DEFENDING AGAINST BACKDOOR ATTACKS IN FEDERATED LEARNING WITH DIFFERENTIAL PRIVACY

Lu Miao[1], *Wei Yang*[1,⋆], *Rong Hu*[1], *Lu Li*[2], *Liusheng Huang*[1]

[1]University of Science and Technology of China
[2]School of Information Engineering, Yancheng Teachers University

ABSTRACT

The training process of federated learning is known to be vulnerable to adversarial attacks (e.g., backdoor attack). Previous works showed that differential privacy (DP) can be used to defend against backdoor attacks, yet at the cost of vastly losing model utility. To address this issue, we in this paper propose a defense method based on differential privacy, called Clip Norm Decay (CND), to maintain utility when defending against backdoor attacks with DP. CND reduces the injected noise by decreasing the clipping threshold of model updates through the whole training process. In particular, our algorithm bounds the norm of malicious updates by adaptively setting the appropriate thresholds according to the current model updates. Empirical results show that CND can substantially enhance the accuracy of the main task when defending against backdoor attacks. Moreover, extensive experiments demonstrate that our method performs better defense than the original DP, further reducing the attack success rate, even in a strong assumption of threat model.

Index Terms— differential privacy, backdoor attack, federated learning, adversarial machine learning

1. INTRODUCTION

Nowadays, more and more machine learning systems need to collect a large amount of personal information, and privacy preservation has become a hot issue. In order to protect privacy, the concept of federated learning (FL) [1, 2] was proposed, in which a client does not provide its dataset, but downloads the model from the server to the local device for training and uploads model updates.

However, the process of local training is not under the control of the server and therefore is highly vulnerable to attacks in presence of malicious clients. Attackers may tamper with their datasets to inject a backdoor into the model. Backdoors [3] are hidden patterns learned by a model, misleading the model to output wrong labels when inferring samples with backdoor features. Backdoor features can be existing features in training data, or patterns designed by the attackers.

Traditional backdoor detection methods either assume an IID setting [4, 5], which is not realistic for FL, or require the defender to access training data or the final model [6], violating the privacy principle of FL. Differential privacy (DP) [7] was originally used to provide different levels of privacy guarantee, i.e., record-level [8] and user-level [9, 10]. Recently, there has been some research on the defense against backdoor attacks with DP. Weak DP [11] could reduce the success rate of backdoor attacks to a relatively low level. CDP and LDP in [12] could further reduce the success rate by injecting much more noise, whereas at the cost of decreasing the main task accuracy heavily.

In this paper, we propose a method to maintain a high accuracy on the main task when defending against backdoor attacks with DP. The method is called Clip Norm Decay (CND), which decreases the original clipping threshold of model updates before a round starts, and sends the new threshold with the global model to selected clients. Since the magnitude of injected noise is proportional to the clipping threshold, reducing the clipping threshold can introduce less noise and obtain a higher model accuracy.

We implement our method against two kinds of backdoor attacks: single-pixel attack and semantic backdoor attack, under the assumption that the attacker can modify the training dataset and the training process of malicious clients. Experimental results show that, CND indeed greatly improves the accuracy on the main task (more than 20%), making it possible to apply DP under a low privacy budget while maintaining model utility. Besides, it also reduces the success rate of backdoor attacks compared with the original DP, on account of CND suppressing the norms of malicious updates throughout the whole training process.

2. DIFFERENTIAL PRIVACY WITH CLIP NORM DECAY

2.1. Threat Model

In our threat model, we assume that the server is honest, and an adversary controls a subset of clients, called malicious

⋆ Corresponding author: qubit@ustc.edu.cn

This work was supported by the National Natural Science Foundation of China (No. 62172385) and the Anhui Initiative in Quantum Information Technologies (No. AHY150300).

(poisoned) clients. The adversary aims to make the global model misclassify the samples with backdoor features into a wrong label assigned by the adversary. In addition to achieving the backdoor task as accurately as possible, the adversary tries to maintain a high accuracy on the main task as well, because a model with a high accuracy on both main task and backdoor task can hardly be detected as abnormal.

