2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) | 979-8-3503-8199-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/TrustCom60117.2023.00046

FedRSM: Representational-Similarity-Based Secured Model Uploading for Federated Learning

Gengxiang Chen[a], Sheng Liu[a], Xu Yang[a], Tao Wang[a], Linlin You[a]*, Feng Xia[b]
[a] School of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen, China
[b] School of Computing Technologies, Royal Melbourne Institute of Technology, Melbourne, Australia
{chengx77, liush235, yangx359}@mail2.sysu.edu.cn, {wangt339, youllin}@mail.sysu.edu.cn, {feng.xia}@rmit.edu.au

***Abstract*—As a novel learning paradigm, Federated learning (FL) aims at protecting privacy by avoiding raw data shifting between distributed clients and central servers. However, recent researches demonstrate the vulnerability of FL against gradient-based privacy attacks, in which, gradients intercepted by malicious adversaries may result in data leakage. Current defense methods suffer from performance drops, low privacy guarantees, and high communication costs. Motivated by this, we propose FedRSM, a Representational-Similarity-Based Secured Model Uploading for Federated Learning. FedRSM splits Deep Neural Networks (DNNs) into layers, calculates Representational Dissimilarity Vector (RDV), measures the similarity between local RDV and global RDV of each model layer, and constructs secured local model to be uploaded based on Representational Consistency Alteration (RCA). According to the evaluation result, FedRSM can improve testing accuracy by up to 2%, significantly reduce communication costs, and avoid data leakage under different model complexities.**

***Index Terms*—Federated Learning, Privacy Protection, Secured Model Uploading**

I. Introduction

In recent years, Deep Learning [1] has achieved significant success in various domains, e.g., computer vision [2] and natural language processing [3]. To train Deep Neural Networks (DNNs) with high performance, conventional methods rely on centralized servers to control the whole learning process from data collection to model training. However, in the ubiquitous Internet of Things (IoT) systems and services, massive data generated by distributed smart devices trigger serious privacy protection problems and require high network throughput. Aiming at providing a privacy-preserving and cost-efficient training procedure, Federated Learning (FL) [4] is developed.

Instead of sharing the raw data of clients with the server, FL allows clients to train models locally and transmit only gradients or model parameters to the global server. Nevertheless, recent research on gradient-based privacy attacks has demonstrated the vulnerability of FL [5]. By intercepting gradients uploaded by clients, malicious adversaries are capable of recovering raw data. With the rapid development of attack algorithms [6]–[9], the privacy-preserving capability of FL awaits to be strengthened. To limit the risk of data leakage during the learning process, current research, which mainly focuses on securing information transmitted to the server, can be categorized into three groups, namely: 1) Homomorphic Encryption (HE) that encrypts information sent to the sever [10], 2) Differential Privacy (DP) that adds Gaussian or Laplace noises to the raw data [11], and 3) Data Masking that masks part of the data to the global server.

Due to the additional operations or information added to be processed, existing methods suffer from serious performance deterioration in either learning speed or cost. To be specific, to support the gradually growing IoT systems and services, FL faces several key challenges to ensure the privacy of clients without leaking raw data to malicious adversaries, and in the meanwhile, to train a high-performance global model in an acceptable period of time. Therefore, a holistic method that is of high security to inherit the benefits of FL awaits to be discovered.

Motivated by this, we propose FedRSM, a Representational-Similarity-Based Secured Model Uploading for Federated Learning. In general, firstly, to guarantee user privacy and reduce communication costs, FedRSM splits DNNs into layers and chooses to hide some of them (without uploading them to the server) based on their individual contributions. Secondly, to measure the importance of each layer, based on Representational Similarity Analysis (RSA) [12]–[14], FedRSM calculates Representational Dissimilarity Vector (RDV) using stimulus data pairs. By further computing Representational Consistency (RC) and Representational Consistency Alteration (RCA), the value of each layer can be compared and thus construct a secured model locally. Finally, while the secured local models are uploaded, the global server adopts a layer-wise aggregation mechanism to update the global model. Accordingly, the main contributions of this paper can be summarized as follows:

- FedRSM is proposed by introducing Representational Consistency Alteration (RCA), which is less computation complexity than the calculation of representational dissimilarity matrix, to avoid data leakage attacks by masking layers according to their values of RCA.
- FedRSM implements a weighted and layer-wise model aggregation function to process secured local models uploaded from clients, through which, the heterogeneity among secured local models can be remedied and the global model can be updated more accurately.
- FedRSM is compared with four state-of-the-art baselines (i.e., FedAVG, DP with different noise levels, FedSplit, and FedCG) to train three kinds of DNNs (i.e., LeNet,

*Corresponding author: Linlin You, e-mail: youllin@mail.sysu.edu.cn

2324-9013/23/$31.00 ©2023 IEEE
DOI 10.1109/TrustCom60117.2023.00046

ConvNet, and ResNet18) based on two standard datasets (i.e., German Traffic Sign and CIFAR-10). The evaluation results show that FedRSM can achieve a joint improvement in both training performance and security protection.

