Sybil Attacks on Differential Privacy based Federated Learning

A Dissertation Presented in Fulfillment of the Requirements for the Degree of Master of Research

Yupeng Jiang

Supervisor: Dr Xi Zheng
Associate Supervisor: Dr Yipeng Zhou

MACQUARIE
University
SYDNEY · AUSTRALIA

Department of Computing
Faculty of Science
Macquarie University, NSW 2109, Australia

Submitted October 2020

Declaration

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Signed: ....

Date: 24 October 2020

Acknowledgements

I would like to express my sincere appreciation to my supervisor Dr Xi Zheng and my associate supervisor Dr Yipeng Zhou, for their comprehensive guidance and great encouragement for the completion of this thesis. I also would like to thank my peers in the research group for their discussion on related research questions and preliminary study. In addition, many thanks to the staff in the Department of Computing for their administrative help. Finally, special thanks to my wife Zisu for her warm support in my life. She takes a good care of our lovely son Bradley.

Abstract

Machine learning and deep learning techniques have become prevailing in artificial intelligence. The rise of applications in autonomous vehicles, healthcare, and finance introduce practical challenges against various attacks towards these systems. Learning from unbalanced and non-IID (not independently and identically distributed) data while preserving privacy, federated learning is proposed to train global models on distributed devices. In federated learning, each device as a client owns a private training dataset that is invisible to other parties, which protects data privacy and data security. However, the loose federation of participating devices in this decentralized approach could bring potential security threats between the communications among these nodes. The state-of-the-art privacy-preserving technique in the context of federated learning is user-level differential privacy. It substantially reduces information disclosure about decentralized datasets rather than individual records. Despite this, such a mechanism is vulnerable to some specific model poisoning attacks such as sybil attacks. A malicious adversary could create multiple fake clients or collude compromised devices in sybil attacks to mount direct model updates manipulation. Recent works on novel defense against model poisoning attacks are difficult to detect sybil attacks when differential privacy mechanism is utilized, as it masks clients' model updates with perturbation. This thesis is based on the scope of federated learning settings where user-level differential privacy is deployed. There are three contributions in this work as follows.

The first contribution of the work is to implement sybil attacks on differential privacy based federated learning architectures and show the impact of model poisoning attacks on model convergence. The attack intensity depends on the number

of sybil clients and the noise levels of each sybil reflected by the local privacy budget ϵ of differential privacy.

The second contribution of the work is to propose a method to detect and defend sybil attacks for a differential privacy based federated learning setting. The key insight is that the poisoned model parameters from sybil clients can be identified by their induced higher loss values of prediction on the global model than those from honest clients in each iteration round of training. When the central server aggregates the clients' models, the model updates obtained from sybil clients may induce higher cost in the global model than those from honest clients, which affects the convergence of the global model.

The third contribution of the work is to apply our attacks to two recent Byzantine-resilient aggregation defense mechanisms, called Krum and Trimmed Mean. Our evaluation results on the MNIST and CIFAR-10 datasets demonstrate that our proposed sybil attacks increase the training loss of the global model tremendously on various state-of-the-art defense mechanisms. We also conduct an empirical study which shows that our defense approach effectively mitigates the impact of our model poisoning attacks on model convergence.

Contents

List of Publications

This thesis is largely based on a submission to *IEEE Transactions on Industrial Informatics* (Q1, Impact Factor: 9.112).

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Federated Learning

Machine learning and deep learning techniques have become prevailing in artificial intelligence. Along with this rise of application in autonomous vehicles, healthcare, and finance, data privacy and security are the major concerns during machine learning training and test procedure. Learning from unbalanced and non-IID (not independently and identically distributed) data while preserving privacy, federated learning methods are proposed to train global models on distributed devices [19, 20, 25, 28, 37, 38, 39]. In federated learning, each device as a client has private data of training that is inaccessible to other clients and the server, which protects data privacy and data security.

Federated learning can be classified into cross-device and cross-silo settings. Cross-device federated learning may contain up to 10^{10} mobile or IoT devices to solve the optimization problem collaboratively, while in cross-silo setting it typically involves 2 - 100 distributed organizations or data centres [17]. Such a distribution scale introduces practical challenges on reliability against various attacks towards the system.

In federated learning, learning models are shared globally. When training a global model, each client computes an update to the server by performing a local iterative algorithm to achieve the learning objective. Stochastic gradient descent (SGD) algorithm is commonly used towards a local minimum in the practice of federated learning [1, 29]. The central server is responsible for aggregating all clients' updates with an aggregation rule in the current round. Suppose we have K clients in total. An aggregation rule can be formally expressed by $w = \mathscr{A}(w_1, w_2, \cdots, w_K)$. This procedure requires iterative communication rounds during the entire learning process to train a high-quality model. However, this framework is vulnerable to model poisoning attacks [2, 5, 13], and it becomes even worse to some specific model poisoning attacks, called sybil attack [9]. In one study, Fung et al. [14] demonstrate that a deep learning network model in federated learning can be easily subverted by using the sybil attack. In such attacks, the clients' model updates are tampered with a backdoor into the learned model, even if a small fraction of the client devices are compromised [3, 5].

1.2 Differential Privacy

Differential privacy is a privacy preservation technique to quantify and limit leaking sensitive data [10, 11, 12]. It masks user responses with perturbations when submitting queries to the statistical database, which aims to maximize the utility of accuracy, meanwhile minimize reveal of individual records. Specifically, the query results for adjacent datasets are close enough such that no information can be inferred from their difference. The notion of adjacent datasets is referred to as they differ by only one record. Formally, a query algorithm $\mathscr{M}$ satisfies (ϵ, δ)-differential privacy if for all adjacent datasets $\mathscr{D}$ and $\mathscr{D}'$:

$$P(\mathscr{M}(\mathscr{D}) \in S) \leq e^{\epsilon} P(\mathscr{M}(\mathscr{D}') \in S) + \delta \tag{1.1}$$

where ϵ is the privacy budget, δ is the confidence, and S denotes the output space of the query. There have several popular differential privacy mechanisms of perturbation, including Laplace, Gaussian, and exponential mechanism. In this thesis, we use Laplace mechanism to distort the output of the user.

In Laplace mechanism, the perturbation to output data is achieved via Laplace distributed additive noises with probability density function

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{1.2}$$

where μ is a location parameter and b is a scale parameter. A Laplace mechanism $\mathscr{L}(\mu, \frac{\Delta f}{\epsilon})$ satisfies (ϵ, μ)-differential privacy where $\Delta f = \max_{\mathscr{D},\mathscr{D}'} \|f(\mathscr{D}) - f(\mathscr{D}')\|_1$ if $\mathscr{D}$ and $\mathscr{D}'$ are adjacent datasets. In this Laplace differential privacy mechanism, for any $\epsilon > 0$, the scale of additive noise is increased when ϵ is reduced, corresponding to an increased level of privacy protection. In the next section, we introduce the application of differential privacy in the context of federated learning.

1.3 Differential Privacy based Federated Learning

In this thesis, we name our federated learning setting *differential privacy based federated learning*. As a defense technique in cryptography, differential privacy is utilized in federated learning model to protect data confidentiality between the communications among server and multiple clients. By adding a certain distribution of random noise on each client's update locally or on the aggregated global model, user-level differential privacy [30] can be achieved when training a global model. Similar to (1.1), if $\mathscr{D}'$ distinguishes from $\mathscr{D}$ by all the records of a single client, the algorithm $\mathscr{M}$ satisfies (ϵ, δ) - user-level differential privacy. Accordingly, we use Laplace distributed additive noises with the following probability density function:

$$f(w) = \frac{1}{2b} \exp\left(-\frac{\|w\|_1}{b}\right) \tag{1.3}$$

where w represents model parameters shared globally. This Laplace mechanism $\mathscr{L}(0, \frac{\Delta f}{\epsilon})$ satisfies $(\epsilon, 0)$ - user-level differential privacy if $\mathscr{D}$ and $\mathscr{D}'$ are two user-adjacent datasets. Moreover, in this setting, any specific user's data will not influence the behaviours of trained model no matter it is used for training or not during the learning process.

