GANFed: GAN-based Federated Learning with Non-IID Datasets in Edge IoTs

Xin Fan*, Yue Wang†, Weishan Zhang‡, Yingshu Li†, Zhipeng Cai†, and Zhi Tian‡
*School of Information Science and Technology, Beijing Forestry University, Beijing, China
†Department of Computer Science, Georgia State University, Atlanta, GA, USA
‡Department of Electrical & Computer Engineering, George Mason University, Fairfax, VA, USA
E-mails: fanxin@bjfu.edu.cn, {ywang182, yili, zcai}@gsu.edu, {wzhang23,ztian1}@gmu.edu

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10622211

***Abstract*—Federated learning (FL) is a promising distributed learning framework in terms of privacy protection and communication saving. Most existing FL techniques are developed for independent-and-identically-distributed (IID) datasets, but suffer from performance degradation under Non-IID datasets. To cope with this issue, most existing work designs solutions from data perspectives (e.g., sharing some data samples between local devices) to eliminate the heterogeneity of distributed datasets, which causes extra communication overhead and may expose user privacy that contradicts FL's original intention. Unlike the existing data-based methods, we propose a generative adversarial network (GAN) based FL, named as GANFed, which is designed from a feature perspective. Specifically, we embed a discriminator into the FL network, which works with the shallow layers as a generator to form a GAN in FL. By incorporating such a GAN, the output of the shallow layers tends to present more IID features compared with the original Non-IID input data. These extracted features from the shallow layers are then used to train the deep layers of the FL network. In this way, the proposed GANFed reduces the weight divergence of the local models, and hence improves the performance of FL. Without data exchange, our GANFed avoids the leakage of user privacy and reduces the communication overhead. Experimental results show that our GANFed outperforms the standard FedAvg on Non-IID dataset in terms of improved test accuracy.**

***Index Terms*—Federated learning, Non-IID data, generative adversarial network, FedAvg, edge IoTs.**

I. Introduction

With the rapid development of wireless communication technologies, a large number of IoT devices have emerged in edge networks, generating a large amount of data [1]–[3]. By utilizing artificial intelligence technologies (e.g., machine learning), extracting effective information from these data can stimulate many potential applications, such as unmanned driving, home automation, smart forestry and telemedicine [4], [5]. Traditional centralized machine learning requires distributed devices to transmit all raw data to a parameter server (PS), which not only increases communication overhead and divulges privacy, but also challenges the storage and computing capabilities of the PS [6], [7]. Alternatively, federated learning (FL) is proposed to train a global model by using a Federated Averaging (FedAvg) algorithm [8]–[12]. FedAvg provides a possibility for edge-intelligent devices to cooperatively learn a global model without raw data transmission. FedAvg only involves model updates between distributed devices and the PS, which not only saves communications, but also protects user privacy [13]–[15].

While FedAvg is an emerging approach in distributed learning, it encounters many challenges when applied in practice [16], [17]. One critical issue is that the performance of FL is inevitably deteriorated when FedAvg is applied on non-independent-and-identically-distributed (non-IID) datasets [18]–[20]. This is because that the heterogeneous nature of the distributed non-IID datasets drives the weights of the local models updated at the distributed devices to divergence [21]. According to [19], even though local models are trained from the same initial shared global model, they would converge to different optimal local models due to the heterogeneity in distributions of local datasets. When the PS aggregates the updated local model parameters from the distributed devices, the divergence between the aggregated model and the ideal one slows down the convergence speed and degrades the learning performance. Such issue can be partially compensated by exchanging data samples between local workers, which however may lead to excessive communication overhead and privacy leakage [22], [23].

In this work, we seek for a privacy-preserving and communication-efficient approach to deal with the non-IID issues in FL. Specifically, we propose a generative adversarial network (GAN) based FL, named as GANFed, which includes a new neural network model architecture and an new iterative training method for distributed learning. The key idea of GANFed is to incorporate a GAN that enforces the output of the shallow layers of the distributed model to present IID features on heterogeneous data. In GANFed, we introduce a classifier model (discriminator) into the target FL model, which is also trained in a distributed manner. Meanwhile, the shallow layers of the target FL model play the role of a generator and work together with the implanted discriminator to form a GAN. In GANFed, the target FL and GAN models are trained alternately until convergence. Without exchanging any data samples, the communication cost in our GANFed depends on the model size regardless the data samples, which is communication-efficient and privacy-preserving, compared with the methods relying on data sample exchanges [22], [23]. Our main contributions of this work are summarized as follows.

