Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 9, No. 2, 2024

Differential Private Defense Against Backdoor Attacks in Federated Learning

Lu Miao *, Weibo Li, Jia Zhao, Xin Zhou, Yao Wu

State Grid Shanxi Electric Power Company, Taiyuan, Shanxi, China
* **Corresponding author**: Lu Miao (Email: miaolu@ustc.edu)

Abstract: Federated learning has been applied in a wide variety of applications, in which clients upload their local updates instead of providing their datasets to jointly train a global model. However, the training process of federated learning is vulnerable to adversarial attacks (e.g., backdoor attack) in presence of malicious clients. Previous works showed that differential privacy (DP) can be used to defend against backdoor attacks, at the cost of vastly losing model utility. In this work, we study two kinds of backdoor attacks and propose a method based on differential privacy, called Clip Norm Decay (CND) to defend against them, which maintains utility when defending against backdoor attacks with DP. CND decreases the clipping threshold of model updates through the whole training process to reduce the injected noise. Empirical results show that CND can substantially enhance the accuracy of the main task. In particular, CND 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. Additional experiments about property inference attack indicate that CND also maintains utility when defending against privacy attacks and does not weaken the privacy preservation of DP.

Keywords: Adversarial Machine Learning; Backdoor Attack; Differential Privacy; Federated Learning.

1. Introduction

Nowadays, more and more machine learning systems need to collect a large amount of personal information, such as mobile keyboard prediction [1], chat bot [2]. With the rising of people's safety awareness, privacy preservation has become a hot issue. Since personal data contains users' private information and cannot be collected directly, traditional centralized machine learning will no longer be applicable. In order to protect privacy, the concept of federated learning (FL) [3, 4] 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. The server collects model updates from various clients and aggregates them to generate a new global model. In this way, all clients train the global model collaboratively without having to provide their own data.

However, there are new threats to federated learning due to its distributed nature. The process of local training is not under the control of the server and therefore is highly vulnerable to attacks. An adversary may tamper with the local dataset to inject a backdoor into the model. Backdoors [5] are hidden patterns learned by a DNN model, misleading the model to output wrong labels when inferring samples with backdoor features (aka trigger inputs). Backdoor features can be existing features in training data, or patterns designed by the attackers. It is difficult for the server to determine whether there is a malicious client. If the attack is successful, it will lead to the misclassification of specific samples, causing serious consequences. For instance, once the image classifier of a self-driving automobile is attacked, it may output a wrong instruction when capturing a carefully crafted picture, which is very dangerous in real life.

Traditional backdoor detection methods either assume an IID setting [6, 7], which is not realistic for FL, or require the defender to access training data or the final model [8], violating the privacy principle of FL. In the context of FL, differential privacy (DP) [9] was originally used to address the threat of privacy leakage, providing different levels of privacy guarantee, i.e., record-level [10, 11] and user-level [12, 13]. Recently, there has been some research on the defense against backdoor attacks with DP. Their motivation is to eliminate the difference between the malicious gradient and the normal gradient by perturbing the gradient uploaded in federated learning. Weak DP [14] could reduce the success rate of backdoor attacks to a relatively low level. CDP and LDP [15] could further reduce the success rate by injecting much more noise, whereas at the cost of decreasing the main task accuracy heavily.

Our research aims to solve the problem in defense methods with DP. We first focus on two kinds of backdoor attacks in FL: single-pixel attack and semantic backdoor attack, and study several factors related to backdoor attacks empirically. We find that the success rate of backdoor attacks rises as the number of malicious clients increases. When the number of backdoored samples is fixed, the attack with denser backdoored samples has a higher success rate. Then, we discover the relation between backdoor attacks and model overfitting. That is, the process of malicious clients training their local models involves overfitting. Experimental results show that when learning parameters are set to suppress overfitting, the success rate of backdoor attacks decreases to some extent.

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. We design a method for CND to set a new threshold according to the

collected model updates. 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. In addition, to verify that reducing clipping threshold does not introduce privacy risk, we implement CND to defend against property inference attack proposed by previous work. The result indicates that our method provides the same privacy preservation capability as the original DP and enhances the main task accuracy significantly. This means CND can be applied in federated learning to defend against both security and privacy attacks.