Since the adversary controls a subset of clients in FL, it knows the model architecture and training parameters shared by all clients, e.g., learning rate, batch size, and the number of local epochs. We assume the adversary in the semantic backdoor attack can modify the training data of poisoned clients, and manipulate the training process like *model poisoning attack* [13], which is a strong assumption of the adversary's capability. As a supplementary, we assume the malicious clients in the single-pixel attack only modify the labels of backdoored samples in their datasets, like *data poisoning attack* [14].

2.2. Algorithm Description

Our algorithm is based on user-level DP proposed by [10], which clips the model update computed at each batch of the data, and then perturbs the aggregated update at the server. Previous work [12] showed that user-level DP could defend against backdoor attacks. However, this method still leads to a great loss of the main task accuracy.

In order to further improve the accuracy of the model while defending against backdoor attacks, we propose a method, termed Clip Norm Decay (CND), to decrease the clipping threshold of model updates in DP as the training goes on. Concretely, we initialize a clip norm threshold c_0 before the training starts and send the threshold along with the global model to selected clients at each round. The clients will compute their local model update at each batch and clip the update using the threshold c_0 if the norm of update exceeds it. As the number of rounds increases, the server will decrease the threshold to a new value c_t, and send it to selected clients in the later rounds. The whole algorithm is illustrated in Alg. 1, where we use Moments Account [8] to compute the privacy budget spent at the beginning of each round, and accumulate the privacy loss after each round.

Our intuition is based on the fact that, as the number of rounds increases, the norm of model update gradually decreases. The reasons are as follows: i) the decrease of the loss leads to the reduction of the gradient; ii) the learning rate decays gradually. A smaller clipping threshold means less noise injected and a higher model accuracy. Therefore, we propose to decrease the clipping threshold as the training goes on.

Alg. 2 describes the process of computing a new threshold after each round. The server first multiplies the threshold by the decay coefficient as a default value, and then calculates the average norm of each client's update. If the average norm is smaller than the default value, the server will set it as the new threshold. Since the model updates are clipped before uploading, the average norm will be no greater than the current threshold. The purpose of this step is to make the threshold adaptively drop. If the initial threshold is too large, the average norm will be much lower than the default value, so the threshold will decrease rapidly. Accordingly, if the initial threshold is small, most updates will be clipped and the clipping threshold will drop slowly. The clipping threshold falls in a reasonable range after the first few rounds, and then only needs to be adjusted at certain intervals. The average norm is also perturbed so as not to reveal privacy, but the scale of noise can be different from Alg. 1.

Algorithm 1 Differential privacy with CND in federated learning.

Input: z: noise scale, ϵ: target privacy budget, δ: target delta, γ: decay coefficient, M: number of clients per round, N: number of total clients;
Output: global model θ;

procedure SEVER EXECUTION
  Initialize: model θ^0, clip norm c^0, Moments Account $\mathcal{M}(\delta, M, N)$;
  for each round $t = 0, 1, 2, ...$ **do**
    if $\epsilon < \mathcal{M}$.get_privacy_spent() **then**
      return θ^t
    $Z_t \leftarrow$ random set of M clients;
    for each client $k \in Z_t$ in parallel **do**
      $\Delta_k^{t+1} \leftarrow$ CLIENT UPDATE(k, θ^t, c^t)
    $\sigma = \frac{z}{M}$
    $\theta^{t+1} \leftarrow \theta^t + \frac{1}{M}\sum_{i=1}^{M} \Delta_i^{t+1} + \mathcal{N}(0, (c^t \cdot \sigma)^2)$
    $\mathcal{M}$.accumulate_spent_privacy(z)
    $c^{t+1} \leftarrow$ NEW THRESHOLD(c^t, γ, t)

function CLIENT UPDATE(k, θ^t, c^t)
  $\theta \leftarrow \theta^t$
  for each local epoch $i = 1, 2, ...E$ **do**
    for batch $b \in B$ **do**
      $\theta \leftarrow \theta - \eta \nabla L(\theta, b)$
      $\Delta \leftarrow \theta - \theta^t$
      $\theta \leftarrow \theta^t + \Delta \min(1, \frac{c^t}{\|\Delta\|_2})$
  return $\theta - \theta^t$

