = 6$	$56.13_{\pm 2.73}$	$81.29_{\pm 0.76}$	$83.34_{\pm 0.93}$	$82.62_{\pm 0.74}$	$84.28_{\pm 1.05}$	$85.29_{\pm 0.76}$
	$IR = 12$	$48.96_{\pm 3.66}$	$72.79_{\pm 0.85}$	$75.54_{\pm 1.05}$	$74.95_{\pm 0.80}$	$76.17_{\pm 1.38}$	$80.64_{\pm 0.88}$

4. Experimental Evaluation

4.1. Dataset

The experiments are conducted on the datasets of Fashion-MNIST [12] and CIFAR-10 [11]. The standard data pre-processing strategies are adopted, including horizontal flip, padding, and random crop [26]. On each dataset, Q classes are randomly selected as imbalanced classes. To measure the degree of imbalance in each class, the Imbalance Ratio (IR) [19] is introduced. The number of IR is calculated by dividing the maximum statistic ζ by the minimum: $IR = \frac{max_i \zeta_i}{min_j \zeta_j}$. Based on hyper-parameters of Q and IR, Non-IID datasets are created across devices.

4.2. Baselines

Standalone is a local training method for classification tasks, where no information is shared across devices. **FedAvg** [18], as the vanilla algorithm, shares the weights of classifiers. **FAug** [7] shares the raw samples and weights of the generator across devices. **XorMixFL** [24] shares the embedding of local samples. We assemble data augmentation approaches (FedDA and FAug) with classifiers (Standalone and FedAvg) to obtain hybrid approaches, **FedDA-Standalone**, **FAug-Standalone**, and **FedDA-FedAvg**.

4.3. Models and Settings

CVAE is utilized as the generator. The structure of the encoder is 2 convolutional layers, each with 64, 4×4 kernels, a stride of 2, and relu. The encoder is followed by 2 fully connected layers, each with 28 units in Fashion-MNIST. The architecture of the decoder is almost the transpose of the encoder, but the output parameters are distributed by pixels. For the dataset of Fashion-MNIST, 4-layer CNN (CNN-4) is applied as the classifier. The structure of CNN is 2 convolutional layers with relu and max-pooling layers and 2 fully connected layers. For the dataset of CAIFR-10, the 11-layer VGG network (VGG-11) is simplified and acted as the classifier. The structure of VGG-11 is transformed by reducing the number of convolutional filters to $[32, 64, 128, 128, 128, 128, 128, 128]$, and the size of the hidden fully connected layers is cut down to 128.

We suppose that there are 20 devices for Fashion-MNIST, and 100 devices for CIFAR-10. All devices fully participate in each global iteration. The neural networks are trained with the mini-batch strategy. For the local training, CVAE is constrained by Adam with the batch size of 64, the local iteration of 100, and the initial local learning rate of 0.0003. Besides, the mean values of latent variables are set as $[0, 6, 12, 18, 24, 30, 36, 42, 48, 54]$, and the σ is set as 1. For the global training, CVAE is restrained with the weight of 0.01 for KD loss function. The training settings for the competitors are the same as their research [7, 18, 24]. For FAug [7] and XorMixFL [24], 4 samples in 4 classes are shared. The data augmentation processes, handled by FedDA, FAug and XorMixFL, are carried out in a federated manner to build local IID datasets. Standalone and FedAvg are employed as classifiers and trained by that built IID dataset. The Standalone is trained locally, while FedAvg is trained federally.

4.4. Accuracy of FedDA-based Classifiers

We validate the performance of these generated samples for the classification task on the Non-IID dataset. The average accuracy of classifiers is recorded in Table 2 and 3. It is apparent that FedDA-Standalone and FedDA-FedAvg can achieve good accuracy on various Non-IID data. Compared with FedAvg, FedDA-FedAvg yields higher accuracy on Non-IID data. This outcome indicates that the FedDA improves the tolerance of the classifier for Non-IID data. Compared with FedAvg, FedDA-Standalone achieves higher accuracy. This result suggests that the samples generated by FedDA offer good substitutions for shared weights. Compared with XorMixFL and FAug-Standalone, FedDA-Standalone is able to match a similar accuracy without having to share data. The higher accuracy of FedDA-Standalone validates the effectiveness of FedDA as a data enhancement algorithm to mitigate the challenges of Non-

Table 3. Average accuracy of VGG-11 on CIFAR-10.