However, there have some weaknesses in these methods. One of which is that differential privacy requires that the scale of additive noises has to match the scale of parameters in the model updates to preserve utility. As discussed in the last section, when the privacy protection level is increased, the scale of additive noises is also increased, which will reduce the convergence rate of the global model or even lead to divergence. Therefore, it needs a comparatively larger number of clients or larger value of privacy budget compared to that in the central setting [17]. We evaluate the negative impacts of differential privacy on model convergence in Chapter 6. Inspired by this limitation of user-level differential privacy which leaves more space for the attack, we introduce our sybil attacks that aim to manipulate the updated parameters inside local models such that the aggregated model in the server has a high cost of the prediction indiscriminately for training examples, which makes the global model converges slowly, or even leads to divergence.

1.4 Challenges

We perform the study on differential privacy based sybil attacks in federated learning settings. A key challenge for the attacker is that how the compromised model updates from sybil clients can obscure the aggregation defense rules in the uncompromised server to deviate the global model from its original prediction. To address the challenge, we apply different strategies to carefully craft local model updates of sybil clients according to different aggregation rules by manipulating different noise levels reflected by the local privacy budget ϵ of differential privacy. Our goal of attack is to introduce higher cost in the global model when the server aggregates all the clients' updates, including our crafted parameters in each training round compared with the original one. Our attack intuition is the accumulated high cost during the learning process may slow down the convergence of the global model significantly.

We replicate two recent defense techniques, called Krum [7] and Trimmed Mean [36], as our targeted aggregation rules. Our evaluation results on MNIST [23] and CIFAR-10 [21] show that our proposed sybil attacks increase the training loss of the global model tremendously in the presence of these Byzantine-resilient defenses throughout rounds of training.

Existing defenses against model poisoning attacks replace the mean aggregation rule in the central server with a Byzantine-resilient algorithm as the robust aggregator [31, 32]. However, these defenses do not take into account the scenarios where differential privacy is applied in the federated learning model. Whereas differential privacy prevents data leakage, federated learning models are still susceptible to model poisoning attacks, especially to sybil attacks. To address this challenge, we propose our defense method to defend against our sybil attacks on differential privacy based federated learning settings. Our proposed defense excludes those client updates inducing high loss values of prediction on the global model where the cost is evaluated based on the loss report from each client. Empirical results show that our proposed defense mechanism effectively mitigates the impacts of our sybil attacks on the convergence of the global model.

1.5 Contributions

There are three main contributions in this thesis as follows:

- We implement sybil attacks on differential privacy based federated learning architectures and show their impacts on model convergence. The attack intensity depends on the number of sybil clients and the noise levels of each sybil reflected by the local privacy budget ϵ of differential privacy on the local model updates of these sybil clients.

- We propose a method to detect and defend our sybil attacks based on the prediction cost reported from each client. The key insight is that the poisoned model parameters from sybil clients can be identified by their induced high loss values of prediction on the global model. When the server aggregates the clients' models, the model updates obtained from sybil clients may induce higher cost in the global model than those from honest clients, which affects the convergence of the global model.

- We apply our attacks to two recent aggregation defense mechanisms, called Krum [7] and Trimmed Mean [36], that are resilient to arbitrary adversarial behaviour. Our evaluation results on the MNIST and CIFAR-10 datasets demonstrate that our proposed sybil attacks increase the training loss of the global model tremendously. We also conduct an empirical study to illustrate that our proposed defense method effectively defends against our sybil attacks.

1.6 Roadmap of the Thesis

This paper is organized as follows. Chapter 2 discusses the state-of-the-art adversarial attacks and defenses methods on federated learning. In Chapter 3, we introduce our attacks on how to compromise Krum and Trimmed Mean. We describe our defense solution in Chapter 5. Chapter 6 analyzes the evaluation results using two public datasets. We finally conclude our work and close with future research direction in Chapter 7.

Chapter 2

Literature Review

Federated learning provides a machine learning setting where the optimization problem can be solved collaboratively, rather than traditional centralized model training. However, this framework is vulnerable to increasing threats from various attacks, even with the presence of differential privacy technique in machine learning tasks. In this chapter, we survey the literature on state-of-the-art attack and defense mechanisms on federated learning.

- Section 2.1 introduces the attack paradigms and their limitations in federated learning.
- Section 2.2 presents state-of-the-art defense mechanisms against malicious attacks that target federated learning.

In this thesis, we focus on the sybil attacks in the context of differential privacy based federated learning. We discuss about sybil attacks and security issues in federated learning in Section 2.1. The literature related to differential privacy are summarised in Section 2.2.2.

2.1 Attacks on Federated Learning

The distributed nature of federated learning architecture introduces increasing threats and attack surfaces. There have been considerable recent works that have proposed various attacks towards the federated learning systems. We review these

Table 2.1: Attacks on Federated Learning

Attack Vector	Goals	Methods	Literature
Model Poisoning	Untargeted	Craft local model updates directly by optimization based methods	[4, 13, 17, 22]
	Targeted	The corrupted model updates are learned from the auxiliary data	[5]
Data Poisoning	Untargeted	Train global model using crafted training data	[6]
	Targeted	Label-flipping or add perturbations to the original data	[8, 14, 18, 26]

attack modes in terms of the attack vector and goals of the attack. The existing attack methods are summarised in Table 2.1.

2.1.1 Model Poisoning Attack

The most distinction between federated learning and centralized machine learning is that federated learning trains a model collaboratively across distributed client devices. Thus, it opens up new attack surfaces such that the adversary is able to manipulate the model updates sent back to the server [4]. This class of adversarial attacks is known as model poisoning attacks. Since the corrupted model updates can be arbitrary, model poisoning attacks are generally viewed as the most powerful and worst-case attacks, which is also referred to as Byzantine attacks [22]. Currently, Byzantine attacks mainly aim to degrade model performance or even make the global model unusable. Based on the goal of the adversary, these Byzantine attacks can be further classified as *untargeted attacks* [17]. For the characteristic of model poisoning attacks, some recent works have shown that the Byzantine-resilient defenses are susceptible to model poisoning attacks in federated learning.

In work [13], Fang et al. showed that their proposed attacks could effectively degrade the trained model in the Byzantine-robust federated learning setting. Their method is to craft local model updates from compromised client devices during the training phase by formulating their attacks as optimization problems. By solving their well-designed optimization problem, the global model deviates from its intended direction without attacks in each training iteration of federated learning. The experiment results show that the error rates of the trained models are increased considerably under their attacks to several Byzantine-resilient methods. This work demonstrated how untargeted model poisoning attacks impact the performance of federated learning methods. However, the limitation is that the attacker has to know about the aggregation rule in the server entirely. Otherwise, the attacks would be much less effective.

Another category of model poisoning attack is *targeted attacks*. The aim is that the trained model is modified in desired behaviour for the adversary, such as misclassification on some specific tasks. Bhagoji et al. [5] revealed that the learned global model could be poisoned to the misclassify targeted objective while preserving the classification accuracy of the trained model. Moreover, it needs only a small portion of client devices to be compromised for their targeted model poisoning attacks. However, it assumes that the server uses the accuracy on validation data to detect anomalous updates. In the training process, validation data are not accurate enough compared to training data due to the stochastic gradient descent algorithm, as we discussed in Section 1.1. Therefore, the effect of this kind of attacks is limited without the appropriate assumptions.

Our proposed sybil attacks are one of the worst-case model poisoning attacks. In sybil attacks, an adversary controls some number of clients to send arbitrary values. Furthermore, while recent Byzantine model poisoning attacks studies focus on Byzantine-robust federated learning, our attacks consider differential privacy preserved federated learning.