978-1-7281-9054-9/24/$31.00 ©2024 IEEE

- We propose a neural network model architecture for distributed learning, which embeds GAN into the target FL model. The components of GAN can assist in the training of the target FL model in a supervised mode.
- We propose a distributed training method that alternately trains GAN and target FL. Such an alternating training is carried out in the form of a tripartite game, which can respectively achieve the convergence of the three part network parameters.
- We investigated the supervisory role of GAN in our proposed GANFed through empirical studies. By adjusting the balance ratio of GAN in GANFed, our GANFed achieves a better learning performance than the classic FedAvg.

The rest of the paper is as follows. We first review the relevant literature in Section II. Then we develop our GANFed algorithm in Section III, which is evaluated by experiments on an image dataset in Section IV. Finally, we make conclusions and discuss some open problems in Section V.

II. Related Work

For standard FL, the heterogeneity in distributions of local datasets causes degradation in learning performance. To overcome such a non-IID issue, some data-based approaches are proposed by modifying the distributions of local datasets. There are two main solutions for data-based approaches, including data sharing and data argumentation.

A. Data Sharing

In [19], the authors propose to use a globally shared dataset with a uniform distribution to alleviate the negative effect of non-IID datasets. A part of globally shared dataset is downloaded to local devices so that local models are trained with both the globally shared data and their own local training data. This approach can improve the learning performance in the presence of non-IID datasets. To obtain such a uniformly distributed global dataset, some researchers propose to share some local data with the server to form a global dataset [22], [23]. Exchanging data samples can improve the global model performance on non-IID datasets, but it violates the requirement of privacy preserving in FL and leads to excessive communication overhead for data sharing.

B. Data Argumentation

Data argumentation is another option that can increase the diversity of training data so as to mitigate local data imbalance issues in FL. In [24], the authors displace some pixels of the original data samples to get the augmented data samples for local devices. Both the augmented data and the original local data samples are used to update the local model parameters. In [25], the authors propose to use GAN to generate the augmented data samples for local devices. The server collects some local seed data samples from local devices to train a good generator. This pre-trained generator is downloaded to local devices for reproducing the data samples, so as to make local datasets IID. Data argumentation also relies on data sharing, exposing data privacy.

Compared with the above data-based approaches, our GANFed is to make features IID rather than make data IID. Without data sharing, our GANFed is communication-efficient and privacy-preserving. Meanwhile, there are some other approaches to addressing the non-IID data issues in FL, such as local fine-tuning [26], [27], personalization layer [28]–[30] and so on [18]. These approaches improve the learning performance by adjusting the algorithm or model structure of FL, some of which can be combined with our GANFed. Here, we do not discuss them in depth.

III. GANFed Algorithm

A. System Model

Let $\mathcal{D}_i = \{\mathbf{x}_{i,k}, \mathbf{y}_{i,k}\}_{k=1}^{K_i}$ denote the local training dataset at the i-th worker, $i = 1, \ldots, U$, where $\mathbf{x}_{i,k}$ is the input data vector, $\mathbf{y}_{i,k}$ is the labeled output vector, $k = 1, 2, ..., K_i$, and $K_i = |\mathcal{D}_i|$ is the number of data samples available at the i-th worker. Note that the local datasets $\mathcal{D}_i,\ \forall i$ are non-IID in this work. With $K = \sum_{i=1}^{U} K_i$ samples in total, these U workers seek to collectively train a learning model parameterized by a global model parameterized by $\mathbf{w} = [w^1, \ldots, w^D] \in \mathcal{R}^D$ of dimension D, by minimizing the loss function $F(\mathbf{w}) := \mathbb{E}[f(\mathbf{w}; \mathbf{x}_{i,k}, \mathbf{y}_{i,k})]$, i.e.,

$$\textbf{P1:} \quad \mathbf{w}^* = \arg\min_{\mathbf{w}} \quad F(\mathbf{w}). \tag{1}$$

The minimization of $F(\mathbf{w})$ is typically carried out through the stochastic gradient descent (SGD) algorithm. The model parameter $\mathbf{w}_t$ at the t iteration is updated as

$$\text{(Model updating)} \quad \mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \frac{\sum_{i=1}^{U} \mathbf{g}_{i,t}}{U}, \tag{2}$$

where α is the learning rate and $\mathbf{g}_{i,t} = \nabla f(\mathbf{w}_{t-1}; \mathbf{x}_{i,k}, \mathbf{y}_{i,k})$ is the local gradient computed at the i-th local worker using its randomly selected the k-th data sample.