Algorithm 2 Computing new threshold by CND.

function NEW THRESHOLD(c^t, γ, t)
  $c^{t+1} \leftarrow \gamma c^t$
  if $t < 10$ or $t = 50, 100, ...$ **then**
    $c \leftarrow \frac{1}{M}\sum_{i=1}^{M} \|\Delta_i^{t+1}\|_2 + \mathcal{N}(0, (c^t \cdot \sigma)^2)$
    $\mathcal{M}$.accumulate_spent_privacy(z)
    if $c < c^{t+1}$ **then**
      $c^{t+1} \leftarrow c$
  return c^{t+1}

2.3. Theoretical Analysis

The perturbing process is conducted at the server in our method and does not rely on clients. Besides, if the attacker refuses to clip its update with the given threshold, it will be detected immediately. Hence, the malicious clients can not quit DP by skipping the process of clipping and perturbing their updates. Next, we show that our algorithms satisfy DP.

Definition 1. *A randomized mechanism $M : D \to R$ provides (ϵ, δ)-differential privacy if for any two neighboring databases, D_1 and D_2, which differ in only a single record, and for any subset of outputs $S \subseteq R$, it holds that*

$$\Pr[M(D_1) \in S] \leq e^{\epsilon} \Pr[M(D_2) \in S] + \delta \qquad (1)$$

Theorem 1. *DP with CND satisfies (ϵ, δ)-differential privacy.*

Proof. In Gaussian mechanism, noise with the normal distribution $\mathcal{N}(0, {S_f}^2 \cdot \sigma^2)$ is added into the query function f to satisfy (ϵ, δ)-DP, where the sensitivity S_f is the maximum distance of two adjacent datasets' outputs $|f(D_1) - f(D_2)|$. No matter for averaging the model updates in Alg. 1 or averaging their norms in Alg. 2, the sensitivity is $\frac{1}{M} c^t$.

According to the Sequential Composition Theorem [15] of DP, multiple applications of a DP algorithm still satisfy DP, and the overall algorithm's privacy budget is the sum of that of every single algorithm. Our algorithms can be divided into two parts, and each part uses Moments Account to accumulate the privacy budget. Since both parts satisfy DP and the privacy budget is ϵ_1 and ϵ_2 respectively, the overall algorithm satisfies DP and the privacy budget is $\epsilon = \epsilon_1 + \epsilon_2$. □

3. EXPERIMENTS

3.1. Experimental Setup

Datasets and DNN Architectures. We use two datasets in our experiments: CIFAR-10 [16] and EMNIST [17]. We split the training images of CIFAR-10 using Dirichlet distribution [18] to simulate Non-IID setting, which is realistic in FL, and use the lightweight ResNet18 [19] as the training model. In the case of EMNIST, we first access the EMNIST Digits split and then distribute the training dataset randomly to all clients. We use a five-layer CNN with two convolution layers, one max-pooling layer, and two dense layers to train on this dataset. In both settings, all clients are allocated a subset of the training dataset without overlap, and share the whole test dataset as their test dataset.

Backdoor Tasks. We conduct a single-pixel attack on EMNIST and a semantic backdoor attack on CIFAR-10. In the single-pixel attack, the attacker changes the bottom-right pixel of all its training images from black to white, and modifies the labels of images to '0'. We modify a fraction of the test images in the same way to measure the success rate of the single-pixel attack, i.e., the proportion of backdoored images classified as '0' to all backdoored images whose true labels are not '0', and use the rest of the test data to observe the main task accuracy. In the semantic backdoor attack, the backdoor feature is cars painted in red. We first distribute images without the backdoor feature to all clients by Dirichlet distribution. Then we distribute images with the backdoor feature randomly to malicious clients and they will classify these images as birds. Examples of backdoored images are depicted in Fig. 1. For EMNIST, the number of test samples used for success rate measurement is 2000, and for CIFAR-10, it is 132.

(a) EMNIST (b) CIFAR-10

Fig. 1. Examples of backdoored images in single-pixel attack (a) and semantic backdoor attack (b).