The rest of this paper is organized as follows. First, Section II summarizes research related to data leakage attacks, privacy-preserving methods, and the utilization of RSA in FL. Second, Section III and Section IV introduce and evaluate the proposed FedRSM, respectively. Finally, Section V concludes this paper and discusses future works.

II. Related work

This section summarizes key related works. First, the state-of-the-art data leakage attack algorithms are introduced as the motivation of this paper. Second, current defense methods are summarized. Third, the utilization of RSA in FL is reviewed.

A. Data Leakage Attack

Many researchers try to recover raw data from gradients. A classic algorithm, named Deep Leakage from Gradients (DLG), [5] first proves the probability of leakage. Furthermore, an improved DLG (iDLG) is proposed to enhance the label inference accuracy [6]. To recover large-size images (i.e., up to 224×224), Inverting Gradients (IG) algorithm [7] decomposes single parameter gradient into corresponding norm magnitude and direction. In [8], Generative Regression Neural Network (GRNN) is introduced by formulating data recovery into a regression problem. Finally, aiming at the recovery of batch data, catastrophic data leakage in vertical federated learning (CAFE) [9] is proposed to recover data with batch size over 100 in vertical FL.

B. Security-enhancing Method

While attack algorithms demonstrate the vulnerability of FL, many efforts have been done to strengthen the security required in FL. Without losing generability, these methods can be categorized into three groups:

- **Homomorphic Encryption (HE)**: Cryptography is used to secure the data and support the training of DNNs [15]. Moreover, BatchCrypt is proposed to speed up encryption-related operations [16], and multi-key HE is introduced into FL to lift the security level [17]. However, HE still suffers from additional computation and communication costs.
- **Differential Privacy**: DP is combined with FL by adding noises into parameters uploaded by clients [18], e.g., Bayesian DP is employed to secure the parameters transmitted through the network [19]. Moreover, by considering the data heterogeneity of IoT systems and services, [20] provides a solution to train personalized models. Nevertheless, DP can not avoid data leakage and may also impact the model performance.
- **Data Masking**: It avoids data leakage by masking part of the transmitted data, e.g., sharing only half of the local model parameters to the server [21]. Moreover, the conditional Generative Adversarial Network (GAN) is applied in FL by decomposing DNNs into a public classifier and a private extractor, and only sharing the classifier and generator [22]. Although masking part of the data can help increase security, how to avoid performance reduction due to the absence of data while maintaining cost-efficient learning remains an open problem.

C. Representational Similarity Analysis in FL

As a useful tool to compare differences among neurons, RSA [12] is introduced into deep learning [13] to analyze layer-wise similarity (i.e., RC) of DNNs. Moreover, RC is also used to enhance model aggregation of FL for a performance boost [14]. However, RSA has not been used to address the privacy concerns of FL as it can be an important indicator to mask unessential information and in turn, save communication costs and improve communication security occurred between the clients and the server.

III. Methodology

As shown in Fig. 1, FedRSM proposed by this work is divided into three phases that execute iteratively, namely:

- **P1: Local Training Phase**: Clients train local models using corresponding private data.
- **P2: RSA Phase**: Clients conduct RSA, and then form secured local models by masking layers.
- **P3: Global Aggregation Phase**: The server updates the global model through layer-wise aggregation.

The rest of this section will introduce the three phases of FedRSM, repsectively.

A. local Training Phase in FedRSM

In this work, we consider an FL system with one central server receiving local models updated by clients, controlling the global model aggregation and distribution process, and K clients which hold private datasets $\{D_1, D_2, ..., D_K\}$. Specifically, each private dataset is divided into a training set D_k^{train} and a testing set D_k^{test}. Moreover, each client k preserves a classification network w_k consisting of L layers, i.e., $w_k = \{w_k^1, w_k^2, \ldots, w_k^L\}$ where $w^i, i \in [1, L]$ denotes the parameters of i^{th} layer.

Assume that once a client k joins the system, first, it retrieves the latest global model w^t corresponding to t^{th} global round from the server, then, sets w^t as its current local model w_k^t, and finally, conducts local training based on its local training set D_k^{train} according to Formula 1,

$$w_k^t = w_k^t - \alpha \nabla_{w_k^t} F_k(w_k^t, D_k^{train}) \tag{1}$$

where α refers to learning rate, and $F_k(w^t, D_k^{train})$ is the loss calculated based on D_k^{train} of client k.