To summarize, our contributions can be described as follows:

- We show several factors that contribute to the success of backdoor attacks in federated learning, e.g., the degree of backdoored samples' concentration. Our experiment also substantiates that overfitting occurs in the process of backdoor attack.
- We propose CND to solve the long-standing problem of losing model utility when applying DP in federated learning. CND enhances the main task accuracy to a quite high level when defending against backdoor attacks. Besides, compared to previous work, our method reduces the success rate of the attack to a lower level.
- We empirically verify that CND also works in the defense against privacy threats, greatly improving the main task accuracy. Moreover, our method does not introduce privacy risk and provides the same capability of privacy preservation as the original DP.

2. Preliminary and Threat Model

2.1. Differential Privacy

Differential Privacy provides precise privacy guarantee to a dataset by perturbing the query results of it. After a randomized algorithm adding calibrated noise to the output of the dataset, an adversary cannot distinguish whether a single data record is included in the dataset or not.

Definition 1 ((ϵ,δ)-differential privacy [9]). A randomized mechanism $M : D \rightarrow R$ 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

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

Here, $\epsilon > 0$ and $\delta \in [0,1]$ control the strength of the privacy guarantee. The privacy budget ϵ measures privacy loss, and a small ϵ indicates that deleting any record from a dataset does not change the probability that the algorithm outputs the same result significantly. The parameter δ is the probability that ϵ-differential privacy does not hold, which is a small number. Hence, a lower (ϵ,δ) means a stronger privacy guarantee. The randomized mechanism determines the amount of additive noise according to the sensitivity of the query function.

Theorem 1 (Sequential Composition [16]). Let randomized mechanism $M_1 : D \rightarrow R_1$ provides (ϵ_1,δ_1)-differential privacy, and $M_2 : D \rightarrow R_2$ provides (ϵ_2,δ_2)-differential privacy. Then their combination, defined to be $M_{1,2} = (M_1, M_2)$, provides $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-differential privacy.

Sequential Composition Theorem gives a way to calculate the privacy budget of multiple queries on the same dataset. However, the privacy budget calculated by this theorem can be loose. A tighter method called Moments Accountant was proposed by Abadi [10], which defines privacy loss as a random variable dependent on the added noise, then bounds the moments of the privacy loss at each step, and computes the cumulative privacy budget of the algorithm.

2.2. Threat Model

In our threat model, an adversary controls a subset of clients, called malicious clients. We assume the aggregator is honest and the threat model of the adversary is as follows:

Adversary's Goal. The adversary attempts to corrupt the global model, making the model misclassify the samples with particular features (backdoor features) into a wrong label assigned by itself. 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.

Adversary's Knowledge and Capability. 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. This assumption is similar to the white-box attacks. To study the factors associated with traditional backdoor attacks, in Section 3, we assume the adversary can only modify the training data of poisoned clients, and cannot manipulate the training process, like data poisoning attack [17]. In Section 5, to verify the effectiveness of our defense method against the most advanced backdoor attack, 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 [18], which is a strong assumption of the adversary's capability.

3. Backdoor Attacks in FL

3.1. Experimental Setup

Datasets and DNN Architectures. We use two datasets in our experiments: CIFAR-10 [19] and EMNIST [20]. CIFAR-10 consists of 60,000 color images in 10 classes, e.g., airplane, automobile, and bird, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images. We split the training images using Dirichlet distribution [21] to simulate Non-IID setting, which is realistic in FL. We use the lightweight ResNet18 [22] as the training model. EMNIST is a set of handwritten character digits derived from the NIST Special Database 19, with the same image size and dataset structure as MNIST. When split by 'Digits', there are 240,000 training images and 40,000 test images in 10 classes. 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.

(a) EMNIST (b) CIFAR-10

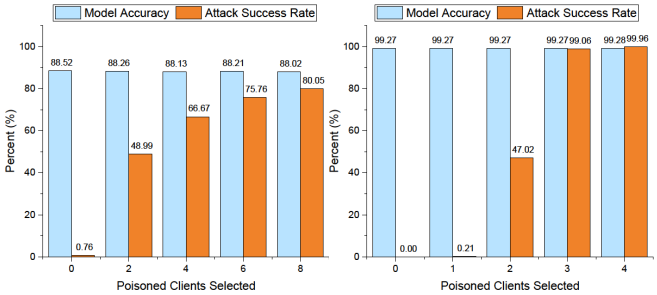
Fig 1. Examples of backdoored images in single-pixel attack (a) and semantic backdoor attack (b)

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 them to '0'. We modify a fraction of the test images in the same way to measure the success rate of the singlepixel 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. This kind of attack does not need to modify images. We first distribute images without backdoor feature to all clients by Dirichlet distribution. Then we distribute images with backdoor feature randomly to malicious clients and they will classify these images as birds. In the testing phase, we measure the success rate of the attack with backdoored images and record the main task accuracy with the other images. 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.

