

Mixed Quantization Enabled Federated Learning to Tackle Gradient Inversion Attacks

Pretom Roy Ovi, Emon Dey, Nirmalya Roy, Aryya Gangopadhyay
University of Maryland, Baltimore County (UMBC)
1000 Hilltop Circle, Baltimore, USA
povil, edey1, nroy, gangopad@umbc.edu

Abstract

Federated Learning (FL) enables collaborative model building among a large number of participants without the need for explicit data sharing. But this approach shows vulnerabilities when gradient inversion attacks are applied to it. FL models are at higher risk in the event of a gradient inversion attacks, which has a higher success rate in retrieving sensitive data from the model gradients, due to the presence of communication in their inherent architecture. The most alarming thing about this gradient inversion attack is that it can be performed in such a covert way that it does not hamper the training performance while the attackers backtrack from the gradients to get information about the raw data. Some of the common existing approaches proposed to prevent data reconstruction in the context of FL are adding noise with differential privacy, homomorphic encryption, and gradient pruning. These approaches suffer from some major drawbacks, including a tedious key generation process during encryption with an increasing number of clients, a significant performance drop, and difficulty in selecting a suitable pruning ratio. As a countermeasure, we propose a mixed quantization enabled FL scheme, and we empirically show that issues addressed above can be resolved. In addition, our approach can ensure more robustness as different layers of the deep model are quantized with different precisions and quantization modes. We empirically proved the validity of our defense method against both the iteration based and recursion based gradient inversion attacks and evaluated the performance of our proposed FL framework on three benchmark datasets and found out that our approach outperformed the baseline defense mechanisms.

1. Introduction

Federated Learning (FL) has emerged as an alternative to the centralized approach of building a machine learning

model, which introduces collaborative training among multiple clients while keeping their data private [15, 16]. Although the inherent architecture of FL eliminates the need for explicit data sharing, it still shows vulnerabilities against gradient inversion attacks. Among the present adversaries in the context of FL, the gradient inversion attacks are considered one of the most harmful ones because attackers can successfully reconstitute sensitive training data by secretly snooping on gradient updates during iterative training and without affecting model training quality. Gradient inversion attacks have been investigated extensively, and there are several such methods, e.g., DLG [25], iDLG [23], InvGrad [6], CPL [20], GradInversion [22], R-GAP [24], COPA [3], etc. These attacks can completely reconstruct the training data and/or associated labels from gradients. Some of these attacks are iteration-based and others are recursion-based. The iteration-based attacks aim to minimize the distance between the dummy gradients and ground-truth gradients. Such attacks consider the distance between the gradients as error and the dummy inputs as parameters and so, the recovery process is formulated as an iterative optimization problem. DLG, iDLG, Grad Inversion are the attacks of iteration-based framework. On the other hand, recursion-based attacks are also capable of reconstructing the original data in a closed-form algorithm. The key insight is to exploit the implicit relationships among the input data, model parameters, and gradients of each layer in order to find the optimal solution with the minimum error. R-GAP, COPA are the attacks of such types.

Some of the most explored prevention methods against those attacks in the present literature are Gaussian or Laplacian noise-based differential privacy (DP), gradient pruning, and homomorphic encryption (HE). In the first approach [14,21], to protect the confidentiality of training data, Gaussian or Laplacian noise is added with gradients during training. But in this method, the accuracy deteriorates below the threshold level. The second is ‘gradient pruning [25]’, where a specific pruning ratio is selected during training to make it robust against gradient inversion attacks. But prun-

ing in the initial rounds of FL training is not advisable, as it may cause the loss of fundamental feature-related information. Homomorphic encryption [2, 4] can ensure protection of data confidentiality, but generating unique keys for each client increases computation complexity. In the FL context, where one of the assumptions is that there will be a large number of participants, implementing an HE-based protection method can be tedious. Also, to ensure all clients are getting the same key, we have to enable key sharing among the clients, but such communication is not desirable in FL.

Moreover, researchers have been investigating quantization, typically a model compression scheme to reduce the computation resource requirement of deep models. We point out a new use case of the quantization approach in tackling the gradient inversion attacks¹. In quantization, the gradient values are transitioned into a less precise form according to our choice of bit size. Unless the attacker has some knowledge about the range information of the unquantized gradient, it is highly unlikely to retrieve sensitive raw data information. We have chosen mixed-precision over single-precision quantization to make our resistance algorithm more robust. Because in mixed-precision, for an attacker, the number of iterations to estimate a hyperparameter jumps up to the power of the layer number of the model. Thus, it can make the data-extracting process significantly resource-exhaustive for the attackers.