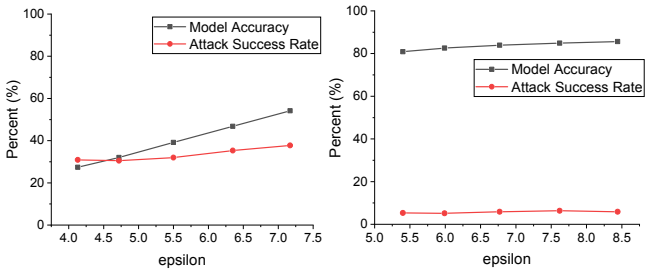
Federated Learning Setting. By default, we have $N = 100$ clients, with $P = 20$ poisoned clients. In each round, we select $M = 20$ clients, among which P_m clients are selected from the poisoned clients and the rest are selected from honest clients. Both poisoned and honest clients are selected randomly from two nonoverlapping client sets. The number of local epoch (E) is set to 5 for EMNIST, while $E = 2$ for CIFAR-10. For both tasks, the batch size is 20, and the learning rate is 0.04. FL runs for 300 rounds and the decay coefficient γ is 0.99. All results are averaged over 5 runs.

3.2. Performance Evaluation of CND

We implement DP with CND and the original DP (CDP) to defend against model poisoning attack on CIFAR-10 dataset ($\delta = 10^{-5}$), in which 8 poisoned clients are selected in rounds 250, 270 and 290, respectively. The poisoned clients take "train-and-scale" method [13] during their rounds and the experimental results are depicted in Fig. 2.

As Fig. 2 shows, there is a trade-off between defense efficiency and data utility, with respect to the magnitude of the noise added through DP. For CDP, adding more noise helps to reduce the success rate of backdoor attacks, but it also greatly brings down the model accuracy. In the case of DP with CND, the increase of perturbation does not lead to an obvious decrease in the model accuracy. Although CND spends a small extra privacy budget (1.27), it achieves at least 20% higher accuracy than CDP, under the same privacy budget.

Moreover, CND significantly reduces the attack success rate. We can learn that in Alg. 1, if the clipping threshold is



(a) CDP (b) DP with CND

Fig. 2. Model accuracy and attack success rate of CDP and DP with CND on CIFAR-10. The initial clip norm is 0.05. $\epsilon = 4.13, 4.72, 5.50, 6.35, 7.17$ in CDP.

a constant, the norm of model update will always be smaller than the threshold from a specific round, which means the update will no longer be clipped. However, in the later stage of the training, the accuracy of the main task tends to converge and the backdoor task is mainly learned. Therefore, the update of malicious clients is generally greater than that of honest clients. If not clipped, the malicious gradient is uploaded to the server in its entirety, playing a dominant role in the aggregated gradient. By contrast, CND enables the model update to be continuously clipped and limits the influence of the malicious gradient. This explains why CND can reduce the attack success rate.

3.3. Comparison

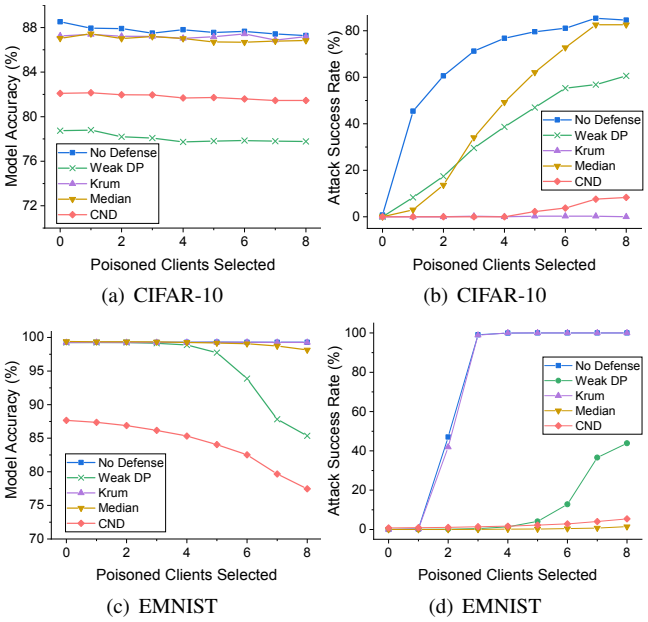
We compare our method with three state-of-the-art defensive mechanisms against backdoor attacks: Weak DP [11], Krum [20], and Median [21].