Federated Learning Setting. By default, we have N = 100 clients, with P = 20 malicious clients, or poisoned clients. In each round, we select M = 20 clients, among which Pm clients are selected from poisoned clients, which is a constant in a single experiment, 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. All results are averaged over 5 runs.

3.2. Analysis of Attack Results

We first study the variation trend of model accuracy and backdoor success rate with the number of poisoned clients. Intuitively, as the number of attackers increases, the success rate of the backdoor attack will rise dramatically. The experimental results on both datasets confirmed this conjecture, as Fig. 2 shows. The increasing poisoned samples facilitate the learning of backdoor tasks, and the success rate exceeds 80% with less than half of the poisoned clients. The backdoor task on EMNIST needs fewer poisoned clients to achieve a high accuracy than that on CIFAR-10, and part of the reason is that the poisoned clients backdoor all their training samples in the single-pixel attack, while in the semantic backdoor attack, the backdoored samples only make up a small proportion in their datasets. Besides, the accuracy of the model almost does not decline as the success rate rises, which implies that the redundancy of model feature space allows the model to learn backdoor features while maintaining the knowledge about the main task.



(a)CIFAR-10 (b) EMNIST

Fig 2. Model accuracy and attack success rate with the number of poisoned clients per round (the number of selected clients per round is 20 for both tasks)

Table 1. Model accuracy and backdoor success rate of two experiment settings on CIFAR-10

Setting	Pm	Poisoning Rate	Acc. (%)	Succ. (%)
1	8	0.5	88.50	54.24
2	4	1.0	88.13	66.67

Next, we attempt to figure out under what setting the success rate is higher when the number of poisoned samples is fixed. We define the poisoning rate as the proportion of backdoor samples that participate in the training. Then, we design two settings: in setting-1, Pm is 8 and the poisoning rate is 0.5; in setting-2, they are 4 and 1.0, respectively. That is, the density of the poisoned samples in setting-2 is twice as large as setting-1. Results in Table 1 show that, when the number of poisoned samples trained per round is fixed, the setting with denser backdoored samples achieves a higher success rate.

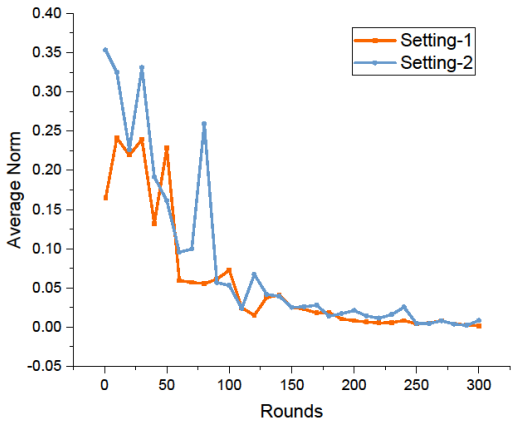


Fig 3. Average norm of poisoned clients' model updates on each round's first batch in semantic backdoor attacks (the total numbers of poisoned samples in both experiments are the same)

We calculate the average norm of the model updates on poisoned clients' first batch, as shown in Fig. 3, and find that the clients with denser backdoored samples have larger model updates through the training process. It can be speculated that, the model update trained with more backdoored samples has a closer orientation to malicious model and larger norm, hence the aggregated model update is closer to malicious model. In experiments below, we keep the poisoning rate as 1.0 by default.

3.3. Effect of Overfitting

During the experiments above, we discover that there is some relation between backdoor attack and model overfitting. We draw the accuracy and the success rate of backdoor attack during 200 rounds in the single-pixel attack, as shown in Fig. 4. The increase of the success rate is obviously behind the increase of accuracy. After the accuracy has converged, the success rate begins to rise gradually. The possible reason is that in early rounds, attacker's updates are hidden among those of other clients, and the global model mainly learns features of the main task. After the learning of the main task is finished, honest clients' updates become small, and the global model begins to learn the feature of the backdoor task. This means that the global model has overfitted.