Considering the local gradients are updated based on the non-IID datasets, the model weights suffer from divergence, which then degrades the performance and convergence of the overall distributed learning. To solve such a problem efficiently and effectively, we next develop a GAN-based FL (GANFed) solution for efficient distributed learning without any data sample exchange.

B. Algorithm

As shown in Fig 1, we introduce a discriminator $\mathfrak{D}$ implanted with the shallow layers of our GANFed model to form a generator-discriminator pair as the GAN module of the proposed distributed learning framework. The shallow layers of our GANFed model is regarded as a feature extractor $\mathcal{G}$ that is also a generator in GAN. The discriminator is defined by the parameter $\boldsymbol{\theta} = [\theta^1, ..., \theta^{D_D}]$ with the dimension D_D. The shallow layers are parameterized by $\underline{\mathbf{w}} = [w^1, \ldots, w^{D_G}]$ with the dimension D_G. Then, the parameters of the target global model can be written as the concatenation of shallow and deep layers: $\mathbf{w} = [w^1, \ldots, w^{D_G}, \ldots, w^D]$.

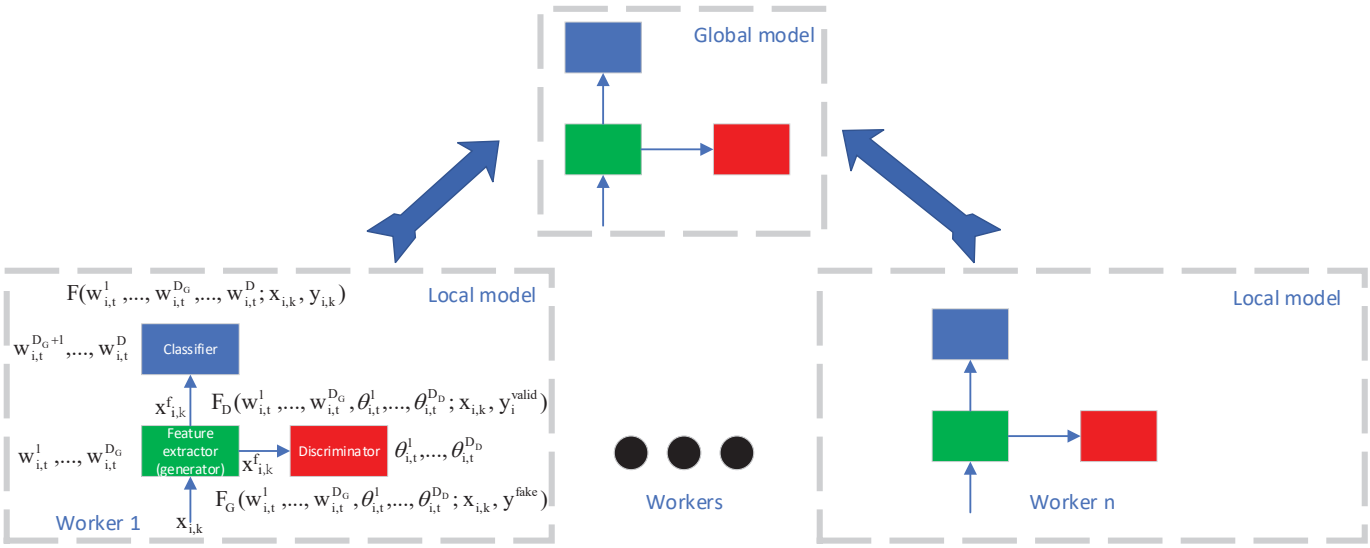

Fig. 1: GAN-based federated learning from Non-IID datasets.