Since the full model uploading may lead to data leakage, FedRSM proposed by this paper aims at strengthening security by uploading local models with layers masked while maintaining high model performance. Therefore, how to avoid the

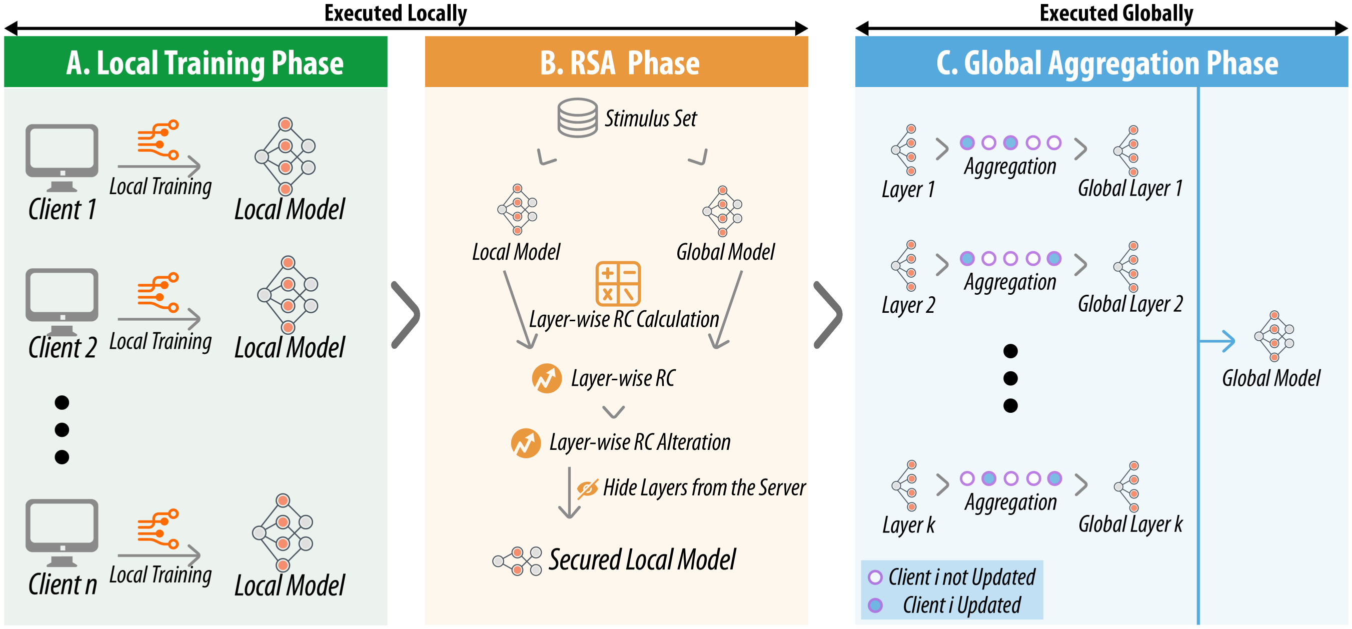

Fig. 1. The overview of FedRSM consisting of A) Local Training Phase, B) RSA Phase, and 3) Global Aggregation Phase.

performance dropping caused by masked and heterogeneous models uploaded by the clients is essential. To tackle this issue, FedRSM introduces a cost-efficient mechanism, called Representational Similarity Analysis (RSA), to calculate the importance of each model layer, and accordingly form secured local models (with related layers masked) to be uploaded.

B. Representational Similarity Analysis in FedRSM

The RSA phase of FedRSM is implemented to construct a secured local model as illustrated in Fig. 2. After local training is completed, each client executes RSA to measure the importance of each model layer. Specifically, given a DNN layer, stimulus data pairs selected from the server-prepared stimulus datasets are processed by the updated local model and the received global model, respectively. After that, the distance between each data pair is calculated according to Formula 2,

$$d^l[i,j] = \sqrt{\sum_{dim=1}^{N}(o_i^l(dim) - o_j^l(dim))^2} \tag{2}$$

where $d^l[i,j]$, corresponding to the l^{th} layer, is the distance between stimulus data i and j, and $o_i^l(\cdot)$ is the output vector including N dimensions given stimulus data i. It is worth noting that different from calculating a Representational Dissimilarity Matrix (RDM) in traditional RSA, FedRSM selects E elements from RDM to form a Representational Dissimilarity Vector (RDV) to reduce the computational complexity.

With the calculation of local and global RDVs, the Representational Consistency (RC) can be defined as Formula 3,

$$RC^l = [\rho(RDV_{glo}^l, RDV_{loc}^l)]^2 \tag{3}$$

where RC^l refers to the RC value of l^{th} layer, which is measured based on the Pearson Correlation Coefficient ρ of two RDVs.

To quantify the importance of each DNN layer, a layer-wise RC Alteration (RCA) is designed according to Formula 4,

$$RCA_{t+1}^l = |\frac{RC_{t+1}^l - RC_t^l}{RC_t^l}| \tag{4}$$

where RCA_{t+1}^l is the RCA value of l^{th} layer in $(t+1)^{th}$ global training round.

Based on RCA, clients can choose to upload layers with higher RCA according to their communication budget (i.e., bandwidth), which together form the secured local model.

C. Gobal Aggregation Phase in FedRSM

Once local training is completed, client k uploads the secured local model $\bar{w}_k^t$ to the server and awaits for the updated global model. As for the server, it distributes the global model to every client at the beginning of each round, receives local models continuously, and starts the global aggregation when all the local models are received.

