CRFL: Certifiably Robust Federated Learning against Backdoor Attacks

Chulin Xie [1] **Minghao Chen** [2] **Pin-Yu Chen** [3] **Bo Li** [1]

Abstract

Federated Learning (FL) as a distributed learning paradigm that aggregates information from diverse clients to train a shared global model, has demonstrated great success. However, malicious clients can perform poisoning attacks and model replacement to introduce backdoors into the trained global model. Although there have been intensive studies designing robust aggregation methods and empirical robust federated training protocols against backdoors, existing approaches lack *robustness certification*. This paper provides the first general framework, Certifiably Robust Federated Learning (CRFL), to train certifiably robust FL models against backdoors. Our method exploits clipping and smoothing on model parameters to control the global model smoothness, which yields a sample-wise robustness certification on backdoors with limited magnitude. Our certification also specifies the relation to federated learning parameters, such as poisoning ratio on instance level, number of attackers, and training iterations. Practically, we conduct comprehensive experiments across a range of federated datasets, and provide the first benchmark for certified robustness against backdoor attacks in federated learning. Our code is publicaly available at https://github.com/AI-secure/CRFL.

1. Introduction

Federated learning (FL) has been widely applied to different applications given its high efficiency and privacy-preserving properties (Smith et al., 2017; McMahan et al., 2017a; Zhao et al., 2018). However, recent studies show that it is easy for the local client to add adversarial perturbation such as "backdoors" during training to compromise the final aggregated model (Bhagoji et al., 2019; Bagdasaryan et al., 2020; Wang et al., 2020; Xie et al., 2019). Such attacks raise great security concerns and have become the roadblocks towards the real-world deployment of federated learning.

[1]University of Illinois at Urbana-Champaign [2]Zhejiang University [3]IBM Research. Correspondence to: Chulin Xie <chulinx2@illinois.edu>, Pin-Yu Chen <pin-yu.chen@ibm.com>, Bo Li <lbo@illinois.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

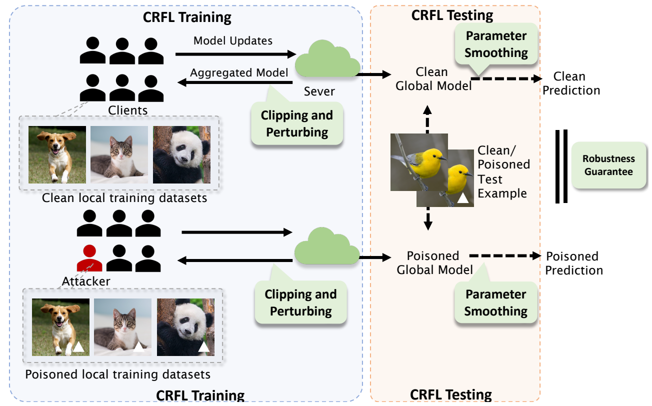


Figure 1. Overview of certifiably robust federated learning (CRFL)

Although there have been intensive studies on robust FL by designing robust aggregation methods (Fung et al., 2020; Pillutla et al., 2019; Fu et al., 2019; Blanchard et al., 2017; El Mhamdi et al., 2018; Chen et al., 2017; Yin et al., 2018), developing empirically robust federated training protocols (e.g., gradient clipping (Sun et al., 2019), leveraging noisy perturbation (Sun et al., 2019) and additional evaluation during training (Andreina et al., 2020)), current defense approaches lack robustness guarantees against the backdoor attacks under certain conditions. To the best of our knowledge, certified robustness analysis and algorithms for FL against backdoor attacks remain elusive.

To bridge this gap, in this work we propose a certifiably robust federated learning (CRFL) framework as illustrated in Figure 1. In particular, during *training*, we allow the local agent to update their model parameters to the center server, and the server will: 1) aggregate the collected model updates, 2) clip the norm of the aggregated model parameters, 3) add a random noise to the clipped model, and finally 4) send the new model parameters back to each agent. Note that all of the operations are conducted on the server side to reduce the load for local clients and to prevent malicious clients. During *testing*, the server will smooth the final global model with randomized parameter smoothing and make the final prediction based on the parameter-smoothed model.

Using CRFL, we theoretically prove that the trained global model would be certifiably robust against backdoors as long as the backdoor is within our certified bound. To obtain such robustness certification, we first quantify the closeness of models aggregated in each step by viewing this process as a Markov Kernel (Asoodeh & Calmon, 2020; Makur, 2019; Polyanskiy & Wu, 2015; 2017). We then leverage the model closeness together with the parameter smoothing procedure to certify the final prediction. Empirically, we conduct extensive evaluations on MNIST, EMNIST, and financial datasets to evaluate the certified robustness of CRFL and study how FL parameters affect certified robustness.