Our approach can overcome the disadvantages present in the existing methods in the sense that, in our approach, we are ensuring state-of-the-art accuracy without dequantizing at the server end. This strategy will prevent the theft of valuable client information even if the server is attacked, because no information regarding full-precision gradient updates is shared on the server. Also, due to the compressed nature of the quantization, the gradient size required for transmission is smaller, thus minimizing the communication overhead compared to other defense strategies. The specific scientific contributions we offer here are:

- We conduct a detailed risk assessment in a Federated Learning (FL) scenario due to gradient inversion attacks and propose a quantization-enabled solution to ensure a more robust FL framework. Specifically, our approach is built upon the concept of mixed-precision quantization, which is applied to the gradients prior the transmission phase.
- We empirically demonstrate the applicability of our proposed algorithm with three popular FL datasets covering both the iteration and recursion-driven model inversion attacks. Through performing a comprehensive baseline comparison, we achieve an average 15% increase in accuracy while keeping the attack success rate to zero.

¹<https://github.com/PretomRoy/Defense-against-grad-inversion-attacks>

- We present a pertinent ablation study to determine the impact of different hyperparameters used in our federated framework. We find that along with the traits of attack resiliency and accuracy retention, our method can offer another desirable property of reduced communication cost.

2. Related work

There can be numerous adversarial motives for inferring private information, including data reconstruction. The purpose of data reconstruction is to reveal training samples utilized by participating clients. In federated learning, the server and clients exchange gradients after each round of training. According to [5, 13], gradients reveal some properties of the training data. Recently, the authors of [25] proposed an algorithm named DLG to reconstruct training samples from the gradients. Then, iDLG [23] is proposed to improve the efficiency and accuracy of DLG. So, existing studies [6, 20, 23, 25] demonstrate that information can be leaked from model updates shared between the server and clients during FL training. Existing solutions against gradient inversion attacks can be divided into a few groups: differential privacy [7, 14, 21], encryption [4, 8], and gradient pruning [25].

Recently, client-level differentially private federated learning are proposed [7, 12] via injecting Gaussian or Laplacian noise to local updates. However, these DP-based methods demand a large number of participants in the training process to converge, resulting in a trade off between data confidentiality and performance. In cross-silo federated setups, it may not be possible to have a large number of clients. Then, the effectiveness of DP is reliant on each client having access to a large dataset, according to [18]. However, the FL setup cannot guarantee that every client will have a large training dataset, and the quantity of available data can differ between clients. In [19], authors have demonstrated the use of Bayesian DP in FL in a context where the data is distributed similarly among the participating clients. However, FL cannot ensure that the data distribution will be similar for all clients. While DP provides a certain level of differential privacy guarantee, it can limit the performance accuracy of deep learning models. Furthermore, careful parameter selection is essential for DP, or else there is a risk of gradient induced data breach.

The encryption algorithms often used in FL can be broadly classified as Homomorphic Encryption (HE) [2, 4] and Secure Multi-party Computing (SMC) [8, 11]. While preserving the confidentiality of training data, HE theoretically ensures no performance loss in terms of model convergence [2, 4]. However, in the context of FL, assigning individual keys for each participating client is a cumbersome task, and data-size of the encrypted models increases linearly with each homomorphic operation [1, 17], hence limiting its applicability in this use case. On the other hand,

Secure Multi-party Computing (SMC) in FL scenarios incurs computational overhead and also requires each worker to coordinate with others during the training process, which is not desirable in FL. Furthermore, the authors of [25] attempt to defend against attacks by using gradient pruning and sparsification. However, such approaches require a high pruning rate from the initial training epoch, which may lead to poor model performance.

3. Methodology

In this section, we describe the detailed working procedure of our proposed federated learning approach along with integration of mixed quantization.

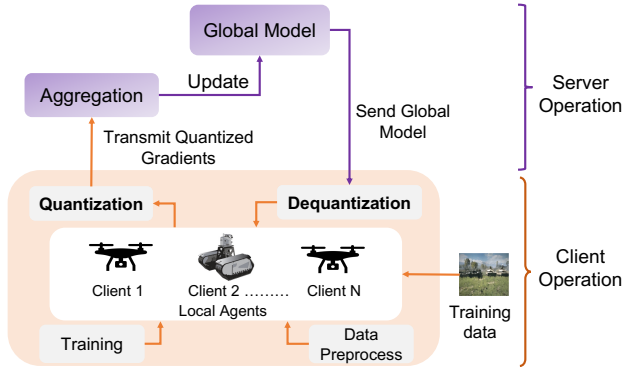


Figure 1. Overview of proposed FL framework.