Dataset		Standalone	FedAvg	XorMixFL	FAug-Standalone	**FedDA**-Standalone	**FedDA**-FedAvg
Class: 5	$IR = 3$	$70.08_{\pm 0.91}$	$72.36_{\pm 0.86}$	$77.23_{\pm 0.78}$	$76.18_{\pm 1.15}$	$74.04_{\pm 0.88}$	$79.97_{\pm 0.91}$
	$IR = 6$	$67.79_{\pm 1.05}$	$71.53_{\pm 1.14}$	$76.82_{\pm 0.83}$	$75.66_{\pm 1.33}$	$73.61_{\pm 0.96}$	$79.14_{\pm 0.85}$
	$IR = 12$	$63.16_{\pm 1.24}$	$71.02_{\pm 1.21}$	$76.02_{\pm 0.96}$	$73.95_{\pm 1.57}$	$72.37_{\pm 1.12}$	$78.38_{\pm 1.03}$
Class: 8	$IR = 3$	$52.82_{\pm 1.05}$	$58.71_{\pm 1.42}$	$73.46_{\pm 0.94}$	$69.74_{\pm 1.32}$	$71.24_{\pm 0.76}$	$76.45_{\pm 0.85}$
	$IR = 6$	$49.35_{\pm 1.32}$	$57.34_{\pm 1.68}$	$71.14_{\pm 1.06}$	$68.15_{\pm 1.29}$	$70.02_{\pm 1.38}$	$75.66_{\pm 0.94}$
	$IR = 12$	$43.49_{\pm 1.41}$	$55.92_{\pm 2.10}$	$69.79_{\pm 1.12}$	$67.68_{\pm 1.37}$	$69.13_{\pm 1.23}$	$74.87_{\pm 1.26}$
Class: 9	$IR = 3$	$37.76_{\pm 1.88}$	$40.72_{\pm 2.30}$	$67.89_{\pm 1.08}$	$65.75_{\pm 1.39}$	$70.27_{\pm 1.14}$	$73.15_{\pm 1.13}$
	$IR = 6$	$31.15_{\pm 2.37}$	$35.54_{\pm 2.77}$	$66.12_{\pm 1.06}$	$63.23_{\pm 1.51}$	$69.87_{\pm 0.87}$	$72.63_{\pm 1.34}$
	$IR = 12$	$28.28_{\pm 2.49}$	$32.13_{\pm 2.62}$	$64.72_{\pm 1.15}$	$61.62_{\pm 1.47}$	$66.54_{\pm 1.32}$	$71.41_{\pm 1.06}$

Table 4. Shared bits in each communication round. When the upload and download processes are different, they are listed separately in the following format: upload/download.

Framework	parameters	Fashion-MNIST	Cifar-10
FedAvg	Weights	18.62 M	27.69 M
XorMixFL	Simples/Weights	0.1 M/18.62 M	0.03 M/27.69 M
FAug-Standalone	Simples/Weights	0.1 M/47.79 M	0.03 M/47.93 M
FedAD-Standalone	Knowledge	0.02 M	0.02 M

IID data. Further analysis from Table 2 and Table 3 shows that FedDA performs well in both datasets, and can be adapted to a large-scale distributed learning paradigm.

4.5. Communication Overhead

We investigate the shared bits in each global iteration and total communication overhead when the pre-trained classifier achieves some accuracy. The model weights and attention vectors occupy equally the 32 bits, while the pixel parameter of the sample consumes 8 bits. The experiments are carried out with $IR = 12$.

The approximate data of shared bits in each global iteration are recorded in Table 4. It can be observed that the weighted bits shared by FedAvg are much larger than the bits of knowledge shared by FAug-Standalone. Compared with FAug-Standalone and XorMixFL, FedDA-Standalone share less bits.

The total shared bits in Fashion-MNIST are calculated and shown in Table 5, the analyses are made as follows. (1) For reachable accuracy, FedAvg transmits the maximum total bits. (2) Compared with FedAvg, FedDA-FedAvg not only achieves higher accuracy, but also shares fewer parameters. This result suggests that FedDA has a positive effect on the task of classifying on Non-IID data. (3) Compared with FedAvg, FedDA-Standalone achieves higher accuracy, with fewer parameters to transfer. (4) During the upload process, FAug-Standalone and XorMixFL share more bits than FedDA-Standalone. (5) For FAug-Standalone and XorMixFL, there are much more bits to download than to upload. Besides, FedDA-Standalone uploads a few more bits than FAug-Standalone and XorMixFL, as the raw (in FAug-Standalone) and the embedding of local samples (in XorMixFL) are not shared. In addition, the bits downloaded by FedDA-Standalone are much smaller than those downloaded by FAug-Standalone and XorMixFL. The reason is that FedDA-Standalone only downloads the attention-based knowledge while FAug-Standalone and XorMixFL download the bits of the models.

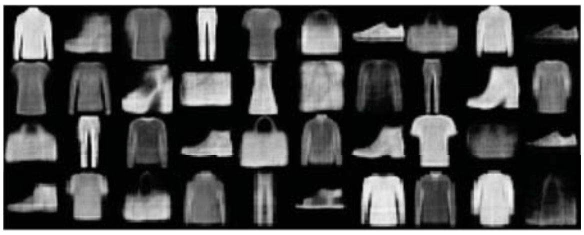

Figure 4. Visualisation of Genentated Samples by FedDA on Fashion-MNIST.