After the secured models are received, the server can update the global model based on according to Formula 5 and 6. In general, a layer-wise model aggregation process is implemented to remedy the impact of secured local models, as they may have some layers absent.

$$w^{t+1} = \sum_{k=1}^{K}(\frac{|D_k|}{\sum_{k=1}^{K}|D_k|}\bar{w_k}) \tag{5}$$

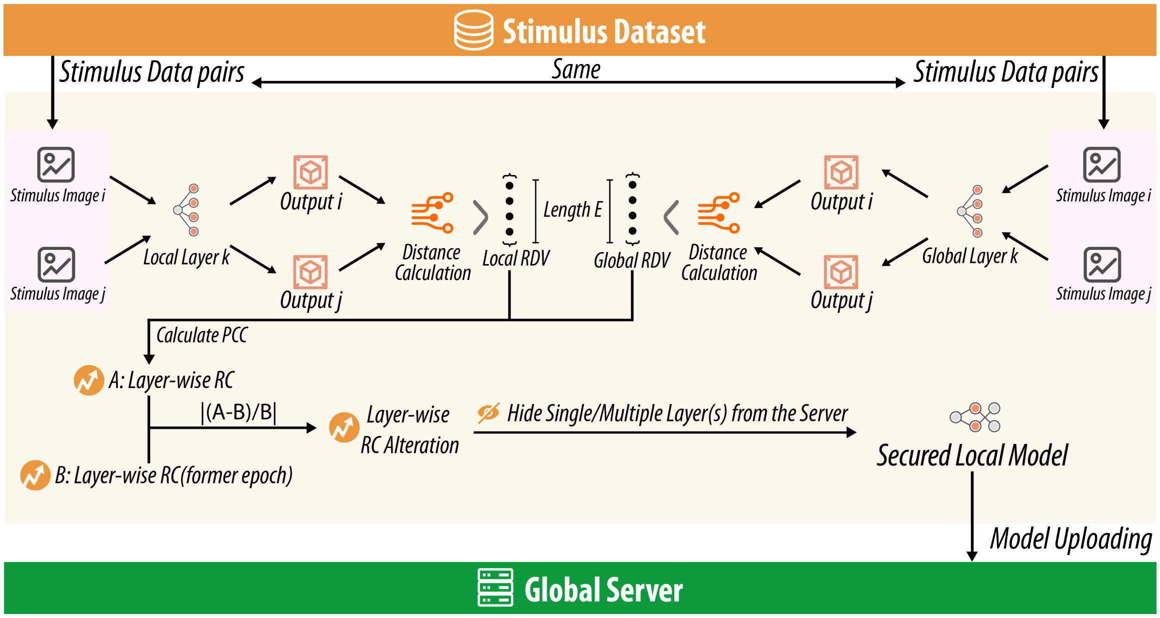

Fig. 2. The workflow of RSA in FedRSM

$$\bar{w_k} = \sum_{l=1}^{L} l_{sign} * w_k^l \tag{6}$$

where $|D_k|$ is the dataset size of client k, $\bar{w_k}$ refers to the secured local model uploaded by the corresponding client, and w_k^l, which refers to the l^{th} layer of local model, is guided by $l_{sign} \in \{0, 1\}$ in model uploading (i.e., 1 for uploading).

D. Algorithm of FedRSM

As illustrated in Algorithm 1, FedRSM runs at both sides of the client and server.

- **Client Side** (Lines 1-11): For each client in FedRSM, it first receives global model w^t from the server and sets it as w_k^t (a.k.a., local model), then conducts local training process and updates w_k^t. Next, it stimulates each model layer l with stimulus data pairs and calculates RCA^l. Finally, it forms and uploads a secured local model $\bar{w_k^t}$ to the server.
- **Server Side** (Lines 12-16): The server of FedRSM keeps receiving local models, and starts the global aggregation procedure once all clients have their local models uploaded. After the update of the global model w^{t+1}, it is distributed to every client to start another round of learning. Accordingly, the global training round t increases to $t + 1$, and the whole system steps into the next round. Note that the learning ends when a stop condition is met (i.e., maximum learning iterations, the target accuracy, etc.).

In summary, FedRSM ensures the security of the local data of clients by using RSA, which can construct a secured local model with masked layers, and also remain high model performance by implementing a layer-wise aggregation. The merit of FedRSM is evaluated in the following section.

Algorithm 1 FedRSM Algorithm

```
Client Side:
 1: for $k \le K$ in parallel do
 2:     Receiving global model $w^t$ from server as $w_k^t$
 3:     Training local model using $D_k^{train}$
 4:     for layer $l \in L$ do
 5:         Stimulating both $w^t$ and $w_k^t$
 6:         Calculating $RC^l$ according to Formula 3
 7:         Calculating $RCA^l$ according to Formula 4
 8:     end for
 9:     Forming secured local model $\bar{w_k^t}$ by Formula 6
10:     Uploading secrued local model $\bar{w_k^t}$
11: end for
Server Side:
12: Receiving local models continuously
13: when $k = K$ do
14:     Constructing global model $w^{t+1}$ by Formula 5
15:     Distributing global model $w^{t+1}$ to clients
16:     Increasing $t$ to $t + 1$
```

IV. Evaluation

In this section, the model performance of FedRSM as well as its security-protecting ability will be evaluated.