We describe the individual operations carried out on clients (data owner) and server (model owner), the two components of a FL system. Some of the notations used in this algorithm description are $\mathcal{N} = \{1, \dots, N\}$ signifying the set of N clients, each of which has its own dataset $D_{k \in \mathcal{N}}$. Each of them trains a local model using its local dataset and only shares the model gradients to the FL server. Then, the global model formation takes place combining all the local model gradient updates. The illustration of our proposed framework is shown in figure 1 and complete pseudocode of our method is shown in algorithm 1. The elaboration of the client and server-side operations is given below:

1. Executed at the server side

- *Weight initialization:* The global model \mathbf{W}_G^0 and its hyperparameters are disseminated from the server side. In our case, the application we chose is image classification and the LeNet-5 model is initiated with randomized weight during the first training round. As soon as the first round finishes, the initial weights are quantized and updated with the aggregated quantized gradients sent from the clients.
- *Aggregation and global update:* The server first aggregates the quantized gradient updates from each client. The formula for the gradient aggregation through averaging at the t^{th} iteration is given below:

Algorithm 1 Proposed Federated Learning algorithm

Require: Clients number n per iteration, learning rate η , local dataset D , local minibatch size B , and total number of iteration T

Ensure: Global model \mathbf{W}_G .

- 1: [Step 1](Server)
- 2: Initialize \mathbf{W}_G^0
- 3: [Step 2](Client)
- 4: **LocalTraining**(i, W) : \triangleright (Training with W_G^0 during first round)
- 5: Split local dataset D_i to minibatches of size B which are included into the set \mathcal{B}_i .
- 6: **for** each local epoch **do**
- 7: **for** each $b \in \mathcal{B}_i$ **do**
- 8: $\Delta W(\mathbf{W}; b) = \Delta L_{CE}(\mathbf{W}; b)$ $\triangleright \Delta W$ is the gradient on b .)
- 9: **end for**
- 10: **end for**
- 11: **Gradient Quantization:** $\Delta W_q \leftarrow \text{Quantize}(\Delta W)$
- 12: [Step 3](Server)
- 13: **Gradient Aggregation:**
- 14: $\Delta W_{qG}^t = \frac{1}{\sum_{i \in \mathcal{N}} |D_i|} \sum_{i=1}^N D_i \Delta W_{q_i}^t$ \triangleright (Aggregation through average)
- 15: **Updater:**
- 16: Updated Global model, $W_{qG}^t = W_{qG}^{t-1} - \eta \Delta W_{qG}^t$
- 17: [Step 4](Client)
- 18: **Gradient Dequantization:**
- 19: $W_G^t \leftarrow \text{Dequantize}(W_{qG}^t)$
- 20: **for** each iteration t from 2 to T **do**
- 21: Randomly choose a subset \mathcal{S}_t of n clients
- 22: **for** each client $i \in \mathcal{S}_t$ **paralelly do**
- 23: $\Delta W_i^{t+1} \leftarrow \text{LocalTraining}(i, W_G^t)$
- 24: **end for**
- 25: **end for**

$$\Delta W_{qG}^t = \frac{1}{\sum_{k \in \mathcal{N}} |D_k|} \sum_{k=1}^N D_k \Delta W_{q_k}^t \quad (1)$$

As we are not dequantizing the updates at the server side, the server should not have the information to extract the exact full-precision values of quantized gradients. Thus, this approach minimizes the chance of retrieving the raw data from the gradients even if there is an attack on the server side. The aggregated gradient updates, ΔW_{qG}^t are multiplied with the learning rate η to achieve the updated global model, W_{qG}^t which is used to check the performance. This process is repeated until the global loss function converges or a desirable training accuracy is achieved. We have found mini-batch SGD [9] optimizer worked better during the training process for our specific use case.

2. Executed at the client side

In the first training round, each client initiates the local training upon receiving the unquantized global model from the server. The client tries to minimize the loss function [10] L_{CE} and generates the gradient of each data batch b , ΔW_k^t .

$$L_{CE} = - \sum_{i=1}^C y_i \log(p_i), \text{ for } C \text{ classes} \quad (2)$$

where y_i is the truth label and p_i is the Softmax probability for the i^{th} class. In our case, we have used the unquantized W_G^0 as an argument for the local training during first round. This round will be used to generate the initial gradients for each client. For the second round on-wards, all participating clients receive the updated quantized weights from server and received quantized weights will be dequantized before running the ‘Local Training’ function. As soon as the training is completed on the client side, the model gradients are quantized to ΔW_q^t , where t stands for each iteration index, using scalar quantization and transmitted to the server. This step prevents the framework from the gradient inversion attacks as training samples cannot be reconstructed from the quantized gradients. Even if the attackers try to de-quantize it to extract the ground truth gradients, the permutations required to perform the dequantization and retrieve the raw data make the process very arduous for the attackers.

3. Quantization and Dequantization

The quantization approach we have implemented here depends on three major tuning parameters:

- Minimum and maximum range of the gradient updates
- Mode of Quantization
- Mechanism for rounding the float values to their quantized equivalents

Upon selecting the process by setting up the ‘mode’ attribute, we determine which calculations are carried out in order to convert the float values into the quantized equivalents of those values. We vary the quantization ‘mode’ and ‘rounding mechanism’ of float values for each layer of the model, and this information is only kept locally for the quantization and dequantization processes. This approach brings us two-fold advantages in terms of maintaining client data confidentiality.