4.6. Visualisation of Generated Samples

We visualize generated samples of FedDA on Fashion-MNIST with $Q = 5, IR = 12$. The generated samples are shown in Fig. 4. It is obvious that those generated samples are of high quality and close to the real samples. Those samples can effectively supplement the missing data and create an IID data foundation for the classifier.

4.7. "Reality" of Generated Samples

We measure the "Reality" of generated samples on Fashion-MNIST with $Q = 5, IR = 12$. We trained the Standalone classifier on IID data, and they have 95.2% accuracy on the test set. This classifier is deployed on each device and is applied to evaluate whether the generated samples contain the correct classification information. We record the times when the classified label of generated sam-

Table 5. The total bits of training process in Fashion-MNIST. For FAug-Standalone and XorMixFL, due to the different information in the upload and download processes, the bits are listed separately. “U”, ‘D” and “-” are short for “Upload”, “download” and “Unattainable”.

Dataset	Accuracy	Standalone	FedAvg	XorMixFL		FAug-Standalone		**FedDA**-Standalone	**FedDA**-FedAvg
		U/D	U/D	U	D	U	D	U/D	U/D
Class: 5	86	-	-	0.01 M	18.62 M	0.01 M	47.79 M	1.53 M	410.30 M
	84	-	353.87 M	0.01 M	18.62 M	0.01 M	47.79 M	1.20 M	205.15 M
	82	-	279.37 M	0.01 M	18.62 M	0.01 M	47.79 M	0.93 M	130.55 M
Class: 8	84	-	-	0.01 M	18.62 M	0.01 M	47.79 M	1.71 M	876.55 M
	82	-	949.87 M	0.01 M	18.62 M	0.01 M	47.79 M	1.48 M	578.15 M
	80	-	875.37 M	0.01 M	18.62 M	0.01 M	47.79 M	1.13 M	447.60 M
Class: 9	82	-	-	0.01 M	18.62 M	0.01 M	47.79 M	1.91 M	1920.94 M
	78	-	-	0.01 M	18.62 M	0.01 M	47.79 M	1.58 M	1566.59 M
	74	-	-	0.01 M	18.62 M	0.01 M	47.79 M	1.25 M	1249.54 M

Table 6. “Reality” Score of CNN-4 on Non-IID data.

Dataset		FAug	XorMixFL	**FedDA**
Class: 5	$IR = 3$	89.72	87.37	86.54
	$IR = 6$	86.36	84.21	83.86
	$IR = 12$	82.58	80.14	79.27
Class: 8	$IR = 3$	86.31	81.15	83.46
	$IR = 6$	82.82	77.45	80.25
	$IR = 12$	77.43	73.14	76.17
Class: 9	$IR = 3$	79.73	75.12	77.46
	$IR = 6$	76.21	72.34	74.65
	$IR = 12$	72.67	68.53	70.39

Table 7. Shared bits in each global iteration.

Framework	Class:5	Class:8	Class:9
FedDA-V1-Standalone	87.08	83.96	73.41
FedDA-V2-Standalone	85.74	81.58	70.33
FedDA-Standalone	88.26	85.12	76.17

ples is the same as the conditional label, and describe it as the “Reality” time. We calculate the “Reality” score based on the “Reality” time and the total generated samples. The higher the score, the more realistic the generated samples are.

The “Reality” score is recorded in Table 6. Compared with XorMixFL, FedDA achieve higher score when the Non-IID data becomes more extreme. Although the “Reality” score of FedDA is lower than that of FAug, FedDA avoids sharing raw data.

4.8. Ablation Study

We conduct a detailed ablation study to analyze the effectiveness of FedDA’s components. We transform FedDA into the following variants by deleting or replacing particular constraints and compare classification performance with FedDA. (1) FedDA-V1: without distribution alignment for the mean value of latent variables. (2) FedDA-V2: replacing the attention-based knowledge as Per-label hidden-layer feature-based knowledge. The experiments are conducted on Fashion-MNIST with $IR = 12$. The results are presented in Table 7. It can be seen that the introduction of distributed alignment and attention-based knowledge is beneficial for classification tasks of Non-IID data. By comparing the three approaches, we observe that the application of attention-based knowledge contributed most to the performance improvement.

