# Decentralized Federated Learning: A Defense Against Gradient Inversion Attack

Guangxi Lu[1], Zuobin Xiong[1], Ruinian Li[2], and Wei Li[1(✉)]

[1] Department of Computer Science, Georgia State University, Atlanta, GA, USA
{glu3,zxiong2}@student.gsu.edu, wli28@gsu.edu
[2] Department of Computer Science, Bowling Green State University, Bowling Green, OH, USA
lir@bgsu.edu

**Abstract.** Federated learning (FL) is a machine learning technique that enables data to be stored and calculated on geographically distributed local clients. In centralized FL, there is an orchestrating system server responsible for aggregating local client parameters. Such a design is vulnerable to gradient inversion attacks where a malicious central server can restore the client's data through the model gradients. This paper proposes a Decentralized Federated Learning (DFL) method to mitigate the gradient inversion attack. We design a federated learning framework in a decentralized structure, where only peer-to-peer communication is adopted to transfer model parameters for aggregating and updating local models. Extensive experiments and detailed case studies are conducted on a real dataset, through which we demonstrate that the proposed DFL mechanism has excellent performance and is resistant to gradient inversion attack.

**Keywords:** Federated learning · Peer to peer network · Privacy protection

## 1 Introduction

Training machine learning model requires massive date, but the required training data is often stored at different silos or devices, making it difficulty for aggregation. Federated Learning (FL), as a prevalent distributed machine learning framework, can effectively solve the data island problems by hiring a central server to help multiple silos/devices train the model collaboratively. By training a global model without sharing local data, FL achieves privacy protection compared to traditional machine learning methods. Following this idea, FL has been applied in various fields, such as Electronic Finance [1], Recommendation System [2], and Biomedicine [3].

Recent research has shown that federated learning does not perfectly protect local data [4–6]. Although all data are stored locally during federated learning,


Z. J. Haas et al. (Eds.): WiCON 2022, LNICST 464, pp. 44–56, 2023.
https://doi.org/10.1007/978-3-031-27041-3_4

the transmission of model parameters during training can still leak important private information. Specifically, an attack known as gradient inversion attacks threatens the private data in federated learning. Since the central server holds all the model parameters in the training process, it can obtain exact training gradients from the parameters change and rely on these gradients to restore the private data maliciously.

Removing the central server is an intuitive way to resist gradient inversion attacks. Following this idea, we proposed privacy-preserving decentralized federated learning method(DFL) to resist this gradient inversion attack by removing the central server and hide the gradients from the attacker. In DFL, no centralized parameter server is required to maintain a global model over the local clients. Instead, all clients in the system are directly connected to their neighboring clients, forming a peer-to-peer (P2P) network/graph structure where they only communicate with their one-hop neighbors. In this DFL method, the clients trains their model locally with their own data, transmit the model to one-hop neighbors through the peer-to-peer network, and then update the new model by aggregating received model parameters. As a result, the mighty global model does not exist during model training. Therefore, a malicious attacker cannot obtain accurate model gradients, and the private data of clients is protected.

To validate the privacy-preserving capability and training efficiency of DFL, we conduct extensive experiments on the FASHION-MNIST dataset for image classification tasks. The experiments show that DFL guarantees that parameter transmitted in P2P communication manner does not compromise the privacy of local data. The contributions of this paper are summarized as follows:

- A Decentralized Federated Learning method (DFL) is proposed. Instead of using a central server to aggregate client model parameters, the DFL uses a peer-to-peer network to train models by aggregating parameters from neighboring clients.
- We investigate the problem of privacy leakage in FL and demonstrate that DFL can prevent local data feature leakage by comparing and analyzing the gradient inversion attacks on centralized and decentralized federated learning frameworks.
- Extensive experiments are performed to evaluate the performance of the proposed DFL. In addition, the privacy-preserving capability of the DFL is verified by performing the gradient inversion attacks on the trained DFL model.

The rest of this paper is organized as follows. The current studies in related fields are introduced in Sect. 2. The preliminary is introduced in Sect. 3, following which we propose our DFL method in Sect. 4. The experimental results and analysis are provided in Sect. 5. Finally, we summarize this article in Sect. 6.

## 2 Related Work

In this section, we present related work of gradient inversion attack and introduce the development of Decentralized Federated Learning at the current stage.