A. Experiment Settings

To reveal the capability of the proposed mechanism, i.e., the ability to improve learning performance and data security, FedRSM is compared with four state-of-the-art baselines in training three kinds of DNNs on two standard datasets.

TABLE I
SUMMARY OF MODELS TO BE TRAINED

Term	LeNet	ConvNet	ResNet18
Learning rate	0.001	0.001	0.001
Batch size	20	20	20
Number of convolution layers	3	9	17
Number of fully connected layers	1	1	1
Model size (KB)*	166	3625	43828

*The size of complete model.

1) Models: Three network models are used in the experiment, namely:

- **LeNet [5]**: A variant of classic network LeNet-5.
- **ConvNet [7]**: A network containing 9 convolutional layers to support the classification task.
- **ResNet18 [23]**: A variant of a residual training framework to ease the training burden of DNNs.

Note that the configurations, structure, and hyperparameters of each model are listed in Table I.

2) Datasets: Two datasets are chosen for performance evaluation, namely:

- **German Traffic Sign (GTS) [24]**: A benchmark for traffic sign detection containing different types of traffic signs from Germany. The dataset involves 34,799 training pictures and 12,630 testing pictures divided into 43 categories. Each picture is a 32×32×3 color image.
- **CIFAR-10 [25]**: A dataset with reliable labels comprising 10 classes of non-overlapping images (e.g., automobile, truck, and bird). It consists of 50,000 training images and 10,000 testing images with the size of 32×32×3.

3) Baselines: We compare the performance of FedRSM with 4 baselines, mostly with security-protecting abilities:

- **FedAVG [4]**: It is a classic FL method where clients upload full models and the server aggregates them for the global model based on the average function. It is the one vulnerable to the attacks.
- **Differential Privacy [11]**: It is a widely used method in cryptography. We implement DP by adjusting different levels ($\sigma^2 = 0.001$ and $\sigma^2 = 0.1$) of Gaussian noises and adding the noises to the uploaded local models.
- **FedSplit [21]**: It is an FL method that shares only partial local models with the server to avoid attacks causing data leakage. We split each network model into half and only transmit the latter parts to the server.
- **FedCG [22]**: It is an FL algorithm that creates the model with a private extractor and a public classifier, uses GAN to mimic the output of the extractor, and shares the parameters of the generator and classifier with the server.

4) Evaluation Metrics: Two kinds of evaluation metrics are utilized, namely:

- **Performance Evaluation**: An FL system with 10 clients and 1 server is created to train 6 DNNs (i.e., three models times two datasets) per method. Specifically, each client is assigned with the same amount of training and testing images (i.e., 1,000 for GTS and 500 for CIFAR-10) with batch size 20 and a learning rate of 0.001. Note that for FedRSM, we choose not to upload the layer with the highest RCA value, and two images from each category are prepared as stimuli with RDV dimension E of 50. The performance of FedRSM is evaluated by three metrics, i.e., 1) Top-1 test accuracy according to Formula 7, where TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives, respectively; 2) the total communication cost measured by Formula 8, where n_k^t refers to the size of local model uploaded by client k in the t^{th} communication round and T represents the number of global training round; and 3) the accumulated training time according to Formula 9, where $Time_{train}^t$ refers to the training time of the t^{th} communication round.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

$$Cost = \sum_{t=1}^{T}\sum_{k=1}^{K}(\frac{n_k^t}{1024 \times 1024}) \tag{8}$$

$$Time_{train} = \sum_{t=1}^{T} Time_{train}^t \tag{9}$$

- **Privacy Evaluation**: The worst case with the most data leakage probability is considered in the experiment. To be specific, assume that the server is malicious and tries to recover the original images of clients from variation $w^t - w_k^t$ by using IG algorithm [7]. Moreover, clients only process single data with batch size 1. Attacks against each image repeat three times, each of which lasts 24,000 rounds, and the recovered image with the highest Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) will be presented as the final result.

B. Performance Evaluation

The performance of the compared methods is compared and discussed according to the three performance metrics, namely: Top-1 test accuracy, total communication cost, and total training time. The results are shown in Table II.

1) Top-1 test accuracy: First, as illustrated in Fig. 3, compared to other models, ConvNet achieves the highest accuracy on both datasets. Second, FedRSM outperforms baselines on 3 out of 6 trained DNNs, i.e., achieving an improvement of 2% on GTS under LeNet, CIFAR-10 under LeNet, and CIFAR-10 under ConvNet, respectively. Moreover, FedRSM maintains the same performance with FedAVG and DP(σ^2 = 0.001) on GTS under ConvNet and ResNet18. It shows the competitive performance of the proposed method in handling the heterogeneous information preserved by different clients.

TABLE II
THE COMPARISON* OF FEDRSM AND BASELINES ILLUSTRATING 1) TOP-1 TEST ACCURACY, 2) COMMUNICATION COSTS, AND 3) TRAINING TIME.