5. Conclusion

In this paper, a generative models-based federated data augmentation strategy (FedDA) is proposed to implement Federated learning on Non-IID data with communication efficiency. In FedDA, the Conditional AutoEncoder (CVAE) is employed as the generator to generate the missing samples on Non-IID datasets. To achieve Federated learning, the Knowledge Distillation Mechanism is introduced. Instead of model weights or gradients, knowledge is shared between devices and servers. The knowledge is designed based on the hidden-layer features to reduce the communication overhead and protect data privacy. Meanwhile, to generate cross-class samples that are easy to classify, the mean value of latent variables for each class is constrained and the attention mechanism is introduced.

Acknowledgments

This work was supported in part by Sichuan Science and Technology Program under Grant 2020YFG0481, in part by CCF-Baidu Open Fund (NO.2021PP15002000), in part by CCF-Tencent Open Fund (NO. RAGR20210107). and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2021B1515140013.

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018. 1

[2] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[3] Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *Proceedings of the international conference on computer design (ICCD)*, 2019. 2

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of The Advances in Neural Information Processing Systems*, 2014. 1, 4

[5] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *Proceedings of The International Conference on Machine Learning*, 2020. 1

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proceedings of The Advances in Neural Information Processing Systems 2015 Deep Learning Workshop*, 2015. 1, 4

[7] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *Proceedings of The Advances in Neural Information Processing Systems 2018 the Second Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018. 1, 2, 6

[8] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. *Proceedings of The Advances in Neural Information Processing Systems*, 2021. 5

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of The International Conference on Learning Representations*, 2014. 4

[10] Jakub Konečný, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 1

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[12] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. 6

[13] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 68–79. IEEE, 2021. 2

[14] Guanglei Li, Bohao Feng, Huachun Zhou, Yuming Zhang, Keshav Sood, and Shui Yu. Adaptive service function chaining mappings in 5g using deep q-learning. *Computer Communications*, 2020. 1

[15] Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. Divergence-agnostic unsupervised domain adaptation by adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[16] Jingjing Li, Mengmeng Jing, Hongzu Su, Ke Lu, Lei Zhu, and Heng Tao Shen. Faster domain adaptation networks. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 1

[17] Jiachun Li, Yan Meng, Lichuan Ma, Suguo Du, Haojin Zhu, Qingqi Pei, and Sherman Shen. A federated learning based privacy-preserving smart healthcare system. *IEEE Transactions on Industrial Informatics*, 2021. 1

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of The International Conference on Artificial Intelligence and Statistics*, 2017. 1, 6

[19] Jonathan Ortigosa-Hernández, Inaki Inza, and Jose A Lozano. Measuring the class-imbalance extent of multi-class problems. *Pattern Recognition Letters*, 2017. 6

[20] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. Federated learning in vehicular networks: opportunities and solutions. *IEEE Network*, 2021. 1

[21] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *Proceedings of The International Conference on Machine Learning*, 2020. 2

[22] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 1, 2

[23] Stefano Savazzi, Monica Nicoli, Mehdi Bennis, Sanaz Kianoush, and Luca Barbieri. Opportunities of federated learning in connected, cooperative, and automated industrial systems. *IEEE Communications Magazine*, 2021. 1

[24] MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. In *Proceedings of the International Conference on Machine Learning*, 2020. 1, 2, 6

[25] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Proceedings of The Advances in Neural Information Processing Systems*, 2015. 1, 4

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6

[27] Saeed Vahidian, Mahdi Morafah, and Bill Lin. Personalized federated learning by structured and unstructured pruning under data heterogeneity. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 2021. 1, 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017. 5

[29] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *Proceedings of The IEEE International Conference on Computer Communications*. IEEE, 2020. 1

[30] Hui Wen, Yue Wu, Jingjing Li, Chenming Yang, Hancong Duan, and Yang Yang. Transferring inter-class correlation for teacher-student frameworks with flexible models. *Knowledge-Based Systems*, 2022. 1, 4

[31] Hui Wen, Yue Wu, Chenming Yang, Hancong Duan, and Shui Yu. A unified federated learning framework for wireless communications: towards privacy, efficiency, and security. In *Proceedings of The IEEE International Conference on Computer Communications 2020 The Second International Workshop on Intelligent Cloud Computing and Networking*, 2020. 1, 2

[32] Bangzhou Xin, Wei Yang, Yangyang Geng, Sheng Chen, Shaowei Wang, and Liusheng Huang. Private fl-gan: Differential privacy synthetic data generation based on federated learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020. 4

[33] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. Energy efficient federated learning over wireless communication networks. *arXiv preprint arXiv:1911.02417*, 2019. 1

[34] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. *Proceedings of The International Conference on Learning Representations*, 2021. 2

[35] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[36] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 1, 2

[37] Zhongyuan Zhao, Chenyuan Feng, Wei Hong, Jiamo Jiang, Chao Jia, Tony QS Quek, and Mugen Peng. Federated learning with non-iid data in wireless networks. *IEEE Transactions on Wireless Communications*, 2021. 2