The basic idea of our GANFed is to make the outputs of the shallow layers at all local workers tend to present IID features and then facilitate the training of the deep layers with such IID features. We aim to achieve that:

- The features (as transformed versions of the input data) extracted from different input data are expected to retain the useful information for training, i.e., the classifier can still classify different features as the same as their input data, the entire model still achieves the training objective as if it is given raw input data samples;
- The features extracted from different local workers are tend to IID, i.e., the discriminator cannot distinguish which local worker the features come from.

To this end, we introduce two types of labels for each input sample $\mathbf{x}_{i,k}, \forall i,k$, including the valid label $\mathbf{y}_i^{\texttt{valid}}$ and the fake label $\mathbf{y}^{\texttt{fake}}$, which are used to train the GAN. By the two types of labels, all the input data samples are categorized by the index of local worker i. Thus, the dimension of $\mathbf{y}_i^{\texttt{valid}}$ and $\mathbf{y}^{\texttt{fake}}$ are both U.

Specifically, the valid label $\mathbf{y}_i^{\texttt{valid}}$ carries the ground truth for data $\mathbf{x}_{i,k}$ and is used for updating the discriminator so that the discriminator can classify which worker the input data comes from. The elements of $\mathbf{y}_i^{\texttt{valid}}$ are all 0 except for the i-th element which is 1. The fake label $\mathbf{y}^{\texttt{fake}}$ is used for updating the feature extractor to fool the discriminator that cannot distinguish which worker the features after passing through the feature extractor comes from, the elements of which are all 1. The binary cross entropy (BCE) loss functions for training the discriminator and the feature extractor are given by $F_D(\underline{\mathbf{w}}, \boldsymbol{\theta}) := \mathbb{E}[f_D(\underline{\mathbf{w}}, \boldsymbol{\theta}; \mathbf{x}_{i,k}, \mathbf{y}_i^{\texttt{valid}})]$ and $F_G(\underline{\mathbf{w}}, \boldsymbol{\theta}) := \mathbb{E}[f_G(\underline{\mathbf{w}}, \boldsymbol{\theta}; \mathbf{x}_{i,k}, \mathbf{y}^{\texttt{fake}})]$, which are used for updating the parameters of the discriminator and the feature extractor, respectively.

Denote the extracted feature that the generator extracts from the input data $\mathbf{x}_{i,k}$ as $\mathcal{G}(\mathbf{x}_{i,k}) = \mathbf{x}_{i,k}^f$. From the labels and the loss functions, we can see that, if we use $(\mathbf{x}_{i,k}^f, \mathbf{y}_i^{\texttt{valid}})$ to train the discriminator with the BCE loss function F_D, the discriminator would adjust its parameters $\boldsymbol{\theta}$ to reduce the BCE loss so that it can classify the different extracted feature $\mathbf{x}_{i,k}^f$. On the other hand, if we use $(\mathbf{x}_{i,k}^f, \mathbf{y}^{\texttt{fake}})$ to train the generator with the BCE loss function F_G, the generator would adjust its parameters $\underline{\mathbf{w}}$ to reduce the BCE loss so that $\mathbf{x}_{i,k}^f$ belongs to all classes in the discriminator's eyes. This adversarial training will make the extracted features tend to IID. The loss function of GAN is given by

$$\min_{\mathcal{G}\triangleq\underline{\mathbf{w}}} \max_{\mathcal{D}\triangleq\boldsymbol{\theta}} V(\mathcal{G},\mathcal{D}) = \sum_i \sum_k y_i^{\texttt{valid}} \log(\mathcal{D}(\mathcal{G}(\mathbf{x}_{i,k}))) + \sum_i \sum_k y^{\texttt{fake}} \log(1-\mathcal{D}(\mathcal{G}(\mathbf{x}_{i,k}))). \tag{3}$$

To retain the useful information in the extracted feature $\mathbf{x}_{i,k}^f$, the generator should be trained together with the classifier by using $(\mathbf{x}_{i,k}, \mathbf{y}_{i,k})$ with the loss function F.