- If a specific client is compromised, the range information of the gradient of that client and information on the ‘mode’ variable may be leaked. But this leaked information from one client cannot hamper the data confidentiality of other participating clients as the compromised client doesn’t have access to the minimum and maximum range information of the gradients of another participant.

- Even if the server is vulnerable to inversion attacks, the information about each client can still be kept safe. This is because the minimum and maximum ranges of the gradient updates, which are one of the fundamental parameters for dequantization, aren’t shared with the server.

The elaboration of our chosen quantization approach is as follows:

Scaling factor determination is the key factor during the quantization process. Since we have implemented mixed-precision quantization, the bit size that is used for each layer has a different range of options available to it. The scaling factor ensures that all the values which fall within the minimum and maximum ranges of full precision gradients are able to be represented by chosen bit size of the output tensor. After setting the scaling factor, it is utilized to make the modifications to the minimum and maximum ranges. As soon as these steps are completed, a quantized version of the input tensor can be obtained by clipping the values to the minimum and maximum range (rounding mechanism) and then multiplying by the scaling factor. The dequantization process uses the same set of parameters to convert the quantized values into their full-precision forms according to the minimum and maximum ranges of unquantized gradients.

4. Experiment Setup and Result Analysis

This section will step through the detailed experiment setup and result analysis.

4.1. Experiment Setup

Attack Methods: In this experiment, we aimed to implement both iteration-based and recursion-based gradient inversion attacks. The iteration-based attacks, including DLG [25], iDLG [23], InvGrad [6], CPL [20], and GradInversion [22], were launched by minimizing the distance between the dummy gradients generated by the dummy data and the real ones. On the other hand, the recursion-based attacks, such as R-GAP [24], were launched by exploiting the implicit relationships among the input data, model parameters, and gradients of each layer to find the optimal solution with the minimum error. These attacks were implemented using different convolutional neural network (CNN) architectures, such as LeNet for DLG, iDLG, and CPL attacks, ResNet for Grad Inversion, and ConvNet for R-GAP attacks. It is worth noting that some of the attack methods can reconstruct a single training sample, whereas some can reconstruct batch samples from the gradients.

Baseline Defenses: We conducted a comparative analysis of our proposed method with existing baseline defense strategies, namely Gradient Pruning and Differential Privacy. Gradient Pruning is a method that prunes gradi-

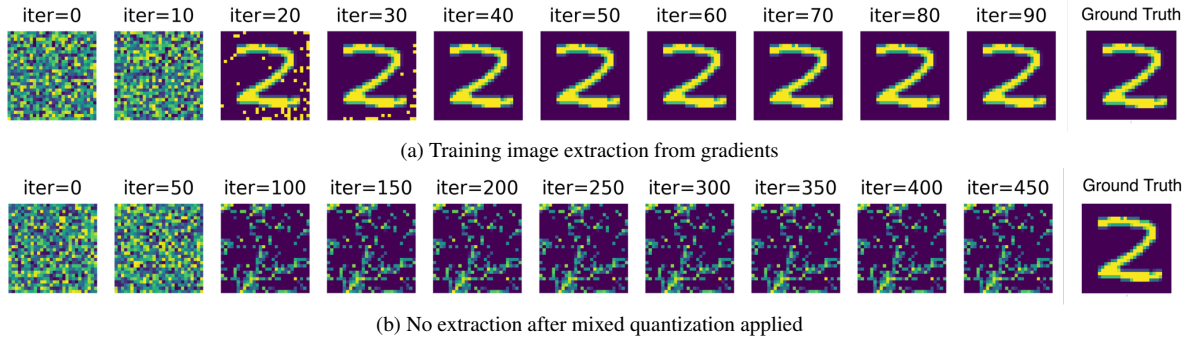


Figure 2. Reconstructed sample after applying DLG attack. (a) Training sample retrieved within 30 iterations from raw gradients, (b) No extraction from quantized gradients.

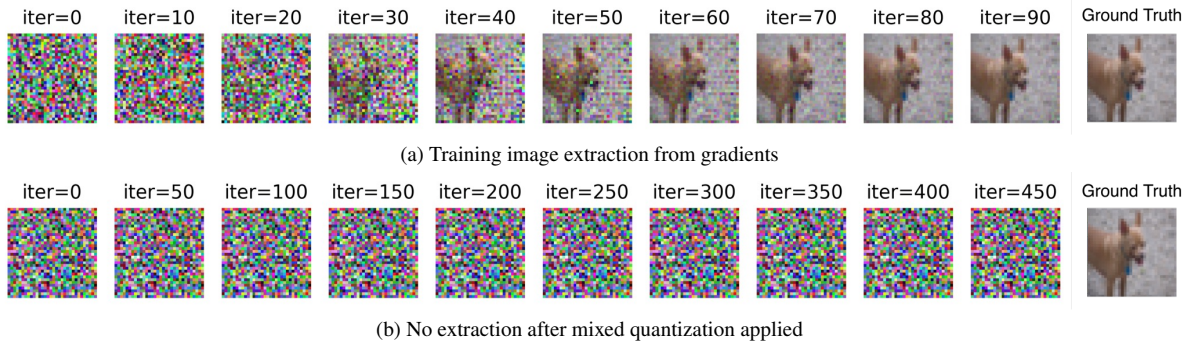


Figure 3. Reconstructed sample after applying InvGrad attack. (a) Training sample retrieved within 40 iterations from gradients, (b) No extraction from gradients when mixed quantization applied.