### 2.1 Gradient Inversion Attack

The training process of federated learning is considered as a privacy-preserving process because the local dataset never leaves local clients. However, recent studies have shown that attackers can effectively restore local data through gradient inversion attack, which significantly affects the privacy-preserving capability of federated learning. Le *et al.* [7] first proposed that model gradients may leak private information about the training data. Hitaj *et al.* [8] proposed a Generative Adversarial Network (GAN) based gradient inversion attack model. This GAN-based framework uses the global model parameters as discriminators, generates false data through generators, and makes the generated data with the same distribution as the private training data. Zhu *et al.* [4] proposed a deep leakage of gradient (DLG) framework. The DLG method first generates dummy gradients by inputting dummy data and dummy labels into a model. Then, the distance between the dummy gradients and the true gradients is optimized so that the dummy data and labels are approximately equal to the private training data and labels. Zhao *et al.* [9] proposed an improved DLG model, which adds a label prediction module to the original DLG. This improved DLG model only needs to optimize the dummy data, which significantly improves the inference efficiency and accuracy of the model. In the work of [5], authors proposed a gradient inversion attack model for multi-batch federated learning. This attack model is able to restore the original image from the model gradient even if the batch size is up to 100. Yin *et al.* [6] proposed the GradInversion method, which can effectively recover the original hidden image from random noise by inverting the given batch-averaged gradients.

The above attacks significantly threaten the security and privacy of federated learning. Although some methods can protect gradients of a trained model, such as differential privacy or homomorphic encryption, they cannot essentially solve the problem of privacy leakage.

### 2.2 Decentralized Federated Learning

Research on decentralized federated learning is still at the early stage. Currently, there are two general schemes for decentralized federated learning. The first type of schemes select a node from a peer-to-peer network as a temporary central parameter server through some election mechanism, thus ensuring the fairness and security of the network. Behera *et al.* [10] proposed a decentralized federated learning method based on the Raft consensus algorithm. This method does not have a constant central server but achieves aggregation of models by continuously selecting temporary nodes as central servers. Wang *et al.* [11] proposed a novel federated framework called "swarm learning", in which the model selects a central server in the network through the blockchain to enable the secure aggregation of model parameters.

The other type of schemes obtain global consensus through the peer-to-peer network and train the model. Lalitha *et al.* [12] proposed a fully decentralized federated learning method. This method introduces the concept of data distribution beliefs and uses a Bayesian-like approach to allow all clients to train a

global model jointly. Based on this work, Lalitha *et al.* [13] proposed an improvement method. In the improved method, each client obtains a local belief, and a global belief among all nodes is obtained through communication. Then, all clients jointly train a global model with this global belief. Hu *et al.* [14] proposed a decentralized federated learning training method based on gossip communication protocol. This model achieves convergence of the federated learning by passing the model parameters in the segment.

For the above methods, a malicious client can still restore the private data of the target victim by inputting the neighboring model parameters and the global model parameters. There still does not exist a decentralized federated learning method that can defend against gradient inversion attack effectively.

## 3 Preliminaries

Federated learning is proposed to protect data privacy in participating local clients. The most commonly used federated learning algorithm is the FedAvg [15]. Assume that there are $K$ clients and one parameter server involved in the system. The core idea of FedAvg is to train the model on each client $i$, $i \in \{1, 2, \cdots, K\}$ and upload these local models to the server. The server, then, performs global model aggregation based on these uploaded models by FedAvg. Specifically, for each training round $t$, the central server first distributes the global model $\omega_{global}^{t}$ to each client for local training. The initial model of each local client $i$ in round $t$ is $\omega_i^{t_0} = \omega_{global}^{t}$. Then, local training is performed on each client for $E$ epochs, producing an updated local model $\omega_i^{t_E}$. In each local training epoch $e \in \{0, 1, \ldots, E-1\}$, the client picks a local data point $(x_i, y_i)$ to obtain a local model gradient:

$$\nabla \omega_i^{t_e} = \frac{\partial \ell \left( F\left(x_i, \omega_i^{t_e}\right), y_i \right)}{\partial \omega_i^{t_e}}, \tag{1}$$

where $F(\cdot)$ is the predicted value of the model for the input data $x_i$, $\omega_i^{t_e}$ is the local model in the $t$-th training round after $e$ local epochs.