In order to obtain a global model (including the feature extractor, classifier, and discriminator) cooperatively, all workers and the PS train the learning model together in a distributed manner. The training procedure of the tripartite game has the following steps:

1) **Step 1:** The local workers download the global model ($\mathbf{w}$ and $\boldsymbol{\theta}$) from the PS.
2) **Step 2:** The local workers train the feature extractor and classifier using the loss function $F(\mathbf{w})$ and the input data $\mathbf{x}_{i,k}$, $\forall i,k$ with the label $\mathbf{y}_{i,k}$, and then update the parameters of the feature extractor and classifier $\mathbf{w}$.
3) **Step 3:** The local workers train the feature extractor and discriminator using the loss function $F_D(\underline{\mathbf{w}}, \boldsymbol{\theta})$ and the input data $\mathbf{x}_{i,k}$, $\forall i,k$ with the label $\mathbf{y}_i^{\texttt{valid}}$, and then only update the parameters of the discriminator $\boldsymbol{\theta}$, but keep the parameters of the feature extractor $\underline{\mathbf{w}}$

unchanged.

4) **Step 4:** The local workers train the feature extractor and discriminator using the loss function $F_G(\underline{\mathbf{w}}, \boldsymbol{\theta})$ and the input data $\mathbf{x}_{i,k},\ \forall i,k$ with the label $\mathbf{y}^{\texttt{fake}}$, and then only update the parameters of the feature extractor $\underline{\mathbf{w}}$, but keep the parameters of the discriminator $\boldsymbol{\theta}$ unchanged.
5) **Step 5:** The local workers send their own local updated parameters $\mathbf{w}$ and $\boldsymbol{\theta}$ to the PS.
6) **Step 6:** The PS averages all the received parameters from the local workers as the current global update $\mathbf{w}_t = \frac{1}{U}\sum_{i=1}^{U}\mathbf{w}_{i,t}$, and $\boldsymbol{\theta}_t = \frac{1}{U}\sum_{i=1}^{U}\boldsymbol{\theta}_{i,t}$, which are ready to download for distributed workers for next round iteration.

The above steps 1 - 6 are repeated until a predefined convergence condition is satisfied. The proposed GANFed is implemented as **Algorithm 1**.

Algorithm 1 GANFed

Initiation:
$\mathbf{w}_0$, $\boldsymbol{\theta}_0$.
for $t = 1 : T$ **do**
  At the workers:
  Download global model from the PS.
  Update the feature extractor and classifier:
  $\mathbf{w}_{i,t} = \mathbf{w}_{i,t-1} - \alpha\nabla f(\mathbf{w}_{t-1}; \mathbf{x}_{i,k}, \mathbf{y}_{i,k})$.
  Update discriminator:
  $\boldsymbol{\theta}_{i,t} = \boldsymbol{\theta}_{i,t-1} - \alpha\nabla f_D(\boldsymbol{\theta}_{t-1}; \mathbf{x}_{i,k}, \mathbf{y}_i^{\texttt{valid}})$.
  Update feature extractor:
  $\underline{\mathbf{w}}_{i,t} = \underline{\mathbf{w}}_{i,t-1} - \alpha\nabla f_G(\underline{\mathbf{w}}_{t-1}; \mathbf{x}_{i,k}, \mathbf{y}^{\texttt{fake}})$.
  Communication: Send $\boldsymbol{\theta}_{i,t}$ and $\mathbf{w}_{i,t}$ to the PS.
  At the PS:
  Average the parameters received from local workers as $\mathbf{w}_t = \frac{1}{U}\sum_{i=1}^{U}\mathbf{w}_{i,t}$, $\boldsymbol{\theta}_t = \frac{1}{U}\sum_{i=1}^{U}\boldsymbol{\theta}_{i,t}$.
  Break if a convergence condition is satisfied.
end for

IV. Experiments

This section presents the experimental results of the proposed GANFed[1], compared with the standard FedAvg in both IID and non-IID scenarios.

A. System Setting

We evaluate the performance of the proposed GANFed for the image classification tasks on the MNIST dataset[2]. In MNIST dataset, there are 60000 training samples and 10000 test samples.Unless otherwise specified, the system and parameter settings are as follows.

We set the total number of the local workers be 10. We first sort all the 60000 training samples based on their labels (the digits). Then we divide the 60000 training samples into 20 shards, each of which consists 3000 samples. We randomly distribute two shards to the 10 local workers for the distributed learning problem.