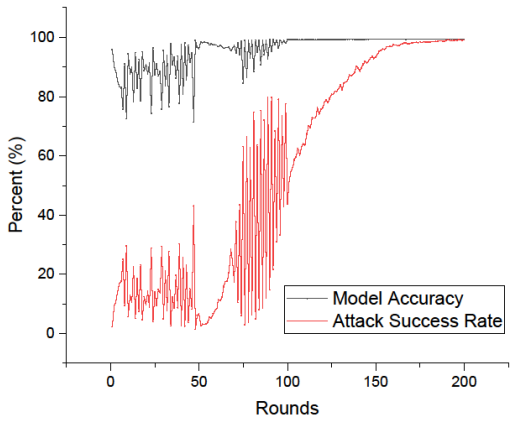
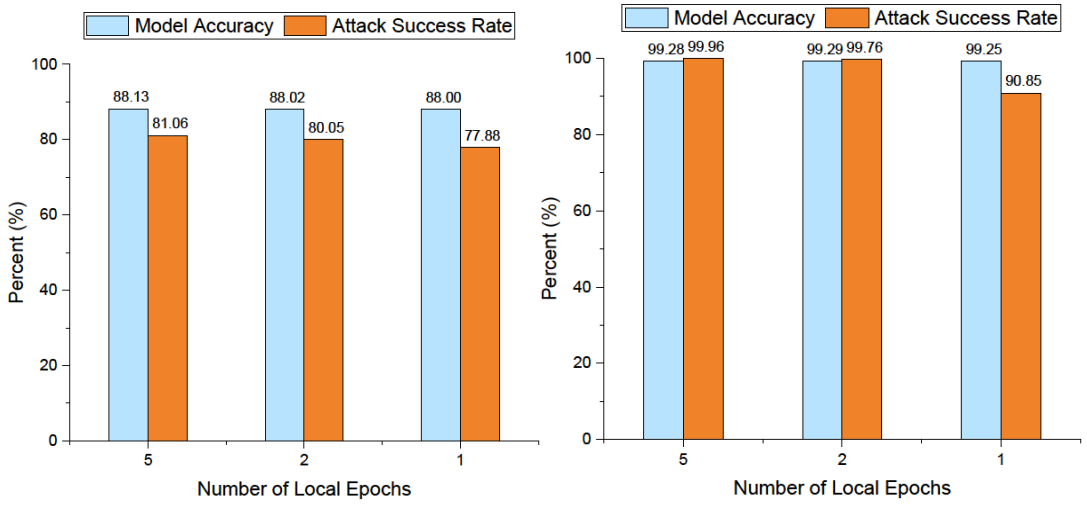


Fig 4. Trends of model accuracy and attack success rate during 200 rounds in the single-pixel attack (20 clients selected per round and 4 of them are poisoned)



(a) CIFAR-10 (b) EMNIST

Fig 5. Variation of model accuracy and attack success rate when changing the number of local epochs (for CIFAR-10, 8 poisoned clients are selected per round and for EMNIST, the number is 4)

In order to verify our speculation, we change the parameters of the training process to alleviate overfitting, and compare the accuracy as well as the success rate. First, we change the number of local epochs from 5 to 2 and 1, which reduces the training times of every client. Fig. 5 indicates that in both backdoor attacks, the success rate decreases to some extent (attacks on EMNIST decrease more) and the accuracy almost remains the same.

Next, we introduce L_2-regularization, setting a series of weight decays, as listed in Table 2. In the single-pixel attack, the success rate drops sharply to almost zero when increasing the weight decay. However, in the semantic backdoor attack, the success rate rises slightly, rather than going down. These distinct results are due to the different degrees of overfitting in the two tasks. In the single-pixel attack, the backdoor feature is a white pixel and all training samples of poisoned clients are labeled the same, causing their local models to learn a single feature. However, the backdoor feature in the semantic backdoor attack is the existence of any kinds of red cars, and the training data are divided into ten classes even in poisoned clients. A more balanced data distribution and a more complicated backdoor task make the model less prone to overfitting in the semantic backdoor attack, so the success rate is less affected by methods alleviating overfitting.

Table 2. Variation of model accuracy and attack success rate with respect to weight decay (the number of local epochs is 5 for EMNIST and 2 for CIFAR-10; the numbers of poisoned clients selected per round are 4 and 8 respectively)

	Weight Decay	0	1e-4	5e-4	1e-3	5e-3
EMNIST	**Acc. (%)**	99.28	99.32	99.10	98.31	97.02
	Succ. (%)	99.96	99.85	67.87	8.17	2.63
CIFAR-10	**Acc. (%)**	88.02	88.13	88.15	88.27	87.44
	Succ. (%)	80.05	80.00	81.06	81.36	84.24

To sum up, the redundancy of model feature space is a basic condition for backdoor attacks, which allows the model to learn backdoor features and maintain the accuracy of the main task simultaneously. In addition, backdoor attacks are related to overfitting. More specifically, it can be considered that overfitting occurs in the process of learning backdoor features when poisoned clients training their local models. The strength of the relation depends on the degree of overfitting. The higher the degree of overfitting, the stronger the relation becomes, and the more effective the mitigation of overfitting is against the backdoor attack.