ents smaller than a certain threshold, while Differential Privacy adds noise to gradients. However, it is worth noting that most DPSGD implementations for image classification tasks use a pre-training and fine-tuning pipeline, making it challenging to compare them to other defense techniques that can be applied directly during model training. As a result, we developed two DP baselines by separately applying Gaussian and Laplacian noise, which we refer to as DP-Gaussian and DP-Laplacian.

Datasets: To evaluate the attack resiliency and performance of our proposed approach, we conducted experiments on MNIST, FASHION-MNIST, and CIFAR10 datasets.

FL Training Setup and Configuration: Experiments were conducted on a system with an Intel(R) Core(TM) i9-11900K CPU (8 cores) and an NVIDIA GeForce RTX 3090 GPU. The federated learning framework was implemented using Keras with a TensorFlow backend. The server controlled the training pace, including the number of epochs per round and the overall number of rounds. The server sets the pace of the training, determines the number of epochs per round, and how many rounds of overall training are to be conducted. We utilized the LeNet-5 CNN architecture as a global model and simulated FL training with 15 clients.

For each round, 1 epoch of local training is conducted on the client side. We used SGD optimizer and set the learning rate η to 0.01 during training. We employed mixed quantization, where the gradients of different layers of the deep model were quantized with different precision and quantization bits. We utilized the combination of int8 and int16 bit quantization to keep the accuracy in the loop. And to initiate the attacks, we used the L-BFGS and Adam optimizer and conducted up to 500 iterations of optimization to reconstruct the raw data.

4.2. Result Analysis

We first showcase the gradient inversion attacks to reconstruct the training data. Figure 2 and figure 3 depict the reconstruction of training samples from the gradients with DLG and InvGrad attack methods respectively. And we determine that recovering monochromatic images with a clean background (MNIST) is easier, whereas recovering relatively complex images (CIFAR-10) requires more iterations. Then, we point out the capability of the proposed mixed quantization in tackling the gradient inversion attacks. In figure 2b and figure 3b, training images are not retrieved from the quantized gradients even after 450 iterations of distance minimization, whereas training samples

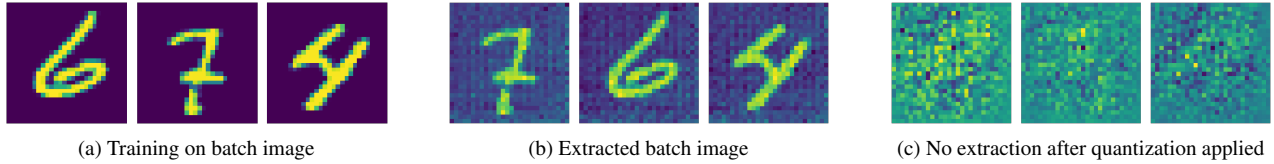


Figure 4. Extracted batch data after applying CPL attack.

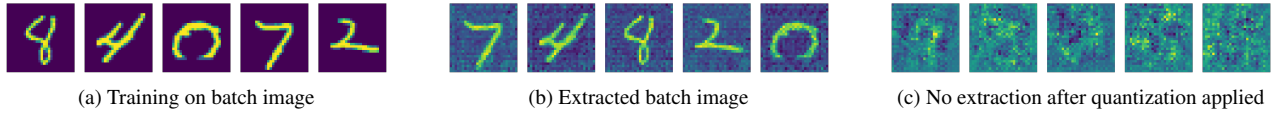


Figure 5. Extracted batch data after applying GradInversion attack.

are retrieved within 40 iterations when gradients are not quantized, shown in figure 2a and figure 3a. In a similar vein, figure 4 and figure 5 depict the reconstruction of batch data from gradients with CPL and GradInversion attacks, respectively. From figure 4c and figure 5c, it is observed that not even a single sample from the batch is extracted by these attacks when the gradients are quantized. In addition to iteration-based gradient inversion attacks, recursion-based attacks, such as R-GAP, are also implemented as shown in figure 6. R-GAP extracts the input training image from the gradients by exploiting the relationship among the input data and gradients, depicted in figure 6b. However, when our mixed quantization-based defense strategy is implemented, R-GAP is unable to extract the training sample, as illustrated in figure 6c.