German Traffic Sign	**LeNet**			**ConvNet**			**ResNet18**		
	Acc	*Cost(GB)*	*Training Time(s)*	*Acc*	*Cost(GB)*	*Training Time(s)*	*Acc*	*Cost(GB)*	*Training Time(s)*
FedRSM	**0.83**±0.01	0.44	2147.98	**0.98**±0.01	9.27	5163.02	**0.93**±0.01	**94.44**	11793.93
FedAVG	0.81±0.02	0.47	2097.75	**0.98**±0.01	10.37	4884.34	**0.93**±0.01	125.39	**10368.47**
DP(0.1)	0.66±0.04		2102.56	0.90±0.02		4903.42	0.84±0.02		10456.70
DP(0.001)	0.81±0.01		2112.49	**0.98**±0.01		4894.11	**0.93**±0.01		10433.33
FedSplit	0.75±0.03	**0.42**	**2093.10**	0.95±0.02	**8.44**	**4876.81**	0.64±0.11	117.85	10399.74
FedCG	0.76±0.03	9.25	37023.33	0.67±0.04	20.57	54441.55	0.61±0.05	126.23	53627.87
CIFAR-10	**LeNet**			**ConvNet**			**ResNet18**		
	Acc	*Cost(GB)*	*Training Time(s)*	*Acc*	*Cost(GB)*	*Training Time(s)*	*Acc*	*Cost(GB)*	*Training Time(s)*
FedRSM	**0.49**±0.03	0.44	1032.77	**0.72**±0.04	9.29	2910.15	0.52±0.03	**93.52**	5537.04
FedAVG	0.46±0.03	0.47	**996.37**	0.70±0.03	10.37	2873.77	**0.53**±0.05	125.39	5236.11
DP(0.1)	0.28±0.03		1002.72	0.46±0.02		2886.01	0.49±0.03		5238.64
DP(0.001)	0.47±0.05		999.54	0.70±0.03		2880.27	0.51±0.04		5232.10
FedSplit	0.41±0.06	**0.42**	1002.66	0.51±0.03	**8.44**	**2872.01**	0.35±0.03	117.85	**5229.05**
FedCG	0.46±0.03	9.25	38822.95	0.40±0.08	20.57	42228.23	0.35±0.04	126.23	45184.65

*Results in **bold** refer to the best performance while underlined results refer to the second best performance.

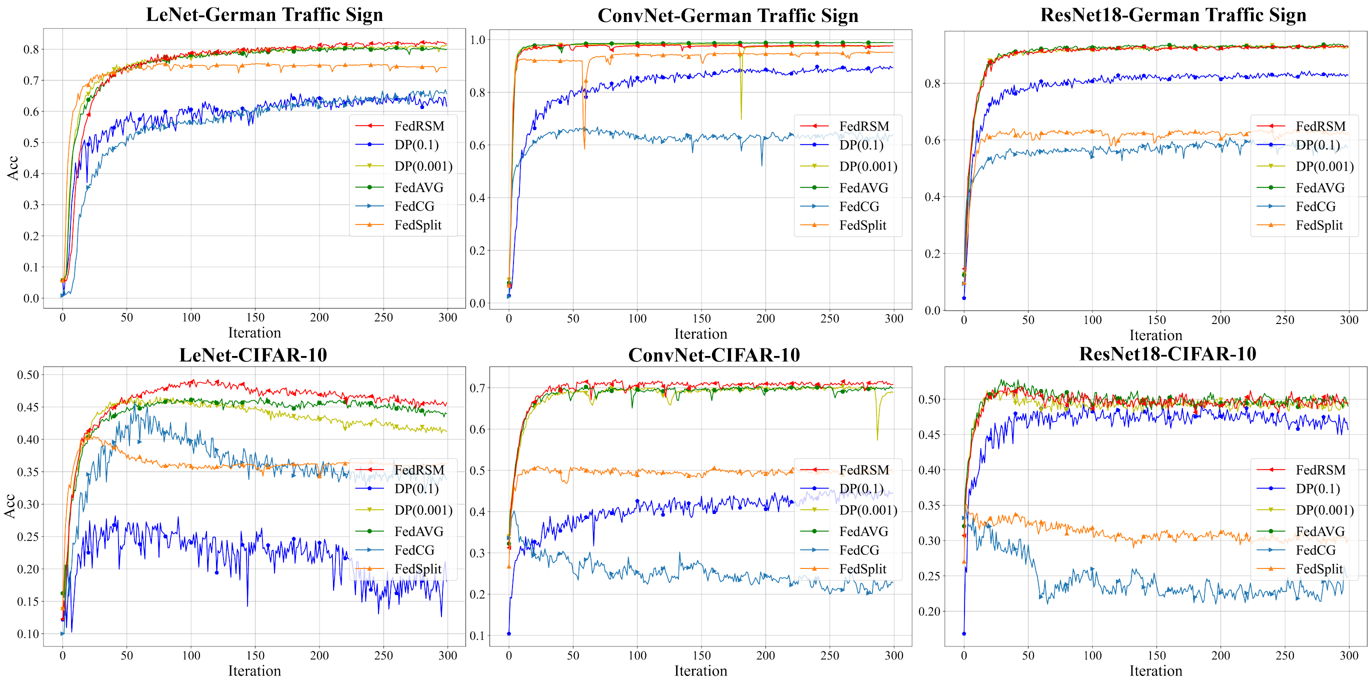

Fig. 3. Top-1 test accuracy on German Traffic Sign and CIFAR-10 under different model architectures.