Technical Contributions. In this paper, we take the *first* step towards providing certified robustness for FL against backdoor attacks. We make contributions on both theoretical and empirical fronts.

- We propose the first certifiably robust federated learning (CRFL) framework against backdoor attacks.
- Theoretically, we analyze the training dynamics of the aggregated model via Markov Kernel, and propose parameter smoothing for model inference. Altogether, we prove the certified robustness of CRFL.
- We conduct extensive experiments on MNIST, EMNIST, and financial datasets to show the effect of different FL parameters (e.g. poisoning ratio, number of attackers, and training iterations) on certified robustness.

2. Related work

Backdoor Attacks on Federated Learning The goal of backdoor attacks against federated learning is to train strong poisoned local models and submit malicious model updates to the central server, so as to mislead the global model (Bhagoji et al., 2019). (Bagdasaryan et al., 2020) studies the model replacement approach, where the attacker scales malicious model updates to replace the global model with local backdoored one. (Xie et al., 2019) exploit the decentralized nature of federated learning and propose a distributed backdoor attack.

Robust Federated Learning In order to nullify the effects of attacks while aggregating client updates, a number of robust aggregation algorithms have been proposed for distributed learning (Fung et al., 2020; Pillutla et al., 2019; Fu et al., 2019; Blanchard et al., 2017; El Mhamdi et al., 2018; Chen et al., 2017; Yin et al., 2018). These methods either identify and down-weight the malicious updates through certain distance or similarity metrics, or estimate a true "center" of the received model updates rather than taking a weighted average. However, many of those methods assume that the data distribution is i.i.d cross distributed clients, which is not the case in FL setting. Other defenses are several robust federated protocols that mitigate poisoning attacks during training. (Andreina et al., 2020) incorporates an additional validation phase to each round of FL to detect backdoor. (Sun et al., 2019) show that clipping the norm of model updates and adding Gaussian noise can mitigate backdoor attacks that are based on the model replacement paradigm. None of these provides certified robustness guarantees.

A concurrent work (Cao et al., 2021) proposes Ensemble FL for provable secure FL against malicious clients, which requires training *hundreds of* FL models and focuses on client-level certification. Our work allows standard FL protocol, and our certification is applicable to feature, sample, and client levels.

3. Preliminaries

3.1. Federated Averaging

Learning Objective Suppose the model parameters are denoted by $w \in \mathbb{R}^d$, we consider the following distributed optimization problem: $\min_{w\in\mathbb{R}^d}\{F(w) \triangleq \sum_{i=1}^{N} p_i F_i(w)\}$, where N is the number of clients, and p_i is the aggregation weight of the i-th client such that $p_i \geq 0$ and $\sum_{i=1}^{N} p_i = 1$. Suppose the i-th client holds n_i training data in its local dataset $S_i = \{z_1^i, z_2^i, \ldots, z_{n_i}^i\}$. The local objective $F_i(\cdot)$ is defined by $F_i(w) \triangleq \frac{1}{n_i}\sum_{j=1}^{n_i} \ell(w; z_j^i)$, where $\ell(\cdot;\cdot)$ is a defined learning loss function.

One Round of Federated Learning (Periodic Averaging SGD) In federated learning, the clients are able to perform multiple local iterations to update the local models (McMahan et al., 2017b). So we formulate the SGD problem in FL as Periodic Averaging SGD (Wang & Joshi, 2019; Li et al., 2020). Specifically, at round t, first, the central sever sends current global model w_{t-1} to all clients. Second, every client i initializes its local model $w^i_{(t-1)\tau_i} = w_{t-1}$ and then performs τ_i ($\tau_i \geq 1$) local updates, such that $w_s^i \leftarrow w_{s-1}^i - \eta_i g_i(w_{s-1}^i; \xi_{s-1}^i)$, $s = (t-1)\tau_i + 1, (t-1)\tau_i + 2, \ldots, t\tau_i$, where η_i is the learning rate, $\xi_s^i \subset S^i$ are randomly sampled mini-batches with batch size n_{B_i}, and $g_i(w;\xi^i) = \frac{1}{n_{B_i}}\sum_{z_j^i\in\xi^i} \nabla\ell(w; z_j^i)$ denotes the stochastic gradient. The local clients send the local model updates $w_{t\tau_i}^i - w_{t-1}$ to the server. Finally, the server aggregates over the local model updates into the new global model w_t such that $w_t \leftarrow w_{t-1} + \sum_{i=1}^{N} p_i(w_{t\tau_i}^i - w_{t-1})$.