4. Differential Privacy with Clip norm decay

4.1. Algorithm Description

We have discussed the effect of overfitting on backdoor attacks in Subsec. 3.3 and found that mitigating overfitting can reduce the success rate of backdoor attacks, to some extent. However, it is unrealistic to rely solely on mitigating overfitting to defend against backdoor attacks. For one thing, the degree of defense achieved by reducing local epochs is limited. Even training only one local epoch in a round, backdoor attacks can still have a high success rate. For another, the effect of regularization on backdoor attacks is uncertain. In the case of extreme overfitting, regularization could greatly reduce the success rate of the attack, but in other circumstances it might be ineffective. In reality, the defender cannot assume the specific way of the attack and therefore cannot set the optimal training parameters.

Next, we study defenses against backdoor attacks with DP. Our algorithm is based on user-level DP proposed by McMahan [13], which clips the model update computed at each batch of the data, and then perturbs the aggregated update at the server. The result of clipping at each batch is better than that of clipping at each round [12]. Because a round contains multiple epochs and an epoch contains multiple batches, the cumulative update in a round will be very large, and the clipping threshold should also be set very large in order not to affect the model accuracy. If clipping at each batch, the threshold could be a small value, hence injecting less noise. Previous work [15] showed that user-level DP could defend against backdoor attacks. However, this method still leads to a great loss of the main task accuracy.

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 MA(δ, M, N) ;
  for each round t = 0, 1, 2, ... **do**
    if ϵ < MA.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)
    σ = z / M
    $\theta^{t+1} \leftarrow \theta^t + \sum_i \Delta_i^{t+1} / M + \mathcal{N}(0, (c^t \cdot \sigma)^2)$
    MA.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**
      $\theta \leftarrow \theta - \eta \nabla L(\theta, b)$
  $\Delta \leftarrow \theta - \theta^t$
  $\theta \leftarrow \theta^t + \Delta \cdot \min(1, c^t / \|\Delta\|_2)$
  return $\theta - \theta^t$

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 [10] to compute the privacy budget spent at the beginning of each round, and accumulate the privacy loss after each round.

Algorithm 2 Computing new threshold by CND.

Function New Threshold(c^t, γ , t)
  $c^{t+1} \leftarrow \gamma \cdot c^t$
  if t < 10 or t = 50, 100, ... **then**
    $c \leftarrow \sum_i \|\Delta_i^{t+1}\|_2 / M + \mathcal{N}(0, (c^t \cdot \sigma)^2)$
    MA.accumulate_spent_privacy(z)
    if c < c^{t+1} **then**
      $c^{t+1} \leftarrow$ c
  return c^{t+1}

Our intuition is based on the fact that, as the number of rounds increases, the norm of model update gradually decreases, as shown in Fig. 3. 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. Since we do not change σ in Gaussian mechanism, our algorithm can obtain the same privacy guarantee as the original DP.

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.

4.2. Theoretical Analysis

In our method, the perturbing process is conducted at the server 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 algorithm satisfy DP.

Theorem 2. DP with CND satisfies (ϵ,δ)-differential privacy.

Proof. In Gaussian mechanism, noise with the normal

distribution 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 c^t/M. According to the Sequential Composition Theorem 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 algorithm 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 $\epsilon = \epsilon_1 + \epsilon_2$. Similarly, the overall $\delta = \delta_1 + \delta_2$.

5. Experiments

5.1. Performance Evaluation of CND

In the following semantic backdoor attack, we enhance the capability of the adversary by assuming that it can modify the training data of the poisoned clients and manipulate the local training process and training parameters to improve the learning of the backdoor feature. The attacker adopts the model replacement method [18], launching model poisoning attacks in rounds 250, 270 and 290, and set the learning rate as 0.04 and the number of local epochs as 50. Fig. 6 shows the experimental results with different numbers of poisoned clients.

By comparing with the result of data poisoning attacks (Fig. 2), it can be seen that the model poisoning attack has stronger power. For example, when the number of poisoned clients selected in each round is 2, the success rate of data poisoning attack is 48.99%. While, the success rate of the model poisoning attack can reach 60.6% even if only two poisoned clients launch attacks in three rounds.