2.1.2 Data Poisoning Attack

In the literature, data poisoning attacks are explored comprehensively. One fact is that it is more natural to tamper client data for an adversary in the compromised client devices rather than manipulate local model updates in the training process of federated learning. Another possible reason would be that data poisoning attacks somehow induce model poisoning attacks eventually. Since local model parameters are updated by performing an iterative algorithm over the training data for the optimization problem, any violation of client data must eventually result in some alteration of model updates being sent to the server. However, it is still uncertain about the quantity relation between two attack paradigms where additional research is needed [17].

In data poisoning attacks, the adversary tampers the training dataset of clients by replacing labels or adding perturbations to the original data [18, 26]. Recent work by Fung et al. [14] illustrates the vulnerability of federated learning to data poisoning attacks. They train a model of classification in the federated learning setting, where the ten-digit MNIST dataset [23] is distributed to ten clients. Each client is assigned a partition of MNIST dataset with one single digit. Their sybil attacks simulate several fake clients and collude them to train the classifier on the poisoned dataset where only contains images of digit '1' with the incorrect class label '7'. As a result, the learnt model classifies 96.2% images of digit '1' as the class of digit '7' incorrectly, meanwhile maintaining the accuracy rate of classification on other digits at a high level of 88.8%. Such characteristic reflected from their empirical results is strongly related to targeted attacks. In fact, targeted attacks are referred to as backdoor attacks [8], in which the performance of the global model on specific tasks is influenced through manipulating client data. In the work of targeted attacks [5] we have reviewed in Section 2.1.1 for model poisoning attacks, the final weights update sent back by the malicious client is actually learned from the auxiliary data. Therefore, we can see that although the model poisoning attacks are more powerful, it is enormously important to investigate data poisoning attacks for well understanding of the relation between them. Although some past work has explored untargeted data poisoning that reduces the accuracy of the global model notably using crafted training data [6], research directions on targeted data poisoning attacks are dominant.

2.2 Defenses on Federated Learning

Table 2.2 shows the current defense mechanisms against model poisoning and data poisoning attacks.

Table 2.2: Defenses on Federated Learning

Defense Type	Scope	Mechanisms	Literature
Byzantine-resilient	Model Poisoning	Replace aggregation rule with a robust aggregator	[7, 36]
	Data Poisoning	Select clients' updates by ranking-based preference	[35]
Privacy-preserving	Model Poisoning	Train global model with differential privacy	[33]
	Data Poisoning	A subset of clients' updates are randomized	[15, 27]

2.2.1 Byzantine-resilient Defenses

A common method to aggregate the local models is using the mean aggregation rule [29]. However, this model averaging is susceptible to adversarial attacks and hardly provides privacy guarantees. A number of works have explored Byzantine-resilient defense mechanisms for federated learning. Specifically, recent works propose various robust aggregation rules against both untargeted and targeted attacks.

As the popular aggregation defense mechanisms, Krum [7] and Trimmed Mean

[36] are proposed to be robust under untargeted adversarial settings. These methods replace the mean aggregation rule in the central server with a Byzantine-resilient algorithm as the robust aggregator. However, these mechanisms work under appropriate assumptions that provably asymptotic on the number of the client. In our sybil attacks, an adversary is capable of manipulating model updates from a large number of client devices, which significantly influences the performance of the global model, even when these defense aggregation rules are present.

In relation to data poisoning attacks, they can be viewed as special cases of model poisoning attacks. The reason is that compromised training data will induce anomaly in clients' model updates. Therefore, Byzantine-resilient defenses against model poisoning attacks may also work for data poisoning attacks [35].

It is noteworthy that any proposed robust defense has to guarantee the convergence of global model when using gradient descent algorithm on client device.

2.2.2 Robustness of Differential Privacy

Concerning about data privacy, user-level differential privacy [30] is leveraged in the context of federated learning. A number of works have shown that the use of differential privacy effectively defends against privacy disclosures on the scope of targeted model poisoning attacks [33], data poisoning attacks [15, 27], and attacks on adversarial examples [24]. In our study, we implement the first sybil attacks as untargeted model poisoning attacks and defenses on federated learning models with differential privacy applied. In [33], Sun et al. have explored an approach to eliminate the impacts of targeted attacks using differential privacy, while our sybil attacks are untargeted attacks that focus on differential privacy based federated learning.

Chapter 3

Problem Definition and Threat Model

This chapter describes the architecture of our differential privacy based federated learning settings, followed by characterizing the capabilities and goals of adversaries in sybil attacks.

In this thesis, we consider a standard federated learning context, in which there are K clients in total, each owning private training data and the number of c compromised clients at most from K clients. All the clients collaboratively train a classifier by solving the optimization problem

$$\min f(w) \qquad \text{where} \qquad f(w) = \sum_{k=1}^{K} f_k(w) \tag{3.1}$$

where $f_k(w)$ is the objective function for the local dataset on the kth client, and w denotes the parameters of the global model. Specifically, the procedure of federated learning in each round is as follows:

Step1. The central server sends the global model parameters w to each participating client.

Step2. Each participating client computes an update to the server by performing a local stochastic gradient descent (SGD) algorithm using the local dataset. The learning objective is defined in (3.1).

Step3. The server aggregates clients' models by a predetermined aggregation rule $w = \mathscr{A}(w_1, w_2, \cdots, w_K)$ where w_k denotes the parameters in the local model updates of each client.

We call our federated learning settings *differential privacy based federated learning*. In this architecture, the clients' model updates are masked with a user-level differential privacy perturbation as described in Section 1.3 in the form:

$$\widetilde{w}_k = w_k + \widetilde{n}_k \tag{3.2}$$

where w_k is the parameters in the local model updates from the kth client, and $\widetilde{n}_k$ is an additive noise to guarantee differential privacy. Formally, the clients' model update queries Q satisfy $(\epsilon, 0)$ - user-level differential privacy if for all client-adjacent datasets $\mathscr{D}$ and $\mathscr{D}'$:

$$P(Q(\mathscr{D}) \in S) \leq e^{\epsilon} P(Q(\mathscr{D}') \in S) \tag{3.3}$$

where ϵ is the privacy budget and S denotes the output space of the query.

The noise level of each client is reflected by the local privacy budget ϵ of differential privacy on the local model updates. In this thesis, we use Laplace distributed additive noises with the probability density function (1.3). In accordance with Theorem 1 in work [34], we calculate the scale parameter in (1.3) $b = 2D_{max}T/n_k\epsilon$ where D_{max} represents the maximum absolute value of the model update parameters in current communication round, T is the total number of communication rounds, and n_k is the number of examples in training dataset of the kth client. It is proved that the clients' model updates being sent back to the server that perturbed by noise $\widetilde{n}$ with this Laplace mechanism meet $(\epsilon, 0)$ - user-level differential privacy.

Our sybil attack is one of untargeted model poisoning attacks. In this attack model, an adversary can spoof up to the number of c clients and tamper model parameters before sending them back to the server during the training process, as the capability of Byzantine threat model discussed in Section 2.1.1. In this thesis, we assume that the c compromised clients are from a total of K clients and no more fake clients in the system for simplicity. Moreover, we assume that the adversary knows the aggregation rule $\mathscr{A}$ used by the server, as it is usually published for the

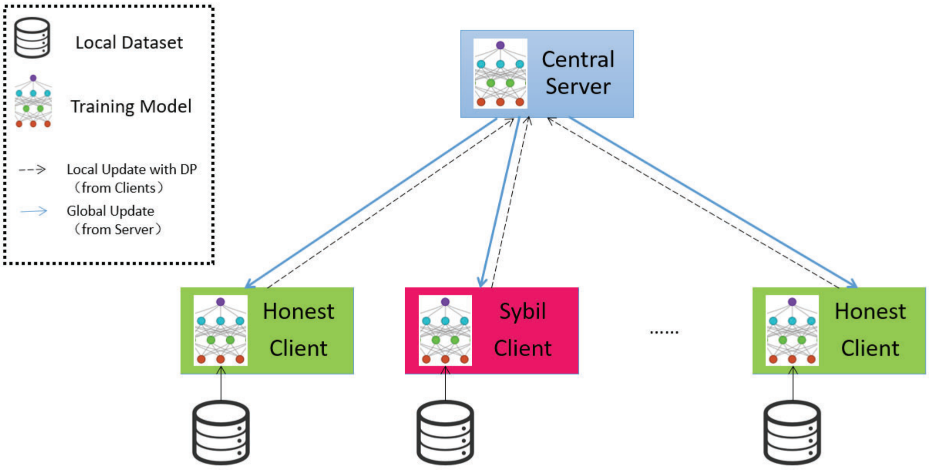


Figure 3.1: Differential privacy based federated learning with sybil attacks.

trust and transparency of the system [28]. The structure of the learning process in differential privacy based federated learning with sybil attacks is shown in Figure 3.1.