Weak DP [11]. The server clips the updates and adds slight Gaussian noise to the aggregated update. We set the clipping threshold to 0.1 and 0.2 for tasks on CIFAR-10 and EMNIST, and set σ to 0.005 and 0.001, respectively.

Krum [20]. For each client's update Δ_i, the server computes the Euclidean distances between it and k closest clients' updates to it, and then selects the update with the smallest sum of distances as the global update. Supposing at most $C = 8$ poisoned clients are selected, then k is $M - C - 2 = 10$.

Median [21]. For each of the model update's parameter $\Delta_{i,j}$, the server sorts the parameter $\Delta_{i,j}$ of all selected clients' updates and takes the median of them as the global update's parameter. When there is an even number of updates, it takes the mean of the middle two parameters.

As shown in Fig. 3, Weak DP only provides a poor defense against two kinds of attacks. Krum picks the gradient with the most "partners" and is therefore vulnerable to collusion. Hence, it fails against single-pixel attacks where attackers modify all their training samples in the same way. On the contrary, selecting the median is not affected by malicious parameters that appear at one end of the normal range, but



(a) CIFAR-10 (b) CIFAR-10

(c) EMNIST (d) EMNIST

Fig. 3. Results of different defensive mechanisms against two kinds of backdoor attacks. (a), (c): Model accuracy. (b), (d): Attack success rate. The initial clip norm of CND is 0.05 and 0.1, respectively. $\epsilon = 4.72 + 1.27 = 5.99$.

it fails when the malicious parameters are scattered over the whole interval. Hence, Median is disabled in semantic backdoor attacks, where attackers are assigned backdoored images randomly and generate more diverge updates.

Importantly, the defender cannot assume the attack strategy employed by an adversary in real life, so neither approach can be applied. Instead, our method provides a general defense against both backdoor attacks, dropping the success rate close to zero. Although DP has been applied to defend against backdoor attacks in early works, they either lost great data utility or set a tiny noise multiplier and obtained a limited defensive effect. As compared above, due to our unique design of CND, we solve the dilemma of choosing perturbation level, and achieve promising results.

4. CONCLUSION

In this paper, we proposed a new defense method based on DP, to solve the problem of losing model utility when defending against backdoor attacks, which decreases the clipping threshold of model updates in the training process. By adaptively setting the appropriate thresholds, our algorithm reduced the noise injection and eliminated the impact of malicious updates. Experiments of CND showed that our method could not only improve the main task accuracy, but also further reduce the success rate of backdoor attacks compared to the original DP. In comparison with the state-of-the-art defensive mechanisms, CND outperforms them by a large margin.

5. REFERENCES

[1] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*.

[3] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy, SP 2019*.

[4] Jacob Steinhardt, Pang Wei Koh, and Percy Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*.

[5] Shiqi Shen, Shruti Tople, and Prateek Saxena, "Auror: defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016*.

[6] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Proceedings*.

[7] Cynthia Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Proceedings, Part II*.

[8] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016*.

[9] Robin C. Geyer, Tassilo Klein, and Moin Nabi, "Differentially private federated learning: A client level perspective," *CoRR*, vol. abs/1712.07557, 2017.

[10] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang, "Learning differentially private recurrent language models," in *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*.

[11] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan, "Can you really backdoor federated learning?," *CoRR*, vol. abs/1911.07963, 2019.

[12] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro, "Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy," *CoRR*, vol. abs/2009.03561, 2020.

[13] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, "How to backdoor federated learning," in *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*.

[14] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang, "Trojaning attack on neural networks," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018*.

[15] Frank McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009*.

[16] Alex Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik, "EMNIST: extending MNIST to handwritten letters," in *2017 International Joint Conference on Neural Networks, IJCNN 2017*.

[18] T. Minka, "Estimating a dirichlet distribution," in *Technical report*. 2000, MIT.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*.

[20] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*.

[21] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.