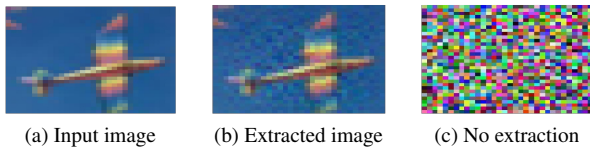
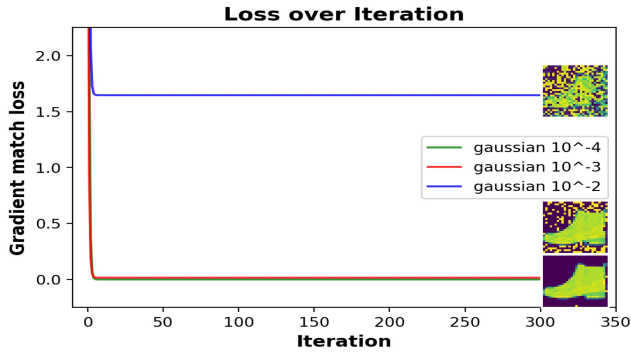


Figure 6. Extracted sample after applying R-GAP attack.

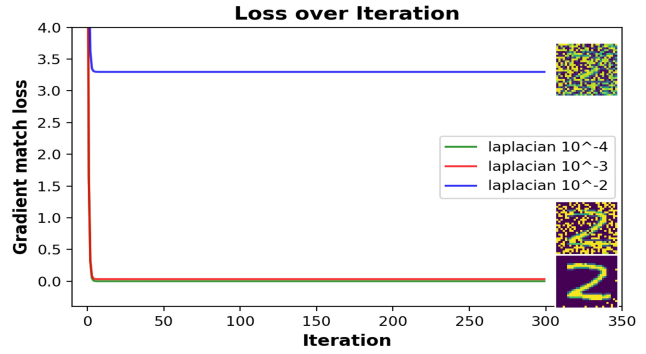
We then compare the effectiveness of our proposed approach with existing baseline defense strategies. One straightforward attempt to prevent the attack is to add noise to gradients prior to share with the server. To evaluate, we experimented with Gaussian and Laplacian noise (widely used in differential privacy studies) variance range from 10^{-1} to 10^{-4} with mean 0. Figure 7 depict the impact of varying noise levels against the attacks. It is observed that when variance is at the scale of 10^{-4} or 10^{-3} , the noisy gradients do not prevent the attacks. For noise with variance level 10^{-2} , though with artifacts, the reconstruction can still be performed. We found out that added noise with variance level larger than 10^{-2} can defend attacks. So, Gaussian/Laplacian noise with a minimum variance level of 10^{-1} should be added to the gradients to defend against

the attacks. However, this amount of noise has a significant degradation in performance in federated training.

Another baseline strategy to prevent the attacks is gradient pruning, depicted in figure 8. From figure 8a, it is observed that pruning in the range of 1% to 10% can not prevent the data reconstruction because the reconstructed image reveals the data and is easily recognizable. For gradients with 20% pruning, though with artifacts, the attack is still successful. But when 30% pruning is applied on gradients, the attack is prevented. So, gradients with above 30% pruning ratio is able to prevent the attacks for Cifar-10 data. But it requires higher pruning ratio for Mnist and Fashion Mnist dataset. From figure 8b, required pruning ratio is 60% for Fashion-Mnist data and above 70% for Mnist data to defend against the attacks. It means that less pruning ratio is required for the complex training samples whereas for monochromatic images with a clean background (e.g., MNIST), it requires higher pruning rate to prevent the attacks. So, the required pruning ratio to prevent attacks may vary depending on the complexity of the training data and that ratio needs to be selected through iterative experimentation. Moreover, the typical method of pruning entails train the network first, then prune the less important part of network by setting it to 0, and finally fine-tune the network. This process involves the removal of the least significant neuron. To incorporate pruning into federated learning (FL) framework as a defense mechanism, the network must be pruned from the initial round of training prior to transmitting the local gradient updates to the server. But in the first training round, the model cannot determine perfectly which neurons are significant and which are not. So, pruning the network from the first epoch of training is not recommended and may result in poor training performance. Moreover, convergence is an issue in federated training and clients may have local training data in an imbalanced fashion with varying class distributions. Therefore, pruning the network during training, particularly in a federated setup, may result in no convergence at all.

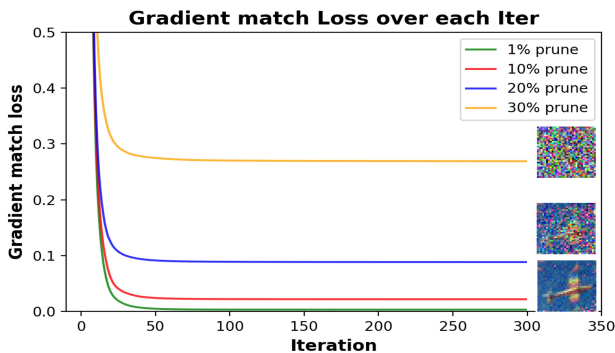


(a) Different magnitude Gaussian noise.