3.2. Threat Model

The goal of backdoor is to inject a backdoor pattern during training such that any test input with such pattern will be misclassified as the target label (Gu et al., 2019). The purpose of backdoor attacks in FL is to manipulate local models and simultaneously fit the main task and backdoor task, so that the global model would behave normally on untampered data samples while achieving high attack success rate on backdoored data samples. We consider the backdoor attack via model replacement approach (Bagdasaryan et al., 2020) where the attackers train the local models using the poisoned datasets, and scale the malicious updates before sending it to the sever. Suppose there are R adversarial clients out of N clients, we assume that each of them only attack once and they perform model replacement attack together at the same round $\mathsf{t_{adv}}$. Such distributed yet coordinated backdoor attack is shown to be effective in (Xie et al., 2019).

Let $D := \{S_1, S_2, \ldots, S_N\}$ be the union of original benign local datasets in all clients. For a data sample $z_j^i := \{x_j^i, y_j^i\}$ in S_i, we denote its backdoored version as ${z'}_j^i := \{x_j^i + \delta_{ix}, y_j^i + \delta_{iy}\}$, and the backdoor as $\delta_i := \{\delta_{ix}, \delta_{iy}\}$. We assume an adversarial client i has q_i backdoored samples in its local dataset S'^i with size n_i. Let $D' := \{S'_1, \ldots, S'_{R-1}, S'_R, S_{R+1}, \ldots, S_N\}$ be the union of local datasets in the adversarial round $\mathsf{t_{adv}}$. Then we have $D' = D + \{\{\delta_i\}_{j=1}^{q_i}\}_{i=1}^{R}$.

Before $\mathsf{t_{adv}}$, the adversarial clients train the local model using original benign datasets. When $t = \mathsf{t_{adv}}$, for adversarial client i, each local iteration is trained on the backdoored local dataset S'_i such that ${w'}_s^i \leftarrow {w'}_{s-1}^i - \eta_i g_i({w'}_{s-1}^i; {\xi'}_{s-1}^i)$, $s = (t-1)\tau_i + 1, (t-1)\tau_i + 2, \ldots, t\tau_i$, where w' is the malicious model parameters, ${\xi'}^i$ is the mini-batch sampled from S'_i and the local model is initialized as ${w'}_{(t-1)\tau_i}^i = w_{t-1}$. Following (Bagdasaryan et al., 2020), we assume the attacker add a fixed number of backdoored samples q_{B_i} in each training batch, then the mini-batch gradient is $g_i(w'; {\xi'}^i) = \frac{1}{n_{B_i}}(\sum_{j=1}^{q_{B_i}} \nabla \ell(w'; {z'}_j^i) + \sum_{j=q_{B_i}+1}^{n_{B_i}} \nabla \ell(w'; z_j^i))$. The poison ratio of dataset S'_i is $q_{B_i}/n_{B_i} = q_i/n_i$. Since for each local iteration, the local model is updated with backdoored mini-batch samples, more local iterations will drive the local model ${w'}_s^i$ farther from the corresponding one w_s^i in benign training process. Then the adversarial clients scale their malicious local updates before submitting to the server. Let the scale factor be γ_i for i-th adversarial client, then the scaled update is $\gamma_i({w'}_{\mathsf{t_{adv}}\tau_i}^i - w_{\mathsf{t_{adv}}-1})$. The server aggregates over the malicious and benign updates into an infected global model w'_t such that $w'_t \leftarrow w_{t-1} + \sum_{i=1}^{R} p_i \gamma_i ({w'}_{t\tau_i}^i - w_{t-1}) + \sum_{i=R+1}^{N} p_i (w_{t\tau_i}^i - w_{t-1})$.In fact, even though the adversarial clients only attack at round $\mathsf{t_{adv}}$ and in the later rounds ($t > \mathsf{t_{adv}}$) they use the original benign datasets, the global model is already infected starting from $\mathsf{t_{adv}}$, so we still denote the global model parameters as w'_t in later rounds.

4. Methodology

In this Section, we introduce our proposed framework CRFL, which is composed of a training-time subroutine (Algorithm 1) and a test-time subroutine (Algorithm 2) for achieving certified robustness.

4.1. CRFL Training: Clipping and Perturbing

During training, at round $t = 1, \ldots, T-1$, local clients update their models, and the server performs aggregation. Then, in our training protocol, the server clip the model parameters $\mathrm{Clip}_{\rho_t}(w_t) \leftarrow w_t / \max(1, \frac{\|w_t\|}{\rho_t})$ so that its norm is bounded by ρ_t, and then add isotropic Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I})$ directly on the aggregated global model parameters (coordinate-wise noise): $\widetilde{w}_t \leftarrow \mathrm{Clip}_{\rho_t}(w_t) + \epsilon_t$. Throughout this paper, $\|\cdot\|$ denotes the ℓ_2 norm $\|\cdot\|_2$. In the next round $t+1$, client i initializes its local model with noisy new global model $w_{t\tau_i}^i \leftarrow \widetilde{w}_t$. In the final round T, we only clip the global model parameters.