[1]The simulation code of this work can be found at: https://github.com/fuanxiyin/GANFed.git

[2]http://yann.lecun.com/exdb/mnist/

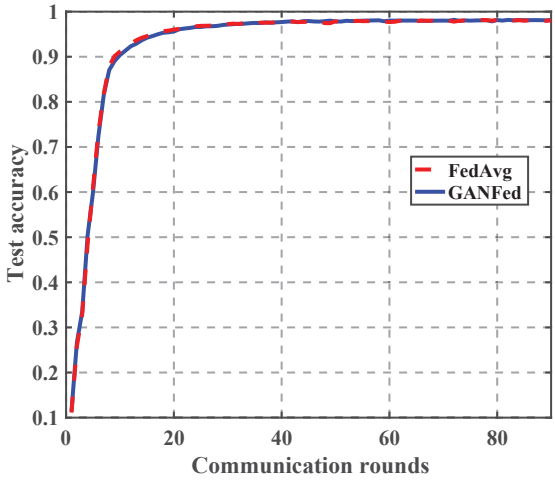

Fig. 2: Test accuracy as the number of communication rounds varies under the IID scenario.

B. Neuron Network Setting

The feature extractor consists of a 28*28-neuron input layer, a 128-neuron hidden layer, a 256-neuron hidden layer, a 512-neuron hidden layer, a 1024-neuron hidden layer, and a 28*28-neuron Tanh output layer. The leaky rectified linear unit (LeakyReLU) is the activation function. All the hidden layers are with batch normalization.

The discriminator consists of a 28*28-neuron input layer, a 512-neuron hidden layer, a 256-neuron hidden layer, and a 10-neuron Sigmoid output layer. The LeakyReLU is the activation function.

The classifier consists of a 28*28-neuron input layer, a 64-neuron hidden layer, and a 10-neuron softmax output layer. The rectified linear unit (ReLU) is the activation function. The hidden layers is with a dropout layer.

In all the training procedures, the loss functions are BCE, the SGD with the momentum=0.5 is adopted and the learning rates are all set to 0.01. We use FedAvg as a base line to compare with our proposed GANFed. Note that, considering the fairness, it is not suitable to compare our scheme to the existing approaches that handle non-IID data by adjusting the algorithm or model structure of FL. Actually, those existing approaches can be combined with our GANFed.

C. Results on IID

For IID scenarios, we randomly divide the 60000 training samples into 1200 shards, each of which consists 50 samples. We randomly distribute some shards to the 10 local workers (the number of shards distributed to each worker is at least one and maximum 30). The experiment results are shown in Fig. 2. As we can see, GANFed and FeDAvg have almost the same performance.

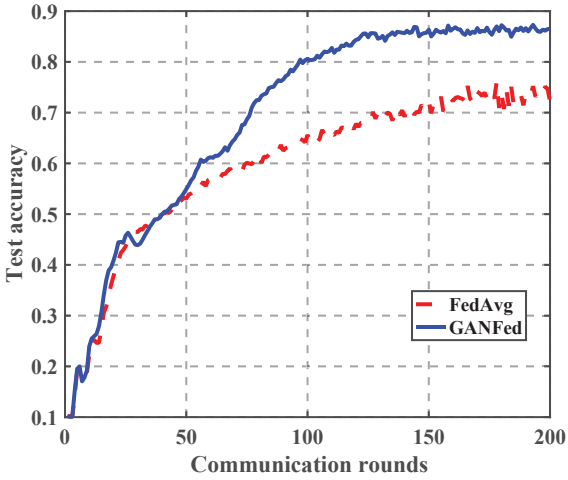

Fig. 3: Test accuracy as the number of communication rounds varies under the scenario where the ratio between the losses of FL and GAN is 1 : 1.

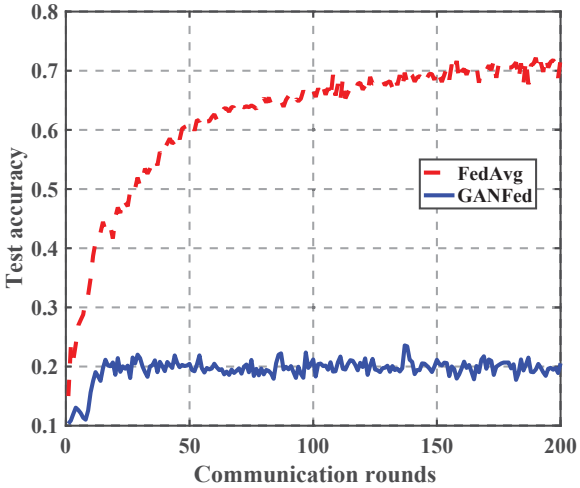

Fig. 5: Test accuracy as the number of communication rounds varies under the scenario where we first well train GAN.