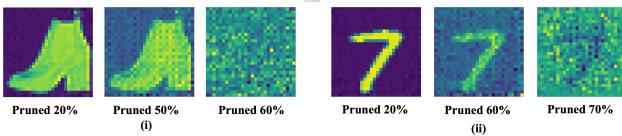


(b) Different magnitude Laplacian noise.

Figure 7. Effect of Gaussian and Laplacian noise against attacks.



(a) Effect of pruning on Cifar10 sample.



(b) Effect of pruning on (i) FashionMnist, (ii) Mnist

Figure 8. Effect of different pruning ratios against attacks.

Performance Evaluation: Finally, we compare the performance of proposed mixed quantization enabled FL with baseline defense strategies by incorporating them in FL framework, results are shown in table 1. And we find that our approach outperforms the baseline defense strategies. We report the performance of the DP-Gaussian and DP-Laplacian approach with variance level 10^{-1} because this level of noise is the least to prevent the attacks. In comparison with base FL (FL with no defense against gradient inversion attacks), our proposed approach achieves almost the same level of performance- only 0.5% drop for MNIST, and around 1% drop for CIFAR10 and FASHION MNIST datasets, demonstrated in table 1. On the other hand, DP-Gaussian based FL degrades around 12% for MNIST and FASHION MNIST, and 18% for CIFAR10. DP-Laplacian based FL has even more degradation in performance. So,

our approach outperforms the DP-Gaussian based FL by 13% on average and the DP-Laplacian based FL by 20% on average. We also evaluate the quality of the extracted images by gradient inversion attacks, in comparison to the ground truth images, both before and after implementing the proposed defense mechanism, shown in table 2. We utilize PSNR as metric to compute the peak signal-to-noise ratio between the original and constructed images. The higher the PSNR, the better the quality of the reconstructed image.

Methods	Dataset		
	Mnist	Fashion Mnist	Cifar10
Base FL (Without any defense)	97.05	86.8	60.04
FL with DP-Gaussian (variance= 10^{-1})	85.54	73.8	42.64
FL with DP-Laplacian (variance= 10^{-1})	78.66	66.19	33.41
Mixed Quantization enabled FL (Proposed)	96.67	85.22	58.9

Table 1. Accuracy comparison with baseline defense mechanisms.

Attack Methods	Without defense	With proposed defense
DLG	55.31126	8.155
InvGrad	23.282996	7.89
R-GAP	13.526749	8.6

Table 2. PSNR comparison between extracted images without any defense mechanism and after applying our proposed defense.

5. Ablation Study

In this section, we analyze our framework from two different points of view. While implementing our quantized framework, we employ a hyperparameter named ‘Quantization mode’. This parameter determines which calculation procedure will be used to determine the modified maximum and minimum data range. It is varied across layers, keeping accuracy in the loop. The mode that was used for quantization must be selected for dequantization to get the ground truth gradients otherwise dequantized gradients will have a significant mismatch. To showcase this, we illustrated in figure 9 that the ground truth image cannot be recovered

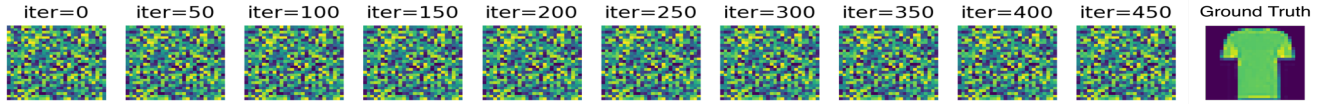


Figure 9. Failure in image recovery when dequantized modes are mismatched with quantized modes.

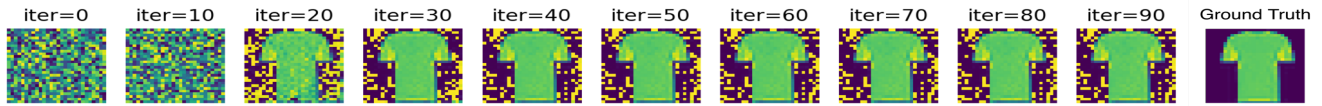


Figure 10. Recovered image from dequantized gradients with negligible noise in the background.

even from dequantized gradients if generated by a different mode whereas ground truth image is recovered in figure 10 if generated by the same mode that was used for quantization. But this recovered ground truth image has a few negligible noises in the background pixels because dequantization is not a fully reversible process.

Secondly, the mixed-precision quantization ensures higher security against the attacks aimed at revealing the actual gradients through dequantization. There are certain number of ‘modes’ for quantization and dequantization. The set of operations that is used to quantize, the same set of reverse operations is required during dequantization to get the ground truth gradients. In mixed quantization, each layer can be quantized with separate set of quantization modes and quantization bits. Thus the required permutations to perform dequantization to retrieve ground truth gradients are exponentially high that make the process very arduous for the attackers. For example, for single-bit quantization, the attacker needs to try m combinations of dequantization to get the ground truth gradients whereas for mixed bit quantization, the attacker needs to try m^L combinations to crack it, where m is the number of quantization mode and L is the number of layers in any deep model.