We consider the goal of an adversary is to slow down the convergence rates of the global model or even diverge the model in the training phase. In this thesis, we assume the loss function of models is smooth and strongly convex. Although the loss function for high-dimensional networks is usually non-convex, it can still achieve a local minimum using stochastic gradient descent (SGD) algorithm iteratively on each client when training a model in federated learning.

Under the assumption above, the lower bound of global convergence rate in federated learning settings can reach to $O(\frac{1}{T})$, where T denotes the total number of communication rounds to train a model [16]. As we discussed in Section 1.3, differential privacy with a small value of ϵ reduces the convergence rate of the global model. Given the lower bound of the model convergence rate, we propose a search algorithm to choose an optimal ϵ for differential privacy in the system. Specifically, we select one from the following values as the local privacy budget ϵ of clients: 0.1, 0.3, 0.5, 1.0, 2.0, 5.0, 8.0, 10.0. These typical values of ϵ have been evaluated in recent works for the trade-off investigation between privacy and utility of differential privacy [1, 34, 40]. We first initialize $\epsilon = 10.0$ for all clients and calculate the global convergence rate based on the average loss of prediction on the global model in 50 iterations of training. If it is greater than $O(\frac{1}{T})$, then we choose the next smaller value of ϵ from candidates and repeat this process until the global convergence rate is less than $O(\frac{1}{T})$. This process determines the optimal value of ϵ for honest clients, which guarantees differential privacy, meanwhile preserves the convergence rates of the federated learning model. To solve ϵ value on sybil clients, we will introduce our attack strategies according to different aggregation rules, respectively, in the next chapter.

Chapter 4

Our Attack

In the user-level differentially privacy-preserving federated learning setting, for any $\epsilon > 0$, the scale of additive noise over the client updates is increased when ϵ is reduced. We leverage this characteristic to introduce a larger variance on model updates from sybil clients using a smaller value of ϵ relative to it on honest clients, which will induce higher loss of prediction on the global model in each iteration round of training.

In this chapter, we introduce our sybil attack strategies for three aggregation rules in the central server of federated learning, including one widely used aggregator and two state-of-the-art defensive mechanisms.

4.1 Our Attack to FedAvg

One commonly used aggregation rule in federated learning is FederatedAveraging (FedAvg) [29]. In FedAvg algorithm, the global model in each communication round of training is the average of all clients' model parameters.

$$w_{t+1} = \frac{1}{K}\sum_{k=1}^{K} w_t^{(k)} \tag{4.1}$$

where $w_t^{(k)}$ represents the local parameters from the kth client in the current round t, and w_{t+1} is the aggregated global model for next training round. The user-level differential privacy on each client's model update is applied by:

$$\widetilde{w}_t^{(k)} = w_t^{(k)} + \widetilde{n}_t^{(k)} \tag{4.2}$$

where $\widetilde{n}_t^{(k)}$ is Laplace additive noise $\sim \mathscr{L}(0, \frac{\Delta f}{\epsilon})$ with the optimal ϵ we choose using our search algorithm. Based on this optimal ϵ, the differential privacy can be guaranteed while preserving the convergence rate of global model above the lower bound. From (4.1), it is easy to get the aggregation with differential privacy:

$$\widetilde{w}_{t+1} = \frac{1}{K}\left(\sum_{k=1}^{K} w_t^{(k)} + \sum_{k=1}^{K} \widetilde{n}_t^{(k)}\right) \tag{4.3}$$

As discussed in Section 2.2.1, FedAvg is vulnerable to adversarial attacks. We can increase additive noise $\widetilde{n}_t^{(k)}$ in (4.3) from sybil clients to achieve a large variance on $\widetilde{w}_{t+1}$ by reducing the local privacy budget ϵ on sybil clients.

We propose two strategies to attack FedAvg. One obvious method is to use any ϵ_s on sybil clients for $0 < \epsilon_s < \epsilon_h$ where ϵ_h is the optimal value we choose for honest clients using our search algorithm in Chapter 3. In this thesis, for simplicity, we assume that all sybil attackers use the same ϵ_s for attacks, and all honest clients use the same ϵ_h for differential privacy. The smaller value of ϵ_s corresponds to the stronger attack intensity, easier to be detected, however. To choose the value of ϵ_s from candidates, we evaluate this attack method using different ϵ_s and the different number of sybil attackers in Section 6.2. The other more stealthy method is using synchronous additive noises on these collusive sybil clients. In this method, the noises added to the model updates of sybil clients are from either the positive or negative part of Laplace distribution in phase. According to (4.3), even with a small magnitude of additive noises, the attack intensity will be amplified by the sum operation of FedAvg when the server aggregates all the clients' model updates. We evaluate two methods in our experiment, respectively.

4.2 Our Attack to Krum

Recent work [7] proposed Krum to increase the robustness of the aggregation rule against Byzantine attacks. The basic idea is that it selects one of the model updates from all the clients as the global model instead of using the mean of them. The selection criterion is based on the similarity concerning Euclidean distance between two clients' model updates. Specifically, suppose we have K clients in total and c sybil clients among them, it first calculates the Euclidean distance between each client's model update. Then for each model update, it computes the squared sum of the smallest $K-c-2$ Euclidean distances. Finally, the Krum algorithm selects the model update with the minimum squared sum as the global model. It has been proved that the global model can converge to a local minimum under Byzantine attacks when $c < \frac{K-2}{2}$ by using Krum. This literature also proposed Multi-Krum algorithm as a variant version of Krum to speed up the convergence when training a global model. In Multi-Krum, it selects m clients' model updates with the smallest squared sum instead of one in Krum, then uses the mean of selected model updates as the global model. We can see that when $m = 1$, Multi-Krum is same as Krum, and when $m = K$, Multi-Krum is the FedAvg aggregation rule.

As Krum selects one model update from K clients as the global model for the next communication round, our idea is that this model is from one of c sybil clients. The goal is this selected model deviates the global model from its intended converge direction before attacks. The key challenge of the attack is that each crafted local model with added random noises will induce large Euclidean distance to the models from honest clients. As a result, Krum can easily exclude our crafted local models in such aggregation rule. To address this challenge, we let model updates from sybil clients maintaining the same to achieve a zero Euclidean distance between each of them. Then we carefully adjust ϵ_s in these sybil clients such that their Euclidean distances to honest models are comparable with those among honest clients. This collusion of sybil clients ensures our crafted model update to be selected by Krum. In implementing our attack, we evaluate both Krum and Multi-Krum on different training models, respectively, to maximize the attack impacts in experiments.

4.3 Our Attack to Trimmed Mean

Another aggregation rule, Trimmed Mean [36], considers element-wise algorithm in the model updates. Similar to Krum, Trimmed Mean requires an explicit number of compromised clients. As we assumed those c sybil clients as mentioned above, it removes the largest and smallest c elements in model parameters among all clients' updates. After that, it uses the average of the remaining elements as the corresponding parameter in the global model. In Trimmed Mean, the variance of model parameters in clients' update is constrained to a benign magnitude, which mitigates the impacts of Byzantine attacks. The authors also proved that the global model converges when $c < \frac{K}{2}$ and the statistical error rates achieves $O(\frac{c}{K\sqrt{n}} + \frac{1}{\sqrt{Kn}})$ for strongly convex loss functions, where n is the number of examples in training dataset of each client. We notice that when $c = 0$, i.e. there is no attack, the Trimmed Mean algorithm is equivalent to FedAvg.