2) *Communication Cost:* In general, FedAVG and DP require the same communication cost as they upload full models during the training. Since FedCG requires uploading the parameters of generators and classifiers, it consumes more communication resources. On the contrary, FedSplit and FedRSM can lessen the communication cost, since both of them only need to upload partial models. To be specific, FedSplit achieves the lowest cost to train LeNet (0.42 GB) and ConvNet (8.44 GB), whereas FedRSM maintains the second best. Moreover, FedRSM overcomes FedSplit in training ResNet18 with a cost of 94.44 GB. It implies that with the growth of model complexity, due to the randomness of RSA, FedRSM can be more cost-efficient than FedSplit, as FedRSM can upload different types of model layers adaptively according to their RSA values, and instead, FedSplit needs to consistently upload fully connected layers, which, in general, contain a large number of parameters.

3) *Training Time:* : First, in the case of using GTS to train the three models, more training time is required than CIFAR-10, as GTS contains more images. Second, FedAVG, DP, and FedSplit share almost the same training time. Third, since FedRSM adds additional operations to calculate RSA, the training time grows with the increase in model complexity, i.e., about 2.7%, 3.4%, and 9.5% for LeNet, ConvNet, and

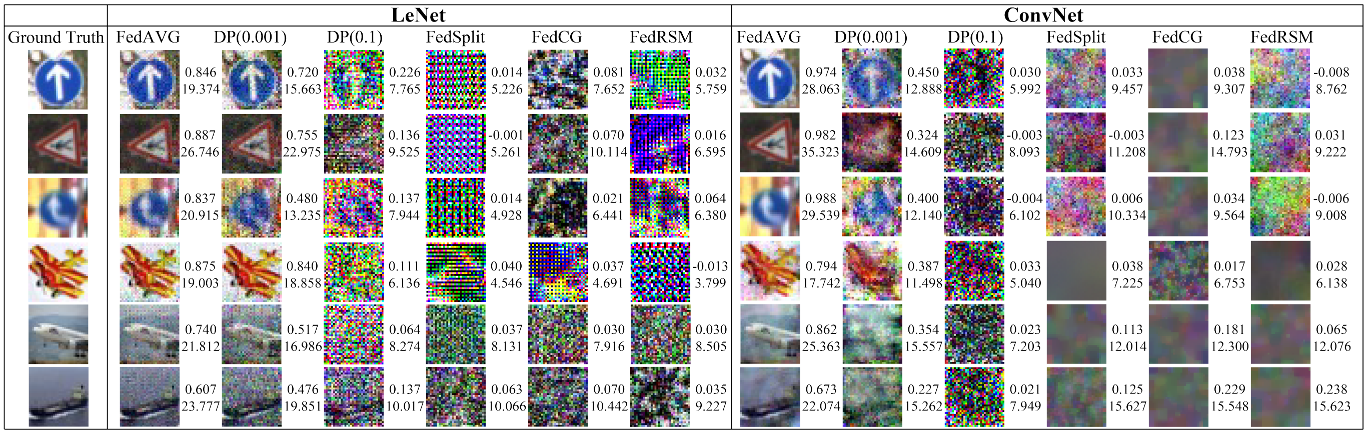

Fig. 4. Comparison of ground truth images and recovered images. The top three images are from German Traffic Sign while the rest are from CIFAR-10. SSIM and PSNR are listed to the right of each picture (the first row refers to SSIM and the second row refers to PSNR with unit dB).

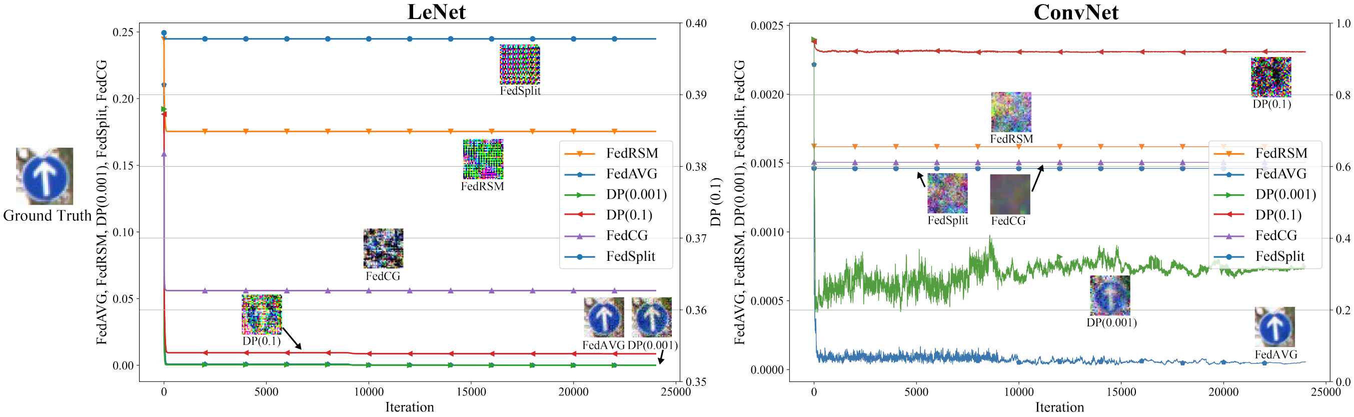

Fig. 5. The loss of attack algorithm (Inverting Gradients) under LeNet and ConvNet.