Then, the local client model is updated by stochastic gradient descent (SGD):

$$\omega_i^{t_{e+1}} = \omega_i^{t_e} - \eta \nabla \omega_i^{t_e}, \tag{2}$$

where $\eta$ is the learning rate. After $E$ epochs of local training, the final local model on client $i$ is $\omega_i^{t_E}$, which can be represented as $\omega_i^{t+1}$ (*i.e.*, $\omega_i^{t+1} = \omega_i^{t_E}$). Then, the local client uploads this local model $\omega_i^{t+1}$ to the central server for aggregation. In the server aggregation, the server averages all the local models to obtain the global model $\omega_{global}^{t+1}$ as Eq. (3),

$$\omega_{global}^{t+1} = \sum_{i=1}^{K} \frac{n_i}{n} \omega_i^{t+1}, \tag{3}$$

where $n_i$ is the size of dataset in client $i$, and $n = \sum_{i=1}^{K} n_i$. Once $\omega_{global}^{t+1}$ is obtained, the server sends this global model to each client to start the next round of local training.

During the FedAvg process, only the local model and global model parameters are transmitted between the server and clients, achieving the joint training of multiple clients without sharing local data. However, these model parameters can still leak private information as pointed out by recent research in Sect. 2. Among those gradient inversion attacks, Deep Leakage from Gradients (DLG) model [4] expresses the severest threat to federated learning. This attack can be used to infer the possible labels and perfectly restore the data of a target client by the gradients. Specifically, the DLG model first randomly initializes a dummy data point $(x_i', y_i')$ for the target client $i$. Then, a dummy gradient $\nabla\omega_i'^t$ at the $t$-th training round is calculated by feeding the data into the global model $\omega_{global}^t$ as Eq. (4):

$$\nabla\omega_i'^t = \frac{\partial\ell\left(F\left(x_i', \omega_{global}^t\right), y_i'\right)}{\partial\omega_{global}^t}. \tag{4}$$

By minimizing the distance between the dummy gradient $\nabla\omega_i'^t$ and the true local gradient $\nabla\omega_i^t$, the DLG method can optimize the dummy data and labels $(x_i', y_i')$ to true data and labels $(x_i, y_i)$. The objective function of the DLG model is as follows:

$$\begin{aligned} x_i'^*, y_i'^* &= \arg\min_{x_i', y_i'} \left\|\nabla\omega_i'^t - \nabla\omega_i^t\right\|^2 \\ &= \arg\min_{x_i', y_i'} \left\| \frac{\partial\ell\left(F\left(x_i', \omega_{global}^t\right), y_i'\right)}{\partial\omega_{global}^t} - \frac{\omega_i^{t_0} - \omega_i^{t_E}}{\eta} \right\|^2 \end{aligned} \tag{5}$$

where $x_i'^*, y_i'^*$ are the recovered data and label. Since the server controls the global model $\omega_{global}^t = \omega_i^{t_0}$ and all local models $\omega_i^{t+1} = \omega_i^{t_E}$ in FedAvg method, it is possible that the malicious server can perform data restoration for all target clients via Eq. (5). It should be noticed that the objective function of DLG is the distance between gradients, and it is necessary to calculate gradient to optimize this function. Therefore, the prediction function $F(\cdot)$ should be second-order derivable. Fortunately, most machine learning models are second-order derivable.

## 4 Methodology

### 4.1 Peer to Peer Network

Our proposed DFL adopts peer-to-peer (P2P) communication to build a fully decentralized federated learning framework. We denote the peer-to-peer communication network with an undirected graph $G = (V, A)$, where $V$ is the set of nodes (*i.e.,* clients), and $A$ is the adjacency matrix of this graph. For any node $i \in V$, we define the neighboring node set as $N(i)$. So, the value $A_{ij} \in \{0, 1\}$ is equal to 1 if and only if $j \in N(i)$. We assume that this network is a strongly connected aperiodic graph, which makes the matrix $A$ aperiodic and irreducible.