To slow down the convergence of the global model in our attack, we craft c compromised local models based on the intended gradient of each element in current training round. Specifically, when one parameter in global model intends to increase upon the previous iteration if there is no attack, we add negative random noise with Laplace distribution onto this element of the corresponding location in each compromised client, such that this parameter with additive noise from each compromised client is smaller than the majority of the corresponding model parameter from the honest clients. As a result, the mean of the remaining $K-2c$ elements according to the Trimmed Mean algorithm is tending to decrease upon the previous iteration. Otherwise, if one parameter in the global model intends to decrease upon the previous iteration, we add positive random noises with Laplace distribution on each compromised client in the same way. In our experiments, we evaluate the values of ϵ_s to nominate the most effective attacks.

Chapter 5

Our Defense

We design a method to detect and defend our sybil attacks on differential privacy based federated learning setting. Compared with state-of-the-art defense mechanisms such as Krum and Trimmed Mean, our proposed algorithm does not require the exact quantity of compromised clients. The experiment results show that our defense method effectively mitigates the impacts of untargeted model poisoning attack on model convergence.

The key insight is that the poisoned model updates from sybil clients can be identified by their induced high loss of prediction on the global model. Technically, the cost of a network is defined as a function $f_i(w) = \ell(x_i, y_i; w)$ which takes model parameters w as its input and maps the loss of output on examples (x_i, y_i) where x_i is input and y_i is label. In our federated learning settings, sybil clients will contribute model updates that appear larger loss values than those from honest clients to affect the convergence of the global model.

Our approach keeps monitoring the convergence rate of the global model throughout all training rounds on the server-side. As we discussed in Chapter 3, the optimal convergence rate can reach to $O(\frac{1}{T})$ for a smooth and strongly convex loss function where T denotes the number of communication rounds. This algorithm evaluates the convergence rate from round 2 by comparing the model loss decrease rate to a pre-determined threshold to detect sybil attacks. This threshold reflects defense intensity. For most of machine learning or deep learning models, the convergence rate is usually between $O(\frac{1}{\sqrt{T}})$ to $O(\frac{1}{T})$ [16]. In this thesis, we set the threshold to $0.8(\frac{1}{t-1} - \frac{1}{t})$ or $0.8(\frac{1}{\sqrt{t-1}} - \frac{1}{\sqrt{t}})$ depending on the loss function in the global model. Note that a ratio of 0.8 is used to tolerate non-malicious failures from unreliable clients.

To locate sybil attackers from participating clients, we use binary search in the vector of client devices. For each half of client devices, the central server sends loss report request to a random fraction C of K clients with model parameters w averaged from model updates in the corresponding half of clients. After that, the loss values reported from selected clients l^k are averaged in the server. We keep searching sybil attackers in the half of client devices vector with larger mean of loss until three clients or 10% of total K clients remaining. Finally, we aggregate all clients' model parameters, excluding these remaining client updates after the binary search has finished for the next global training round. Although there could have some honest client updates sacrificed for a few rounds of communication, our defense does not influence the convergence of the global model. The details of our defense method are introduced in Algorithm 1.

In Algorithm 1, our proposed defense method keeps monitoring the convergence rate of the global model from iteration round 2 until the end of the training period, by comparing the rate to a pre-determined threshold of defense intensity. When this model loss decrease rate drops below the threshold in a certain round t, our method detects sybil attacks and launches the defense procedure. The binary search algorithm is used to locate sybil model updates among all the participating clients. The invariant in binary search is the key insight of our proposed defense method, which is that the average loss of clients containing sybil attackers is larger than that of the other half. Specifically, we split the model updates from all the clients and average the local model parameters of each half, w'_t and w''_t respectively. Then the server sends both global model parameters w'_t and w''_t to randomly selected set of clients S_t for the request of loss report. Each client $k \in S_t$ completes the training

Algorithm 1 Detection and defense

```
Require: Average training loss $l_1$ in round 1
Server executes:
  for round $t = 2, 3, \ldots$ do
    if $(\Delta l_t)/l_1 <$ threshold of defense intensity then
      // The binary search
      // The invariant: the average loss of clients containing sybil attackers is larger than that of the other half.
      $i, h = 1, K$
      while $h - i > \max(K/10, 2)$ do
        $m = \lfloor (i+h)/2 \rfloor$
        $w'_t = \frac{1}{m} \sum_{k=1}^{m} w_t^{(k)}$
        $w''_t = \frac{1}{K-m} \sum_{k=m+1}^{K} w_t^{(k)}$
        // Each client reports loss with model parameters $w'_t$ and $w''_t$
        $S_t =$ random set of $\max(C \cdot K, 1)$ clients
        for each client $k \in S_t$ do
          $l_t^{(k)\prime} = \text{ClientCost}(k, w'_t)$
          $l_t^{(k)\prime\prime} = \text{ClientCost}(k, w''_t)$
        end for
        $l'_t = \frac{1}{num(S_t)} \sum_{k=1}^{num(S_t)} l_t^{(k)\prime}$
        $l''_t = \frac{1}{num(S_t)} \sum_{k=1}^{num(S_t)} l_t^{(k)\prime\prime}$
        if $l'_t < l''_t$ then
          $i = m + 1$
        else
          $h = m$
        end if
      end while
      Exclude client updates ranging from $w_t^{(i)}$ to $w_t^{(h)}$, remaining $K'$ clients
    end if
    $w_{t+1} = \frac{1}{K'} \sum_{k=1}^{K'} w_t^{(k)}$
  end for

ClientCost($k, w$):
  batches $\leftarrow$ training data split into batches of size $B$
  for batch $b$ in batches do
    $l_b = \ell(w; b)$
  end for
  $l = \frac{1}{num(batches)} \sum_{b=1}^{num(batches)} l_b$
  return $l$ to server
```

task based on both global model parameters w_t' and w_t'' and returns loss value $l_t^{(k)\prime}$ and $l_t^{(k)\prime\prime}$ to the server respectively. The client uses the minibatches of size B on SGD algorithm in local. After all the clients in S_t report loss values $l_t^{(k)\prime}$ and $l_t^{(k)\prime\prime}$, the server calculates the mean of these loss values l_t' and l_t'' for each half client vector correspondingly. We keep searching sybil clients in the half of vector with larger loss value by comparing l_t' and l_t''. We repeat this process until there are 10% of total K clients or 3 clients left in the vector. Then we exclude these clients' model updates and average the remaining model updates as the global model w_{t+1} for the next training round. This concludes our proposed defense mechanism against sybil attacks for monitoring and detection implementation.

Chapter 6

Evaluation

6.1 Experiment Setup

In this thesis, our proposed attack and defense approaches are evaluated by CNN and MLP models on two datasets MNIST [23] and CIFAR-10 [21] respectively. The MNIST data are partitioned by non-IID, and CIFAR-10 data are IID. We implement a federated learning prototype of PyTorch based on [29]. The computer environment is Intel® Core$^{\text{TM}}$ i7-4770 CPU @ 3.40GHz processor, 16.0GB RAM and Windows 10 64-bit operating system.

The parameters for our differential privacy based federated learning settings by default are summarised in Table 6.1. The sybil attackers are randomly selected from 100 clients in each communication round. All additive noises are Laplace distributed with a corresponding privacy budget.

Table 6.1: Federated Learning Settings

Parameter	Description	Value
K	Number of clients	100
C	Fraction of clients	0.1
c	Number of compromised clients	20
T	Number of communication rounds	50
B	Local batch size	10
E	Number of local epochs	5
η	Learning rate	0.01
ϵ_h	Privacy budget of honest clients	8.0
ϵ_s	Privacy budget of sybil clients	0.3

To compare with our proposed attack methods, we implement Gaussian attack [13] as a benchmark. This attack injects random noises with Gaussian distribution into the local model updates from sybil clients. In our experiments, we set the mean of the distribution to 0 and its standard deviation to 0.3 for evaluation.