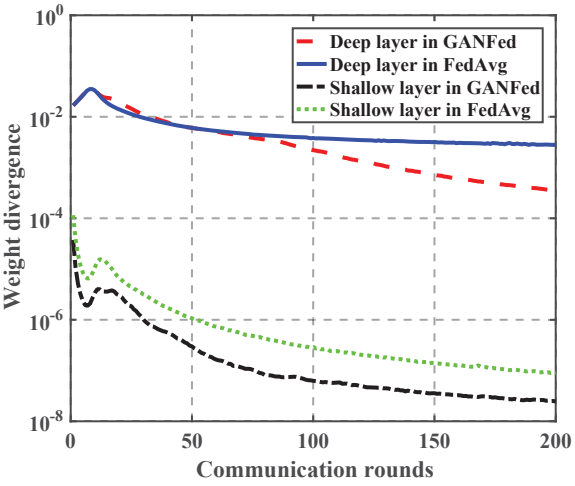

Fig. 4: Weight divergence as the number of communication rounds varies.

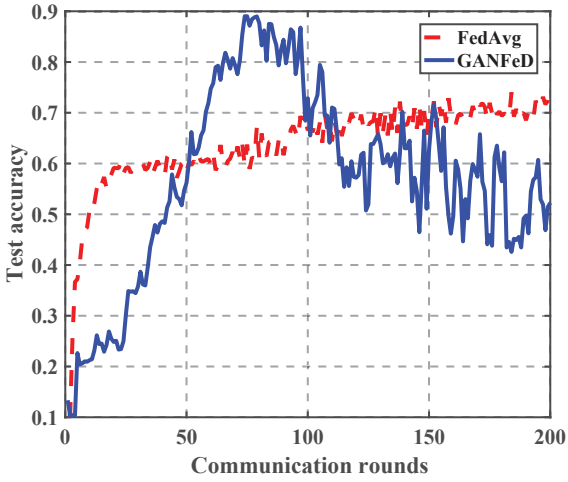

Fig. 6: Test accuracy as the number of communication rounds varies under the scenario where the ratio between the losses of FL and GAN is 1 : 10.

D. Results on Non-IID

The experiment results on non-IID datasets are shown in Fig. 3. As we can see in Fig. 3, the performance of the non-IID scenario is worse than that of IID scenario in both GANFed and FedAvg. However, our GANFed is superior than FedAvg in the non-IID scenario. The accuracy of GANFed is almost 10% higher than that of FedAvg.

In Fig. 4, we show how the weight divergence varies as the increase of the number of communication rounds. We calculate the weight divergence as

$$Weight\ \ Divergence = \frac{\sum_{i=1}^{U} \|\mathbf{w}_t - \mathbf{w}_{i,t}\|^2}{\|\mathbf{w}_t\|^2}. \quad (4)$$

As we can see, the weight divergences of both the shallow and the deep layers in GANFed are lower than that of FedAvg. This result explains the reason why our GANFed can improve performance.

In Fig. 5, we explore the supervisory role of GAN and provide the results. Under the same Non-IID setting as Fig. 3, we first well train GAN and then train the classifier keeping GAN unchanging. As we can see, GANFed can not work. Thus, we conclude that GAN can improve the performance of FL but it can also destroy the usability of data. We should adjust the ratio between the losses of FL and GAN when backward them in the training proceed, so that determine the appropriate level of supervision for GAN. It is better to train GAN and the classifier iteratively, rather than well pre-train GAN and then train the classifier.