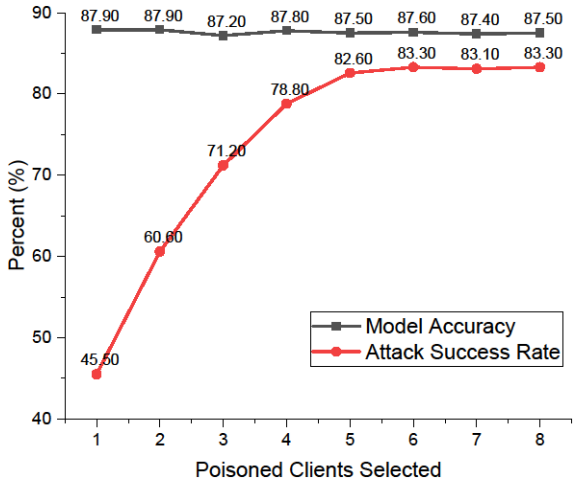
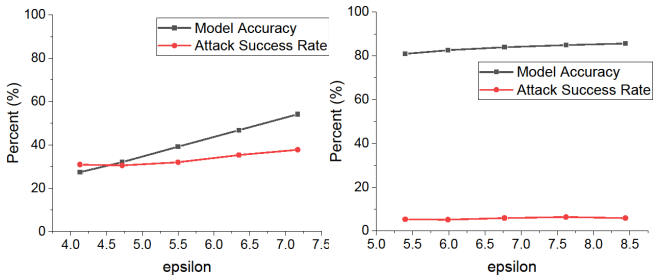


Fig 6. Model accuracy and attack success rate in the model poisoning attack on CIFAR-10 without defense

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 'trainand scale' method [18] during their rounds and the experimental results are depicted in Fig. 7.



(a)CDP (b)DP with CND

Fig 7. Model accuracy and attack success rate of CDP and DP with CND on CIFAR-10 ($c^0 = 0.05$. $\epsilon = 4.13, 4.72, 5.50, 6.35, 7.17$ in CDP)

As Fig. 7 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. From Fig. 3, we can learn that, if the clipping threshold in Alg. 1 is 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 (Fig. 4). Therefore, the update of poisoned 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.

5.2. Comparison

In this subsection, we compare our method with state-of-the-art defensive mechanisms against backdoor attacks. We have shown that our method outperforms CDP. LDP solution [15], which is based on DP-SGD, achieves similar results as CDP does, losing much model utility. And the poisoned clients can quit LDP, by skipping the process of clipping and perturbing their updates. Next, we conduct the comparison with other defenses: Norm Bounding [14], Weak DP [14], Krum [23], and Median [24].

Norm Bounding [14]. The server clips clients' model updates that exceed a threshold. After attempting a wide range of values, we set the thresholds to 0.1 and 0.2 for tasks on CIFAR-10 and EMNIST, respectively.

Weak DP [14]. The server clips the updates and adds slight Gaussian noise to the aggregated update. Based on Norm Bounding, we explore several values of σ of Gaussian noise, and set it to 0.005 for CIFAR-10 and 0.001 for EMNIST.

Krum [23]. 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.

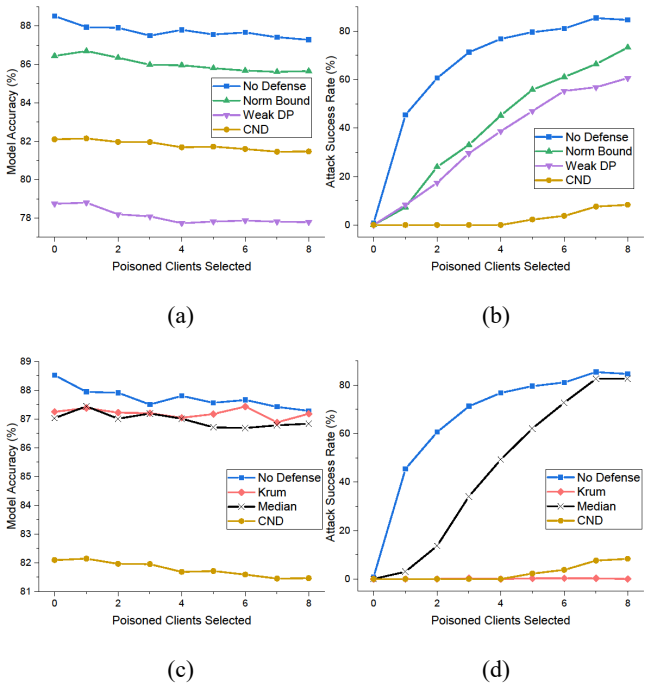


Fig 8. Results of different defense mechanisms against semantic backdoor attacks, (a) (c): Model accuracy, (b) (d): Attack success rate ($c^0 = 0.05$. $\epsilon = 4.72 + 1.27 = 5.99$, $\delta = 10^{-5}$)

Median [24]. For each 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.