As shown in Table 6.1, we perform 50 rounds of training and 5 local epochs on each client in our experiments. The learning rates were tuned between 0.01 to 0.05 for the best performance.

6.2 Evaluation on Model Convergence

In this section, we evaluate the model convergence using different parameter settings in our differential privacy based federated learning architecture.

6.2.1 Impact of Differential Privacy

In our federated learning framework, all clients' model updates are preserved with user-level differential privacy. However, this method has negative impacts on model convergence, as we discussed in Section 1.3. Figure 6.1 shows the model convergence concerning the privacy budget on the MNIST dataset throughout 50 communication rounds. The performance of model convergence is similar when $\epsilon \geq 1$. The convergence rate starts to decrease when ϵ is reduced to 0.5. Furthermore, the global model diverges substantially when $\epsilon = 0.3$.

We notice that there is a significant difference between $\epsilon = 0.3$ and $\epsilon = 0.5$ in Figure 6.1. When $\epsilon = 0.5$, the training loss decreases throughout the communication rounds of training, which is close to the other training curves with larger values of ϵ. It shows that the scale of additive noises are comparable to the scale of parameters in the model updates of local clients. However, when we reduce ϵ to 0.3, the training loss stops decreasing in round 10, and starts to increase dramatically in the rest of training rounds. That means the scale of additive noises are too large, which

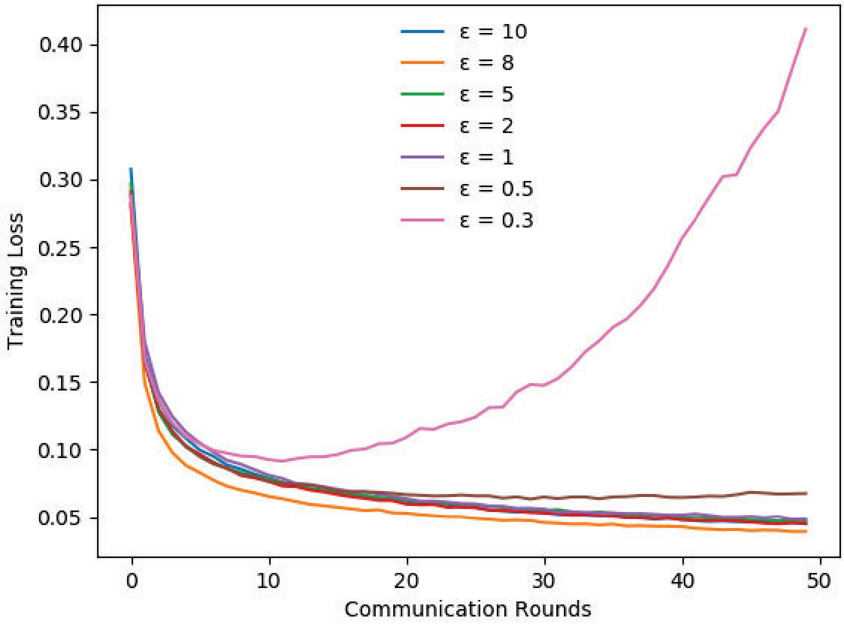


Figure 6.1: Training loss for different privacy budgets on MNIST.

significantly affects the aggregation of model parameters in the server. The huge gap appears when the value of ϵ is reduces by just 0.2. This reveals the issue on the robustness of differential privacy: when the privacy protection level is appropriate to the global model, even a little increase of protection level might affect the model convergence significantly.

We also evaluate these impacts on the CIFAR-10 dataset using MLP classifier, as shown in Figure 6.2. In both scenarios, it is evident that when we reduce the value of privacy budget ϵ on the clients, the convergence rate of the global model is decreased or the model even stops converging. Therefore, we need an appropriate value of ϵ for differential privacy while guaranteeing the convergence. In our experiment for evaluation on the attack, we set $\epsilon_h = 8.0$ for all honest clients.

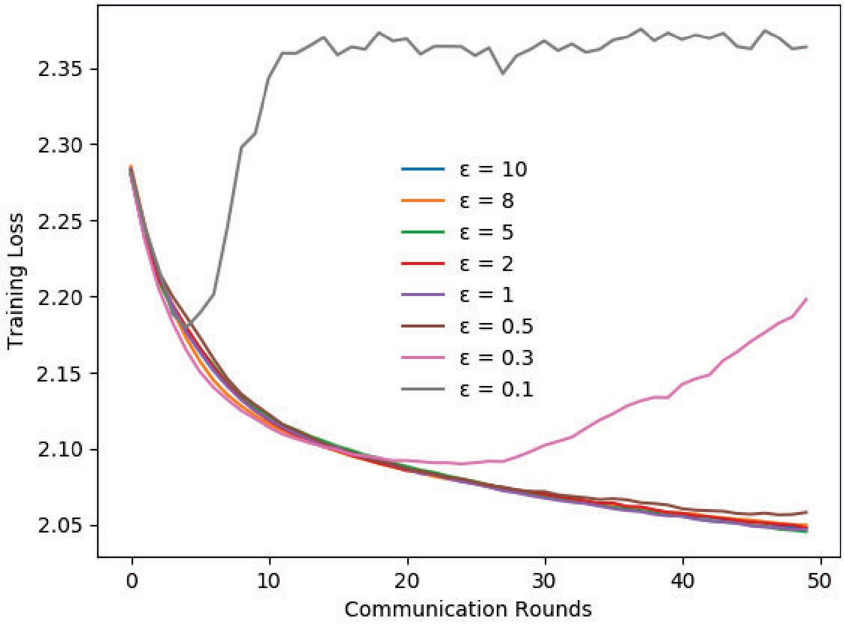


Figure 6.2: Training loss for different privacy budgets on CIFAR-10.

6.2.2 Impact of Compromised Clients Quantity

Intuitively, more compromised clients have more impacts on model convergence. We run the experiment with different fractions of 100 clients on the MNIST and CIFAR-10 datasets, respectively. As shown in Figure 6.3, the model convergence is getting slow when the ratio of compromised clients increases. However, from both Figure 6.3 and Figure 6.4, the impact of compromised clients quantity are not as significant as that concerning privacy budget. Although the percentage of compromised clients can be further increased, it is not practical in real-world federated learning settings.

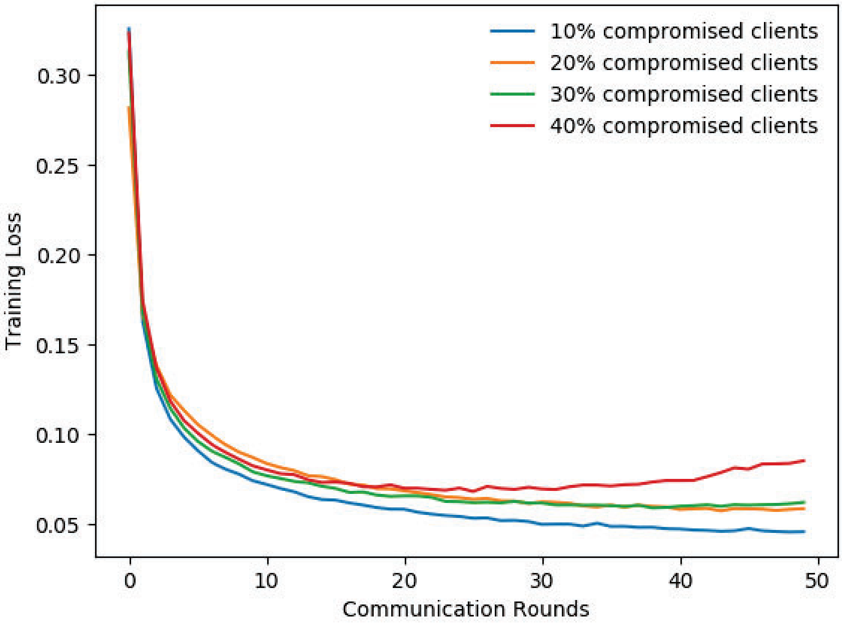


Figure 6.3: Model convergence with respect to compromised clients ratio on MNIST.