In the setting of our DFL, the client $i$ periodically communicates to its neighboring nodes $N(i)$ for model training. Thus node $i$ can only get information

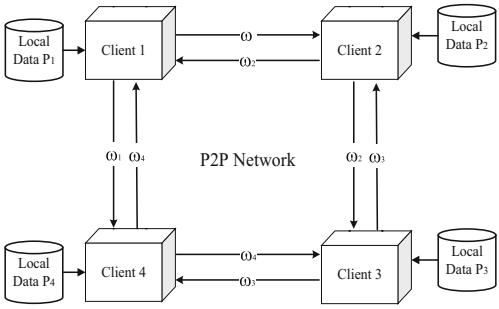



**Fig. 1.** Decentralized federated learning framework structure

about the neighboring nodes $N(i)$ in each communication round. Therefore, if the client $i$ wants to get the information of a node $q$-th hop away, it needs to make $q$ rounds of communication. The theory of six degrees of separation [16] provides the relation between the average path length and the number of neighboring clients, which is expressed in Eq. (6):

$$Q = \frac{\log(K)}{\log(R)}, \tag{6}$$

where $K$ is the number of clients in this network, $R$ is the average number of neighbors per node, and $Q$ is the average path length of any two nodes in this network. The balance between the number of neighboring nodes $R$ and the average path length $Q$ is crucial when building this peer-to-peer network. Too many neighboring nodes will lead to the overhead of single node communications, while too large average path length means that each node needs more communication rounds to obtain information from further clients.

### 4.2 Decentralized Federated Learning Algorithm

The Decentralized Federated Learning (DFL) framework is shown in Fig. 1. Each client $i$ stores the local datasets $P_i$, and connects to the assigned clients through the P2P network.

The DFL framework is divided into three states: Initialization, Local Training, and Parameter Aggregation. The Initialization state is to initialize client models and peer-to-peer networks at the beginning of training. Then, each client performs Local Training and Parameter Aggregation to update their models. We also refer to the execution of these two states (*i.e.,* Local Training and Parameter Aggregation) as one training round of the DFL.

**Initialization.** We need to build the P2P network structure and initialize the local model parameters in the initialization state. We use a server to assist with this state. Note that the server is only used to initialize the DFL framework instead of being a part of the training process. Thus, this framework remains a

decentralized federated learning framework. Specifically, the server generates a graph structure with specific settings and assigns each client with its neighboring clients. Then, each client establishes communications with the neighboring clients according to the server information. At the same time, the initialized parameters of the model are sent to each client, thus ensuring the consistency of the client initialization parameters.

**Local Training.** In the local training state, a client $i$ takes the local model $\omega_i^t$ in training round $t$ as the input model and local dataset $P_i$ as the input data to train an updated model $\overline{\omega}_i^{t_0}$. The local training is set to train $E$ local epochs with local minibatch $B$. In particular, the local model $\omega_i^t$ is optimized by stochastic gradient descent (SGD) via Eq. (1) and Eq. (2). When $E$ local training epochs are completed, the updated model parameters $\overline{\omega}_i^{t_0}$ is equal to the final local model, *i.e.*, $\overline{\omega}_i^{t_0} = \omega_i^{t_E}$.

**Parameter Aggregation.** After the local training state, all clients start the Parameter Aggregation state. There are $D(D \geq 1)$ communication rounds in this parameter aggregation state. For each communication round, client $i$ sends updated model parameters $\overline{\omega}_i^{t_d}, d \in \{0, 1, \ldots, D-1\}$ to all its neighboring clients $N(i)$. For convenience, the client only receives data without any feedback. The model aggregates all the received updated models to form newly updated parameters. The parameter aggregation can be described as:

$$\overline{\omega}_i^{t_{d+1}} \leftarrow \frac{1}{M} \sum_{j=1}^{M} \overline{\omega}_j^{t_d}, \tag{7}$$

where $j \in N(i)$ is the neighboring client of client $i$, $d$ is the current communication round, and $M = |N(i)|$ is the number of neighbors for client $i$. When $D$ communication rounds are completed, the newly obtained local model in $t+1$ round $\omega_i^{t+1}$ is equal to final updated model, where $\omega_i^{t+1} = \overline{\omega}_i^{t_D}$.

The complete DFL algorithm is shown in Algorithm 1.

### 4.3 Training Scheme

Considering the uncertainty of the convergence for decentralized federated learning, we design various training schemes for the proposed DFL with different $D$ values. In the individual scheme (DFL-I), the communication rounds $D$ is set to 1, and the client can only communicate with neighboring nodes within one-hop. As a result, it is more difficult to obtain a global consensus. Therefore, the trained model is prone to oscillation, converging relatively slowly. However, the advantage of DFL-I is that the communication cost for each node is relatively small. In the global scheme (DFL-G), the communication rounds $D$ is set to $\lceil Q \rceil$, in which the model can reach the global consensus in each training round. Consequently, DFL-G tends to train the same model, which requires fewer training rounds for a single client to reach convergence. However, the communication costs for each node are relatively high.