6. Discussions

Our proposed mixed quantization-based gradient inversion tackling technique can readily be extended to address two other important research problems in the FL context.

6.1. Resource Efficiency

Minimizing communication overhead during federated training is another vital aspect to consider. Our proposed approach can also be seen as a communication-efficient federated framework. In our approach, the low-bit quantized gradients are shared between the server and clients rather than transmitting the gradients of float32. For instance, in our case, converting the precision of activation and gradients from 32-bit floats (model size 174 kB) to 8-bit integers (44 kB) results in $4\times$ data reduction, which eventually requires 4 times less transmission bandwidth. Quantized gradients reduce downstream and upstream communication

costs and thus speed up training.

6.2. Adaptive Quantization Parameter Selection

To ensure robustness in terms of accuracy for an imbalanced dataset along with defense against gradient inversion attacks, our quantization approach can be modified into a class-specific one. While implementing the quantization-based compression algorithm in the federated setting, it is possible that some weight values of the specific classes can be eliminated. To deal with this problem and prevent information loss, the compression algorithm will include a class-specific distribution monitoring scheme to make sure that enough information about each class is available during the training time. On the client side, as a future direction, the federated learning framework can be equipped with the ability to determine the class specific data distribution after each training round.

7. Conclusion

In this paper, we propose a mixed quantization enabled federated learning technique to prevent gradient inversion attacks and ensure data confidentiality. We explore both variants of gradient inversion attacks, namely iteration and recursion-based, to ensure a generalized solution. Furthermore, our proposed approach transforms the high-precision gradients to low-bit precision, resulting in faster training, less transmission bandwidth, and lower communication costs. Experimental results on three benchmark datasets demonstrate that our approach outperforms all the baseline approaches in terms of accuracy while preserving the data confidentiality. To conclude, we envision that the proposed federated training framework will have a high impact in cyber-physical applications where high resilience against gradient inversion attacks and competitive accuracy are primary requirements.

ACKNOWLEDGMENT

We acknowledge the support of the U.S. Army Grant #W911NF21-20076, NSF grant #1923982 and ONR grant #N00014-23-1-2119.

References

- [1] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35, 2018. 2
- [2] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2017. 2
- [3] Cangxiong Chen and Neill DF Campbell. Understanding training-data leakage from gradients in neural networks for image classification. *arXiv preprint arXiv:2111.10178*, 2021. 1
- [4] Haokun Fang and Quan Qian. Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet*, 13(4):94, 2021. 2
- [5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 2
- [6] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020. 1, 2, 4
- [7] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017. 2
- [8] Yong Li, Yipeng Zhou, Alireza Jolfaei, Dongjin Yu, Gaochao Xu, and Xi Zheng. Privacy-preserving federated learning framework based on chained secure multiparty computing. *IEEE Internet of Things Journal*, 8(8):6178–6186, 2020. 2
- [9] Zengpeng Li, Vishal Sharma, and Saraju P Mohanty. Preserving data privacy via federated learning: Challenges and solutions. *IEEE Consumer Electronics Magazine*, 9(3):8–16, 2020. 3
- [10] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020. 4
- [11] Jun Liu, Yuan Tian, Yu Zhou, Yang Xiao, and Nirwan Ansari. Privacy preserving distributed data mining based on secure multi-party computation. *Computer Communications*, 153:208–216, 2020. 2
- [12] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017. 2
- [13] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019. 2
- [14] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*, 2020. 1, 2
- [15] Pretom Roy Ovi, Emon Dey, Nirmalya Roy, Aryya Gangopadhyay, and Robert F Erbacher. Towards developing a data security aware federated training framework in multi-modal contested environments. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, volume 12113, pages 189–198. SPIE, 2022. 1
- [16] Pretom Roy Ovi, Aryya Gangopadhyay, Robert F Erbacher, and Carl Busart. Secure federated training: Detecting compromised nodes and identifying the type of attacks. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1115–1120. IEEE, 2022. 1
- [17] Mohammad Saidur Rahman, Ibrahim Khalil, Mohammed Atiqzaman, and Xun Yi. Towards privacy preserving ai based composition framework in edge networks using fully homomorphic encryption. *Engineering Applications of Artificial Intelligence*, 94:103737, 2020. 2
- [18] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013. 2
- [19] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2587–2596. IEEE, 2019. 2
- [20] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursay, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*, pages 545–566. Springer, 2020. 1, 2, 4
- [21] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 797–807. IEEE, 2021. 1, 2
- [22] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021. 1, 4
- [23] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 1, 2, 4
- [24] Junyi Zhu and Matthew Blaschko. R-gap: Recursive gradient attack on privacy. *arXiv preprint arXiv:2010.07733*, 2020. 1, 4
- [25] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3, 4