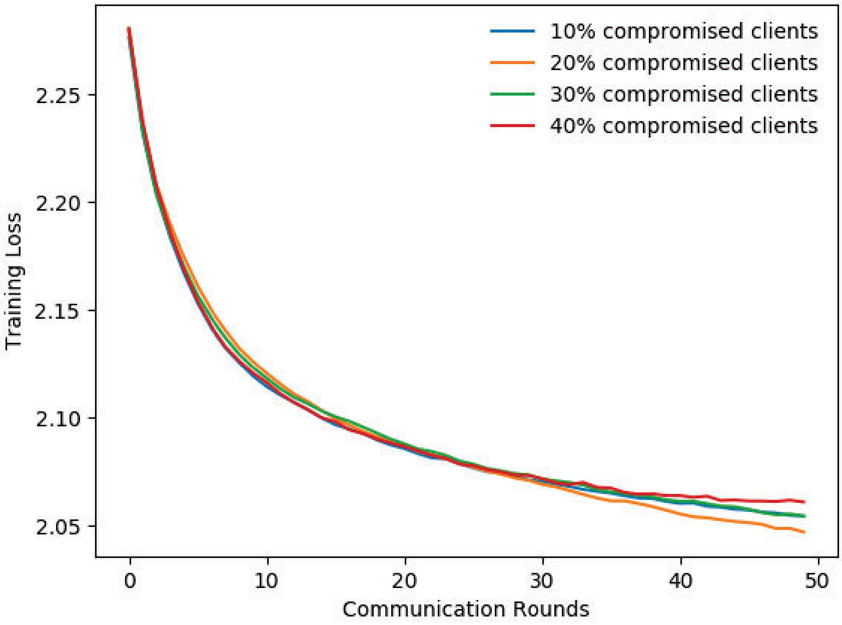


Figure 6.4: Model convergence with respect to compromised clients ratio on CIFAR-10.

6.3 Evaluation on Our Attacks

The empirical results for our attacks are shown in Table 6.2 and Table 6.3. The error rates after our proposed attacks are significantly higher than those after Gaussian attacks on both CNN and MLP models. In particular, when we use our proposed method to attack Trimmed Mean with CNN model as the classifier, the error rate achieves 85%, while Gaussian attack only results in an error rate of 5%. We also notice that the FedAvg aggregator hardly defends against adversarial attacks, which needs to be replaced with a robust aggregation rule.

Table 6.2: Error Rates on CNN Model After Attacks

	No Attack	Gaussian Attack	**Proposed Attack**
FedAvg	0.03	0.24	**0.90**
Krum	0.03	0.03	**0.14**
Trimmed Mean	0.03	0.05	**0.85**

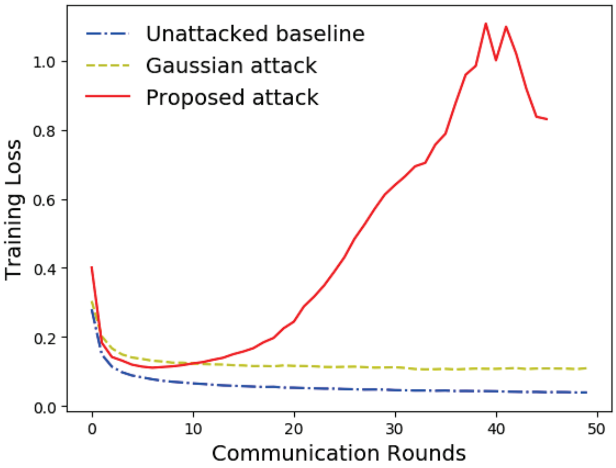
Table 6.3: Error Rates on MLP Model After Attacks

	No Attack	Gaussian Attack	**Proposed Attack**
FedAvg	0.59	0.73	**0.91**
Krum	0.59	0.61	**0.63**
Trimmed Mean	0.59	0.59	**0.65**

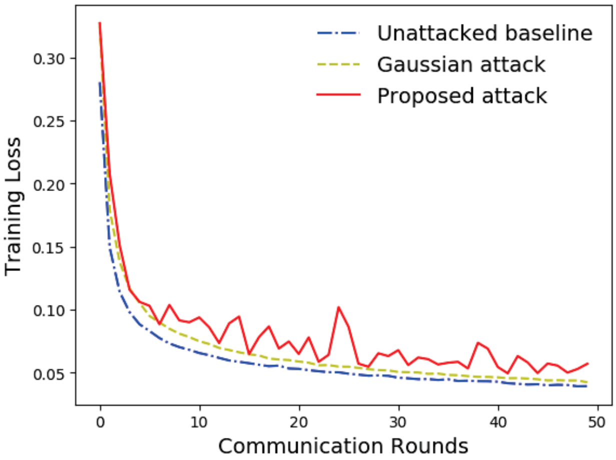
We also explore the model convergence under these attacks for three aggregation rules respectively. In Figure 6.5 and Figure 6.6, it can be seen that our proposed attacks effectively slow down the model convergence when Krum is the aggregation rule in the server, and even lead the model to divergence in the presence of FedAvg and Trimmed Mean.

6.4 Evaluation on Our Defenses

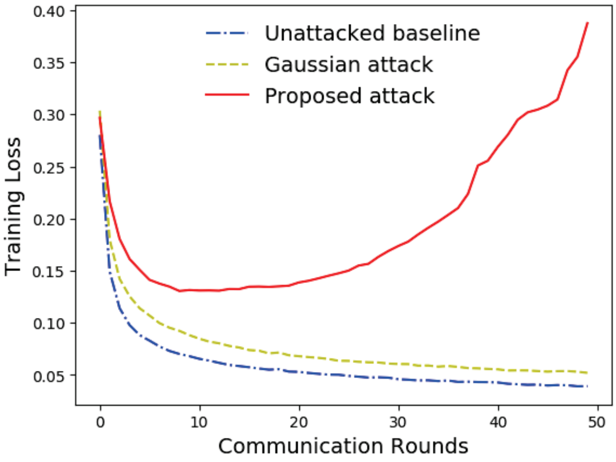
We evaluate our proposed defense algorithm on MNIST and CIFAR-10 datasets using CNN and MLP models respectively. We use the error rate as test metrics for the evaluation of our defense, which is defined in (6.1).