# 5 Experiments

## 5.1 Datasets

In this paper, Fashion-MNIST [17] is used as an experimental dataset. Fashion-MNIST contains 60,000 images for training and 10,000 images for testing. In our experimental setting, the Fashion-MNIST dataset is divided in three ways to represent the different sets of clients. (i) Average division. The dataset is randomly divided into $K$ shards, where each shard has the same amount of data. (ii) Not identically and independently distributed with balanced amount(Non-I.I.D balanced). The dataset is divided into $K$ shards, where each shard has the same amount of data but different labels. (iii) Not identically and independently distributed with imbalanced amount(Non-I.I.D imbalanced). The dataset is divided into $K$ shards, where each shard has a different amount of data and labels.

**Algorithm 1.** Decentralized Federated Learning Algorithm.

Inputs: Dataset $P_i$ from each client $i \in V$.
Outputs: $K$ well trained models $\omega_i^T$ for each client $i \in V$.
**Initialization**:
The server forms a P2P network and initialize the model parameters of each client $\omega_i^0$.
**for** DFL training round $t$ from 0 to $T$ **do**
  **Local Training**:
  $\beta \leftarrow$(split $P_i$ into batches of size B)
  **for** each local epoch $e$ from 0 to $E$ **do**
    **for** batch $b \in \beta$ **do**
      $\omega_i^{t_{e+1}} = \omega_i^{t_e} - \eta \nabla \omega_i^{t_e}$
    **end for**
  **end for**
  $\overline{\omega}_i^{t_0} = \omega_i^{t_E}$
  **Parameter Aggregation**:
  **for** Communication round $d$ from 1 to $D$ **do**
    Send $\overline{\omega}_i^{t_d}$ to all neighborhood node $j \in N(i)$
    Receive $M = |N(i)|$ data parameters
    Updata local model:
    $\overline{\omega}_i^{t_{d+1}} \leftarrow \frac{1}{M} \sum_{j=1}^{M} \overline{\omega}_j^{t_d}$
  **end for**
  $\omega_i^{t+1} = \overline{\omega}_i^{t_D}$

**end for**

## 5.2 Experiment Setting

In order to evaluate the performance of the decentralized federated learning method, we compare the accuracy and convergence efficiency of model training between four baselines and our proposed DFL.

- Centralized machine learning method (CML): The model structure is a four-layer CNN (1*28*28–32*28*28–32*14*14–64*14*14–64*7*7).
- Centralized federated learning method (CFL): We use FedAvg [15] algorithm with the same CNN model structure.
- Decentralized federated learning baseline: We use a Gossip-based (SGossip) [14] and Global Belief-based decentralized federated learning method (GB) [13] with the same CNN model structure.

**Table 1.** The accuracy of image classification on the fashin-MNIST datasets.

| | Average division | Non-I.I.D balanced | Non-I.I.D Imbalanced |
|---|---|---|---|
| CML | 92.98% | | |
| CFL | 90.54% | 88.56% | 88.32% |
| SGossip | 89.21% | 88.18% | 87.62% |
| GB | 87.44% | 86.36% | 86.19% |
| DFL-I | 84.67% | 82.54% | 80.66% |
| DFL-G | 85.55% | 84.48% | 82.26% |

**Table 2.** The accuracy of label restoration attack by DLG.

| Batchsize | 1 | 5 | 10 | 16 | 32 |
|---|---|---|---|---|---|
| CFL | 100% | 97.42% | 95.74% | 94.32% | 88.53% |
| SGossip | 100% | 95.33% | 92.31% | 90.76% | 85.44% |
| GB | 100% | 87.48% | 84.55% | 81.82% | 74.64% |
| DFL-I | 100% | 54.42% | 48.78% | 41.71% | 27.54% |
| DFL-G | 100% | 64.88% | 60.14% | 53.87% | 45.67% |

The initial learning rate of all baselines is set at 0.01. For all federated learning methods, the training rounds are set to 500. We analyze the performance by comparing the classification accuracy and restoration result after DLG attack.