Including DP with CND, we draw the experimental results of the five defense mechanisms in Figs. 8 and 9, with different numbers of poisoned clients selected per round. Norm Bounding provides a poor defense against two kinds of attacks. On this basis, Weak DP can lower the success rate only by a tiny amount.

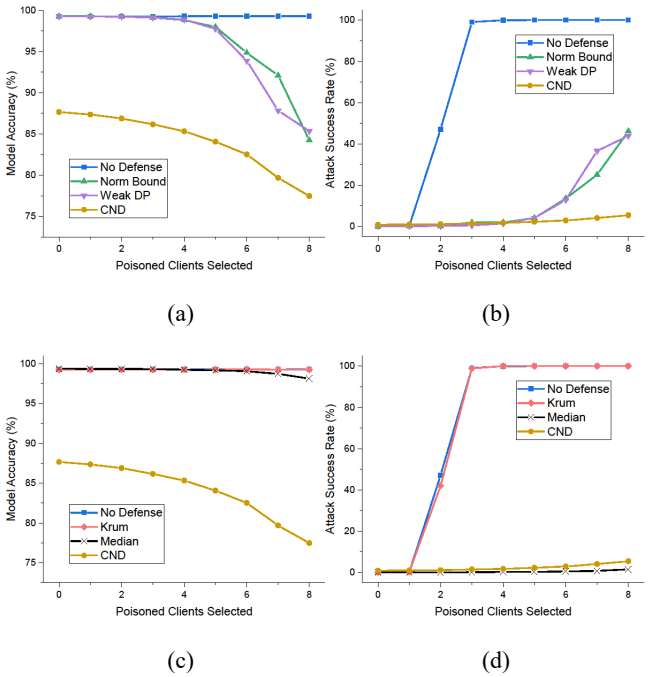


Fig 9. Results of different defense mechanisms against single-pixel attacks, (a) (c): Model accuracy, (b) (d): Attack success rate ($c^0 = 0.1$. $\epsilon = 4.72 + 1.27 = 5.99$, $\delta = 10^{-5}$)

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 and consequently have similar update direction. On the contrary, selecting the median is not affected by malicious parameters that appear at one end of the normal range, but 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. Different from Krum and Median that are designed for Byzantine attacks, our method is not restricted to some specific attack and 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.

5.3. Defending Against Property Inference Attack

We have demonstrated that CND has better performance than the original DP when defending against backdoor attacks. In what follows, we are concerned about whether CND would expand the privacy loss and damage the privacy preservation of DP. In theory, reducing the clipping threshold does not change the privacy guarantee, and our method provides an equivalent capability of the privacy preservation as the original DP. We herein verify this conclusion through experiments about property inference attack [25]. Property inference attack is a kind of privacy attack whose goal is to reveal the properties of data owners. The attacker in property inference attack [25] is a malicious client who aims to calculate the probability that a sample with a certain property is used in the global update. It calculates the aggregated update of clients other than itself as the test sample.

Table 3. Main task accuracy and AUC of the property inference attack in cases of different numbers of clients, averaged on 3 runs ($c^0 = 0.03$, $\delta = 10^{-5}$, ϵ of CND = 6.35 + 1.27 = 7.62)

Clients	No Defense		DP ($\epsilon = 8.0$)		CND($\epsilon = 7.62$)	
	Acc.(%)	AUC	Acc.(%)	AUC	Acc.(%)	AUC
2	91.67	0.81	34.04	0.53	52.30	0.50
3	92.30	0.78	41.40	0.49	65.76	0.49
4	92.40	0.74	58.32	0.53	65.72	0.50
5	92.03	0.71	62.11	0.50	69.44	0.51

We use the Labeled Faces in the Wild (LFW) dataset which contains more than 13,000 images of faces for around 5,800 individuals with property labels (e.g., gender, race, age, and hair color), collected from the web. We use the same CNN architecture as [25], including three spatial convolution layers with 32, 64, and 128 filters and maxpooling layers, followed by two dense layers of size 256 and 2, respectively. The main task is gender classification and the inference task is over race. The performance of the attack is evaluated by Area Under the Curve (AUC). All clients are selected per round (i.e., N = M) and FL runs for 300 rounds. The number of local epochs is 10. The data are equally distributed to clients and only the

attacker and the victim have data with the property. In our experiments, we use Alg. 1 to defend against the attack and compare the results of DP with CND and the original DP. We also record the results of no defense (without clipping and perturbing).