(a) FedAvg

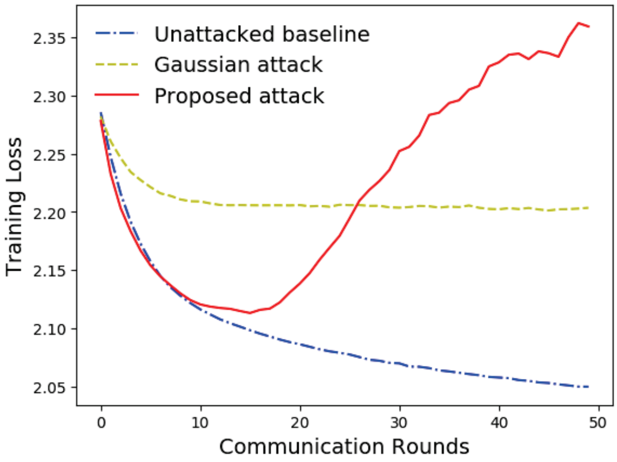


(b) Krum

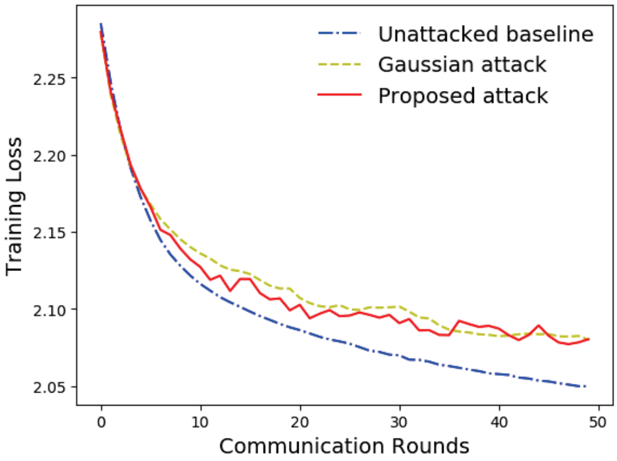


(c) Trimmed Mean

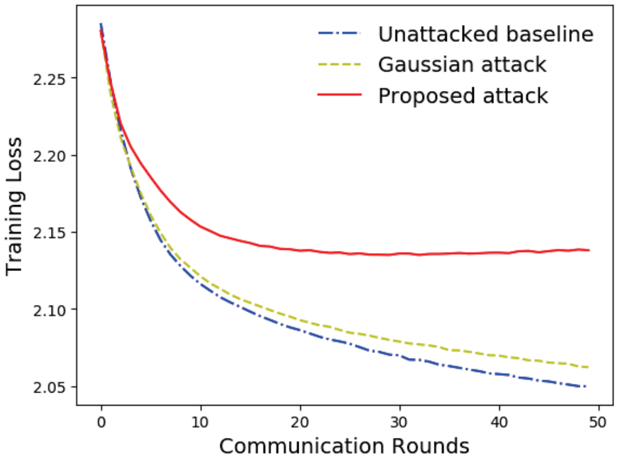
Figure 6.5: Model convergence for different attacks on CNN model.



(a) FedAvg



(b) Krum



(c) Trimmed Mean

Figure 6.6: Model convergence for different attacks on MLP model.

$$Error\ Rate = 1 - Testing\ Accuracy \tag{6.1}$$

As shown in Table 6.4 and Table 6.5, our defense is effective at optimizing the training loss for all scenarios. For example, when defending our proposed attack on CNN model, the error rate remains at 3%, which is same as it when there is no attack. However, Krum and Trimmed Mean are not effective to defend our proposed attack. The error rates increase to 14% and 85% respectively. According to (6.1), a small error rate reflects high testing accuracy, which means that the training loss achieves to a local minimum after 50 rounds of training. This explains why our proposed defense method works, which eliminates the impacts of our proposed attack on both CNN and MLP models.

Table 6.4: Error Rates on CNN Model After Attacks for Defense Results

	No Attack	Gaussian Attack	**Proposed Attack**
Krum	0.03	0.03	0.14
Trimmed Mean	0.03	0.05	0.85
Proposed Defense	**0.03**	**0.03**	**0.03**

Table 6.5: Error Rates on MLP Model After Attacks for Defense Results

	No Attack	Gaussian Attack	**Proposed Attack**
Krum	0.59	0.61	0.63
Trimmed Mean	0.59	0.59	0.65
Proposed Defense	**0.59**	**0.59**	**0.59**

Chapter 7

Conclusion and Future Work

This thesis comprehensively analyzes diverse attacks and defenses on federated learning. We first present preliminary knowledge of federated learning and differential privacy. The application of differential privacy in the context of federated learning is user-level differentially private. Our work is to evaluate the vulnerabilities of differential privacy based federated learning and explore possible defense mechanism. For these aims of research, we have made three major contributions as follows.

1. Sybil Attack Implementation.

   We simulate a federated learning framework and implement user-level differential privacy in the system, namely *differential privacy based federated learning*. In this framework, we perform sybil attacks with different settings of attack intensity and privacy protection level.

2. Detection and Defense against Sybil Attacks.

   Our proposed defense mechanism does not consider the explicit number of compromised clients. Without this assumption, it is more practical to defend against adversarial attacks in industrial applications.

3. Thorough Evaluation of Comparison with State-of-the-art Methods.

   We conduct experimental evaluation to demonstrate that our proposed sybil attacks evidently spoof recent Byzantine-resilient aggregators. Furthermore, the experiment results show that our proposed defense method outperforms these aggregation defense mechanisms in differential privacy based federated learning settings.

Our work raises the interests in the research direction of differential privacy based federated learning. We also explore the attacks and defenses mechanisms in relation to untargeted model poisoning attacks. In future research, targeted model poisoning attacks that are strongly related to data poisoning attacks will be investigated in the real-world application scenarios such as Internet of Vehicles.

Bibliography

[1] Martin Abadi, Andy Chu, Ian 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, pages 308–318, 2016.

[2] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In Advances in Neural Information Processing Systems, pages 4613–4623, 2018.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. arXiv preprint arXiv:1807.00459, 2018.

[4] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In Advances in Neural Information Processing Systems, pages 8635–8645, 2019.

[5] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin B. Calo. Analyzing federated learning through an adversarial lens. In Proceedings of the 36th International Conference on Machine Learning, ICML, pages 634–643, 2019.

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In Proceedings of the 29th International Conference on Machine Learning, ICML, 2012.

[7] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems, pages 119–129, 2017.

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526, 2017.

[9] John R Douceur. The sybil attack. In International workshop on peer-to-peer systems, pages 251–260. Springer, 2002.

[10] Cynthia Dwork. Differential privacy: A survey of results. In International conference on theory and applications of models of computation, pages 1–19. Springer, 2008.

[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, pages 265–284. Springer, 2006.

[12] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407, 2014.

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. arXiv preprint arXiv:1911.11815, 2019.

[14] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866, 2018.

[15] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557, 2017.

[16] Benjamin Grimmer. Convergence rates for deterministic and stochastic subgradient methods without lipschitz continuity. SIAM Journal on Optimization, 29(2):1350–1365, 2019.

[17] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977, 2019.

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Proceedings of the 34th International Conference on Machine Learning, ICML, volume 70, pages 1885–1894, 2017.

[19] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527, 2016.

[20] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492, 2016.

[21] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset.

[22] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. ACM Trans. Program. Lang. Syst., 4(3):382–401, 1982.

[23] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database.

[24] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In IEEE Symposium on Security and Privacy (SP), pages 656–672, 2019.

[25] Yong Li, Yipeng Zhou, Alireza Jolfaei, Dongjin Yu, Gaochao Xu, and Xi Zheng. Privacy-preserving federated learning framework based on chained secure multi-party computing. IEEE Internet of Things Journal, 2020.

[26] 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.

[27] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, pages 4732–4738, 2019.

[28] 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, pages 1273–1282, 2017.

[29] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629, 2016.

[30] 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.

[31] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. arXiv preprint arXiv:1912.13445, 2019.

[32] Lili Su and Nitin H Vaidya. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In Proceedings of the 2016 ACM symposium on principles of distributed computing, pages 425–434, 2016.

[33] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963, 2019.

[34] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In IEEE Symposium on Security and Privacy (SP), pages 466–479, 2020.

[35] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In International Conference on Machine Learning, ICML, pages 6893–6901, 2019.

[36] 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, pages 5636–5645, 2018.

[37] Chong Zhang, Xiao Liu, Xi Zheng, Rui Li, and Huai Liu. Fenghuolun: A federated learning based edge computing platform for cyber-physical systems. In 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pages 1–4. IEEE, 2020.

[38] Tiehua Zhang, Zhishu Shen, Jiong Jin, and Xi Zheng. A democratically collaborative learning scheme for fog-enabled pervasive environments. In 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pages 1–4. IEEE, 2020.

[39] Tiehua Zhang, Zhishu Shen, Jiong Jin, Xi Zheng, Atsushi Tagami, and Xianghui Cao. Achieving democracy in edge intelligence: A fog-based collaborative learning scheme. IEEE Internet of Things Journal, 2020.

[40] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In Advances in Neural Information Processing Systems, pages 14774–14784, 2019.