### 5.3 Performance Evaluation

Table 1 shows the accuracy of each method for the image classification task. According to the results, the centralized machine learning (CML) method has the best image classification accuracy. For decentralized federated learning, the accuracy of gossip-based method (SGossip) is relatively high. This is because the SGossip method consumes high communication costs to enable all clients to jointly train the same global model. Essentially, the training effect of gossip-based decentralized federation learning is similar to CFL. For our DFL structure, the accuracy of DFL-G is higher than DFL-I. Compared to DFL-I where a training round only trains with one-hop neighboring, each DFL-G training round

**Table 3.** The results of image restoration attack by DLG.

| | FFT | PSNR |
|---|---|---|
| CFL | 0.165 | 12.82 |
| SGossip | 0.276 | 12.02 |
| GB | 0.506 | 9.96 |
| DFL-I | 1.301 | 3.22 |
| DFL-G | 0.952 | 4.98 |

trains with more clients and more datas. Therefore, DFL-G can train a more accurate model for image classification. In addition, when the data distribution is Non-I.I.D and imbalanced, the accuracy of centralized federated learning decreases slowly. However, our DFL leads to more degradation in accuracy under the Non-I.I.D and imbalanced distribution. When the data distribution is Non-I.I.D, especially when the local distribution is not consistent with the global distribution, the gap between the local model and the theoretical global optimal model will be more significant, leading to a large classification error. Table 2 shows the results of the label inference attack under the Deep Leakage from Gradients (DLG) attack. The accuracy of label prediction decreases as the batch size increases because the data with batch size $N$ has $N!$ different permutations [4], which makes label inference attacks harder. The label restoration accuracy of DLG for the GB method is lower than that of the SGossip and CFL method, while the label restoration accuracy of the DFL method is much lower than that of the GB method. This is because in our DFL, the attacker cannot obtain the exact training gradient by global model, thus causing errors when optimizing the dummy labels in Eq. (5). Furthermore, the label inference attacks perform worse when DFL-I is used than DFL-G. In the DFL-I scheme, each client is trained jointly with the one-hop neighbor, so each training model has differences based on localization. In the DFL-G scheme, each client is trained jointly with more clients through more communication, reducing the localization of model training and making each training model the same. Therefore, the attacker's model parameters can approximate the victim's model parameters to obtain a more accurate gradient, which increases the accuracy of the label restoration.

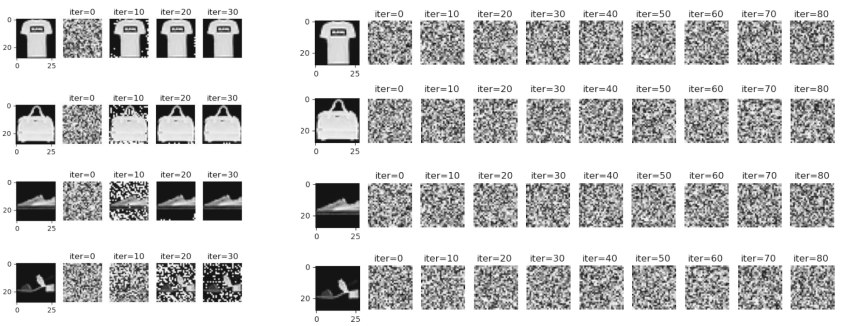
We also compare the performance of DLG for image restoration under different federated learning methods. We use two metrics to determine the restoration results of the images. The first is the cosine distance between the original and restored images after the 2-dimensional Fast Fourier Transform (FFT). A higher FFT value represents a worse restoration result and better privacy-preserving ability of the method. The other metric is the Peak Signal-to-Noise Ratio (PSNR) between the original image and the restored image. A higher value of PSNR means a better restoration result and a worse privacy-preserving ability of the method. Based on the results shown in Table 3, the DFL model achieves higher FFT and lower PSNR values, which implies better privacy protection compared to the baseline model. This result is similar to the results we obtained from label

inference attack. This is because other federated learning methods tend to train a global model, whereas the DFL method trains different local models suitable for their local dataset. As a result, the attacker does not have access to the local model of the target client, and therefore, cannot obtain an accurate gradient inversion attack for image restoration.