ResNet18, respectively. However, while comparing FedRSM with FedCG, the training time reduces by about 10x. It indicates that the training of GAN is time-consuming compared to other methods.

The above results show that even though the calculation of RSA can increase the training time slightly compared to the best baseline, the application of RSA in FedRSM, can, indeed, improve learning performance in terms of model accuracy and communication cost.

C. Privacy Evaluation

The comparison between the ground truth images and the recovered images is shown in Fig. 4. The first three rows are from GRS while the rests are from CIFAR-10. The results of LeNet show that FedAVG suffers from the severest data leakage with the highest PSNR and SSIM, followed by DP ($\sigma^2 = 0.001$). With the increase in the noise level, the security protection ability of DP can increase significantly. Finally, FedSplit, FedCG, and FedRSM are able to avoid data leakage, as their values of PSNR and SSIM are the lowest.

To further reveal the ability of these methods in protecting data security, the recovery loss of the IG algorithm (lower loss value refers to higher recovery quality) is presented in Fig. 5. Under both models, the loss of FedSplit, FedCG, and FedRSM almost remain unchanged during the attack. It is reasonable as models uploaded by these methods are incomplete and therefore, result in the malfunction of the attack algorithm. However, for FedAVG and DP, the algorithm is still functional and can recover images, which is more obvious under ConvNet where the loss of both methods fluctuates with the increase of iterations. To be specific, FedAVG leads to a deep leakage of raw data. In terms of DP, the quality of recovered images is related to model complexity and noise level.

D. Summary

While considering the results in performance and security metrics jointly, the proposed FedRSM achieves a more balanced capability in security protection. Specifically, compared to baselines, by maintaining a similar training time, FedRSM can improve test accuracy for up to 2% and significantly reduce communication costs. In the meanwhile, FedRSM is capable of avoiding data leakage under different model complexity.

V. Conclusion

In this paper, we propose FedRSM, a representational-similarity-based secured model uploading for federated learning. Based on representational similarity analysis, for each FedRSM client, it calculates layer-wise representational consistency alterations and constructs the secured local model by masking layers. For the FedRSM server, since the secured local model is incomplete (with some layers absent), it implements a layer-wise model aggregation function to update the global model.

Moreover, by using two standard datasets (i.e., German Traffic Sign and CIFAR-10) to train three DNNs (i.e., LeNet, ConvNet, and ResNet18), the evaluation results demonstrate that with a slight increase in training time (due to the additional calculation of RSA), FedRSM can improve model accuracy for up to 2%, significantly reduce communication cost, and avoid data leakage under different model complexity.

In the future, FedRSM will be enhanced in three aspects, namely: 1) exploring an adaptive algorithm to choose the number of masked layers for optimal performance, 2) simplifying the calculation of RSA or finding an alternative of RSA with less computation complexity to improve training efficiency, and 3) enabling more attacking scenarios to be supported by FedRSM, and in turn, increase its generability in security protection.

Acknowledgment

The research is supported by the GuangDong Basic and Applied Basic Research Foundation (2023A1515012895) and the National Natural Science Foundation of China (62002398).

References

[1] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis *et al.*, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[3] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282.

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[6] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[7] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[8] H. Ren, J. Deng, and X. Xie, "Grnn: generative regression neural network—a data leakage attack for federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–24, 2022.

[9] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 994–1006.

[10] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[11] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[12] N. Kriegeskorte, M. Mur, and P. A. Bandettini, "Representational similarity analysis-connecting the branches of systems neuroscience," *Frontiers in systems neuroscience*, p. 4, 2008.

[13] J. Mehrer, C. J. Spoerer, N. Kriegeskorte, and T. C. Kietzmann, "Individual differences among deep neural network models," *Nature communications*, vol. 11, no. 1, p. 5725, 2020.

[14] S. Liu, Q. Chen, and L. You, "Fed2a: Federated learning mechanism in asynchronous and adaptive modes," *Electronics*, vol. 11, no. 9, p. 1393, 2022.

[15] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[16] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.

[17] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.

[18] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[19] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2587–2596.

[20] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[21] H. Gu, L. Fan, B. Li, Y. Kang, Y. Yao, and Q. Yang, "Federated deep learning with bayesian privacy," *arXiv preprint arXiv:2109.13012*, 2021.

[22] Y. Wu, Y. Kang, J. Luo, Y. He, L. Fan, R. Pan, and Q. Yang, "FedCG: Leverage conditional GAN for protecting privacy and maintaining competitive performance in federated learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, jul 2022.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[24] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–8.

[25] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.