In order to strengthen the supervisory role of GAN, we raise the loss of training GAN, i.e., we backward 10 times loss when training GAN. As we can see from Fig. 6, the performance of GANFed is weaker than FedAvg at the beginning, but GANFed is superior than FedAvg afterwards. This is because, the feature extractor is not trained very well at the beginning, results in a decline in data availability. Afterwards, the feature extractor is trained better so that it can make the features passed through it IID, which leads to an increase in performance. It is obvious that the performance of the non-IID scenario is worse than that of IID scenario. And our

GANFed is superior than FeDAvg in the non-IID scenario. The accuracy of GANFed is almost 30% higher than that of FedAvg, which is better than that of Fig. 3 where the ratio between the losses of FL and GAN is $1:1$. However, along with the training proceed, GANFed is worsening. This is because GAN's supervisory role is too strong, which destroys the usability of data. Thus, we can conclude that it is important to set a proper ratio to balance the backward loss of FL and GAN, which can be our future work.

V. Conclusion and Discussion

In this paper, we introduce GAN into FL and propose a novel GANFed algorithm for handling the challenging non-IID issue in distributed learning scenarios. Empirical studies show that our GANFed achieves higher accuracy than FedAvg on distributed non-IID datasets. More importantly, our GANFed can save communication and avoid privacy leakage, due to without data sharing. In addition, our study indicates that GAN makes features tend to IID, while it also reduces the usability of data as well. That is, GAN is a supervisor for adjusting the shallow layers of the FL to improve the learning accuracy. However, if the supervision is too strong, it will reduce the accuracy of the final global model. In future work, we will study the balance ratio between GAN and FL in the training proceed of GANFed.

Acknowledgments

This work was partly supported by the Science and Technology Innovation Project of Xiongan Grant 2022XACX0400, the US NSF Grants #1939553, #2003211, #2128596, #2231209, and #2413622, and the VRIF-CCI Grant #223996.

References

[1] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "CB-DSL: Communication-efficient and byzantine-robust distributed swarm learning on non-i.i.d. data," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 1, pp. 322–334, 2024.

[2] Z. Qadir, K. N. Le, N. Saeed, and H. S. Munawar, "Towards 6g internet of things: Recent advances, use cases, and open challenges," *ICT Express*, vol. 9, no. 3, pp. 296–312, 2023.

[3] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[4] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[5] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[6] Y. Wang, Z. Tian, X. Fan, Z. Cai, C. Nowzari, and K. Zeng, "Distributed Swarm Learning for Edge Internet of Things," *IEEE Communications Magazine*, Early Access, 2024.

[7] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 48–54, 2020.

[8] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "1-bit compressive sensing for efficient federated learning over the air," *IEEE Transactions on Wireless Communications*, vol. 22, no. 3, pp. 2139–2155, 2022.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[10] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "Joint optimization of communications and federated learning over the air," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4434–4449, 2022.

[11] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[12] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "BEV-SGD: Best effort voting sgd against byzantine attacks for analog-aggregation-based federated learning over the air," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 946–18 959, 2022.

[13] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, 2021.

[14] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "Communication-efficient federated learning through 1-bit compressive sensing and analog aggregation," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[15] ——, "Joint optimization for federated learning over the air," in *2022 IEEE International Conference on Communications (ICC 2022)*. IEEE, 2022, pp. 1–6.

[16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[17] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "Best effort voting power control for byzantine-resilient federated learning over the air," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022, pp. 1–6.

[18] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *arXiv preprint arXiv:2106.06843*, 2021.

[19] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[20] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Generation Computer Systems*, vol. 135, pp. 244–258, 2022.

[21] G. Zhou, P. Xu, Y. Wang, and Z. Tian, "H-nobs: Achieving Certified Fairness and Robustness in Distributed Learning on Heterogeneous Datasets," in *Thirty-seventh Annual Conf. on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, December, 2023.

[22] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL: Cooperative learning mechanism using non-iid data in wireless networks," *arXiv preprint arXiv:1905.07210*, 2019.

[23] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, "Overcoming noisy and irrelevant data in federated learning," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5020–5027.

[24] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," in *2019 IEEE 37th international conference on computer design (ICCD)*. IEEE, 2019, pp. 246–254.

[25] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[26] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 174–10 183.

[27] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[28] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[29] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[30] X.-C. Li, L. Gan, D.-C. Zhan, Y. Shao, B. Li, and S. Song, "Aggregate or not? exploring where to privatize in dnn based federated learning under different non-iid scenes," *arXiv preprint arXiv:2107.11954*, 2021.