Table 4 shows the required communication rounds for each method. The centralized federated learning method (CFL) requires a small number of communication rounds. The decentralized federated Learning methods with individual training scheme (GB and DFL-I) require more communication rounds, because decentralized methods are more prone to parameter oscillations during model training process compared with centralized methods. The decentralized federated learning methods with global training schemes (SGossip and DFL-G) require the most communication rounds, because the global schemes consume more communication rounds to reach global consensus. There are more clients involved in training process, thus more communication rounds are needed to train a more accurate model rather than those individual training schemes.

**Table 4.** Communication round of each model.

| | Average division | Non-I.I.D balanced | Non-I.I.D imbalanced |
|---|---|---|---|
| CFL | 62 | 82 | 93 |
| SGossip | 972 | 1104 | 1152 |
| GB | 193 | 231 | 241 |
| DFL-I | 437 | 477 | 492 |
| DFL-G | 1173 | 1266 | 1320 |




(a) attack retults on CFL (b) attack results on DFL

**Fig. 2.** Case study of image restoration attack on fashion-MNIST dataset.

Figure 2 shows the image restoration result of DLG on CFL and DFL. Here, the attacker restores four images of the target t-shirt, bag, sneaker, and sandal. In each subfigure, the first column represents the original images, followed by

the images restored by the DLG model as the number of DLG training rounds increases. From Fig. 2, it can be found that the CFL method is vulnerable to DLG attack, where the training data is recovered clearly within 30 rounds. By comparison, the DFL method can resist the image restoration attack by DLG much better. Figure 2 shows that the original image is still not restored after 80 rounds with the DFL model. The case study demonstrates that our DFL method achieves better privacy against DLG attacks.

## 6 Conclusion

This paper proposes a decentralized federated learning method (DFL), in which local clients can transfer model parameters through a peer-to-peer network to their neighboring client and thus train a model jointly without a central parameter server. Through extensive experiments, we compare our proposed DFL with other DFL designs and demonstrate that our DFL method effectively defend against gradient inversion attacks and maintain excellent model performance.

## References

1. Yang, W., Zhang, Y., Ye, K., Li, L., Xu, C.-Z.: FFD: a federated learning based method for credit card fraud detection. In: Chen, K., Seshadri, S., Zhang, L.-J. (eds.) BIGDATA 2019. LNCS, vol. 11514, pp. 18–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23551-2_2
2. Tan, B., Liu, B., Zheng, V., Yang, Q.: A federated recommender system for online services. In: Fourteenth ACM Conference on Recommender Systems, pp. 579–581 (2020)
3. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. J. Healthcare Inform. Res. **5**(1), 1–19 (2021)
4. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
5. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural Inform. Process. Syst. **33**, 16937–16947 (2020)
6. Yin, H. .,Mallya, A., Vahdat, A., Alvarez, J.M., Kautz, J.: See through gradients: Image batch recovery via gradinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16337–16346 (2021)
7. Phong, L.T., Aono, Y., Hayashi, T., Wang, L., Moriai, S.: Privacy-preserving deep learning: revisited and enhanced. In: Batten, L., Kim, D.S., Zhang, X., Li, G. (eds.) ATIS 2017. CCIS, vol. 719, pp. 100–110. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-5421-1_9
8. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 603–618 (2017)
9. Zhao, B., Mopuri, K. R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)
10. Behera, M.R., Shetty, S., Otter, R., et al.: Federated learning using peer-to-peer network for decentralized orchestration of model weights (2021)

11. Warnat-Herresthal, S., Schultze, H., Shastry, K.L., Manamohan, S., Mukherjee, S., Garg, V., et al.: Swarm learning for decentralized and confidential clinical machine learning. Nature **594**(7862), 265–270 (2021)
12. Lalitha, A., Shekhar, S., Javidi, T., Koushanfar, F.: Fully decentralized federated learning. In: Third workshop on Bayesian Deep Learning (NeurIPS) 2018
13. Lalitha, A., Kilinc, O.C., Javidi, T., Koushanfar, F.: Peer-to-peer federated learning on graphs. arXiv preprint arXiv:1901.11173 (2019)
14. Hu, C., Jiang, J., Wang, Z.: Decentralized federated learning: A segmented gossip approach. arXiv preprint arXiv:1908.07782 (2019)
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR, pp. 1273–1282 (2017)
16. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks". Nature **393**(6684), 440–442 (1998)
17. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)