Experimental results are listed in Table 3. From the table, we can see that, the AUC of the attack goes down with the increase of the number of clients when no defense is applied. This is because, as the number of clients increases, the victim's update is aggregated with more updates, and inferring the property of victim's samples becomes harder. Thus, it can be speculated that federated learning with a large number of clients is not vulnerable to the property inference attack. Moreover, user-level DP effectively defends against the attack, reducing the AUC to around 0.5. The reason is that user-level DP perturbs clients' updates at each round so that the adversary cannot tell whether a specific client has joined in training, which is consistent with the definition of DP.

Nevertheless, the noise introduced into the global update by DP makes the update deviate from correct direction, resulting in a drastic decline on the main task accuracy. We also observe that federated learning with more clients is less affected by noise perturbation of DP (62.11% vs. 34.04%). For a binary classifier, an accuracy less than 50% does not make sense. In contrast, CND significantly improves the accuracy of the model to an acceptable level (e.g., from 41.40% to 65.76%), which is similar to the results of defending against backdoor attacks. In all cases, our method achieves an equivalent AUC as DP does, which implies that CND does not introduce privacy risk, and our method provides the same level of privacy preservation as the original DP.

6. Conclusion

In this paper, we studied backdoor attacks in FL and proposed a new defense based on DP. We found several factors that promote the success rate of the attack, e.g., the increase and the concentration of backdoored samples in poisoned clients' datasets. We also discovered the relation between backdoor attacks and model overfitting, and empirically verified that suppressing overfitting helps to defend against backdoor attacks. In particular, we proposed CND to solve the problem of losing model utility when defending against backdoor attacks with DP, which decreases the clipping threshold of model updates in 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 accuracy of the main task, 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 on two datasets, CND outperforms them by a large margin. Moreover, we implemented CND to defend against property inference attack and validated that our method also improves the main task accuracy in the defense against privacy attack and does not risk the privacy preservation of DP.

For future work, we plan to apply DP with CND to more privacy threats to obtain extensive results about the method. After that, we will evaluate the prospect of using this approach to protect federated learning from security and privacy attacks simultaneously.

References

[1] Hard, A., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. CoRR abs/1811.03604 (2018).

[2] Schlesinger, A., O'Hara, K.P., Taylor, A.S.: Let's talk about race: Identity, chat bots, and AI. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018.

[3] Konecný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. CoRR abs/1610.05492 (2016).

[4] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017.

[5] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy, SP 2019.

[6] Steinhardt, J., Koh, P.W., Liang, P.: Certified defenses for data poisoning attacks. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017.

[7] Shen, S., Tople, S., Saxena, P.: Auror: defending against poisoning attacks in collaborative deep learning systems. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016.

[8] Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018.

[9] Dwork, C.: Differential privacy. In: Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006.

[10] Abadi, M., Chu, A., Goodfellow, I.J., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[11] Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I.J., Talwar, K.: Semisupervised knowledge transfer for deep learning from private training data. In: 5th International Conference on Learning Representations, ICLR 2017.

[12] Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. CoRR abs/1712.07557 (2017).

[13] McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning differentially private recurrent language models. In: 6th International Conference on Learning Representations, ICLR 2018.

[14] Sun, Z., Kairouz, P., Suresh, A.T., McMahan, H.B.: Can you really backdoor federated learning? CoRR abs/1911.07963 (2019).

[15] Naseri, M., Hayes, J., Cristofaro, E.D.: Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. CoRR abs/2009.03561 (2020).

[16] Vadhan, S.: The Complexity of Differential Privacy, pp. 347–450. Springer, Cham (2017).

[17] Liu, Y., Ma, S., Aafer, Y., Lee, W., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: 25th Annual Network and Distributed System Security Symposium, NDSS 2018.

[18] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020.

[19] Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009).

[20] Cohen, G., Afshar, S., Tapson, J., Schaik, A.: EMNIST: extending MNIST to handwritten letters. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017.

[21] Minka, T.: Estimating a dirichlet distribution. In: Technical Report. MIT, (2000).

[22] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016.

[23] Blanchard, P., Mhamdi, E.M.E., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017.

[24] Yin, D., Chen, Y., Ramchandran, K., Bartlett, P.L.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018.

[25] Melis, L., Song, C., Cristofaro, E.D., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE Symposium on Security and Privacy, SP 2019.