The procedure is summarized in Algorithm 1 and denoted by $\mathcal{M}$, which outputs the global model parameters $\mathrm{Clip}_{\rho_T}(w_T)$. Then we define $\mathcal{M}(D) := \mathrm{Clip}_{\rho_T}(w_T)$.

Algorithm 1 Federated averaging with parameters clipping and perturbing

Server's input: initial model parameters w_0, $\widetilde{w}_0 \leftarrow w_0$
Client i's input: local dataset S_i and learning rate η_i
for each round $t = 1, \ldots, T$ **do**
  The server sends $\widetilde{w}_{t-1}$ to the i-th client
  for client $i = 1, 2, \ldots, N$ in parallel **do**
    initialize local model $w_{(t-1)\tau_i}^i \leftarrow \widetilde{w}_{t-1}$
    for local iteration $s = (t-1)\tau_i + 1, \ldots, t\tau_i$ **do**
      compute mini-batch gradient $g_i(\cdot;\cdot)$
      $w_s^i \leftarrow w_{s-1}^i - \eta_i g_i(w_{s-1}^i; \xi_{s-1}^i)$
    end for
    The i-th client sends $w_{t\tau_i}^i - \widetilde{w}_{t-1}$ to the server
  end for
  The server updates the model parameters
  $w_t \leftarrow \widetilde{w}_{t-1} + \sum_{i=1}^{N} p_i (w_{t\tau_i}^i - \widetilde{w}_{t-1})$
  The server clips the model parameters
  $\mathrm{Clip}_{\rho_t}(w_t) \leftarrow w_t / \max(1, \frac{\|w_t\|_2}{\rho_t})$
  The server adds noise
  if $t \le T-1$ **then**
    $\epsilon_t \leftarrow$ a sample drawn from $\mathcal{N}(0, \sigma_t^2 \mathbf{I})$
    $\widetilde{w}_t \leftarrow \mathrm{Clip}_{\rho_t}(w_t) + \epsilon_t$
  end if
end for
Output: Clipped global model parameters $\mathrm{Clip}_{\rho_T}(w_T)$

4.2. CRFL Testing: Parameter Smoothing

Smoothed Classifiers We study multi-class classification models and define a classifier $h : (\mathcal{W}, \mathcal{X}) \to \mathcal{Y}$ with finite set of label $\mathcal{Y} = \{1, \ldots, C\}$, where C denotes the number of classes. We extend the randomized smoothing method (Cohen et al., 2019) to *parameter smoothing* for constructing a new, "smoothed" classifier h_s from an arbitrary base classifier h. The robustness properties can be verified using the smoothed classifier h_s. Given the model parameter w of h, when queried at a test sample x_{test}, we first take a majority vote over the predictions of the base classifier h on random model parameters drawn from a probability distribution μ, i.e., the smoothing measure, to obtain the "votes" $H_s^c(w; x_{test})$ for each class $c \in \mathcal{Y}$. Then the label returned by the smoothed classifier h_s is the mostly probable label among all classes (the majority vote winner). Formally,

$$h_s(w; x_{test}) = \arg\max_{c \in \mathcal{Y}} H_s^c(w; x_{test}), \\ \text{where } H_s^c(w; x_{test}) = \mathbb{P}_{W \sim \mu(w)}[h(W; x_{test}) = c]. \tag{1}$$

To be aligned with the training time Gaussian noise (perturbing), we also adopt Gaussian smoothing measures $\mu(w) = \mathcal{N}(w, {\sigma_T}^2\mathbf{I})$ during testing time. In practice, the exact value of the probability $p_c = \mathbb{P}_{W \sim \mu(w)}[h(W; x_{test}) = c]$ for label c is difficult to obtain for neural networks, and hence we resort to Monte Carlo estimation (Cohen et al., 2019; Lecuyer et al., 2019) to get its approximation $\hat{p}_c$. At round $t = T$, given the clipped aggregated global model $\text{Clip}_{\rho_T}(w_T)$, we add Gaussian noise $\epsilon_T^k \sim \mathcal{N}(0, \sigma_T^2\mathbf{I})$ for M times to get M sets of noisy model parameters (M Monte Carlo samples for estimation), such that $\widetilde{w}_T^k \leftarrow \text{Clip}_{\rho_T}(w_T) + \epsilon_T^k,\ k = 1, 2, \ldots, M.$

In Algorithm 2, The function GetCounts runs the classifier with each set of noisy model parameters w_T^k for one test sample x_{test}, and returns a vector of class counts. Then we take the most probable class $\hat{c}_A$ and the runner-up class $\hat{c}_B$ to calculate the corresponding $\hat{p}_A$ and $\hat{p}_B$. The function CalculateBound calibrates the empirical estimation to bound the probability α of h_s returning an incorrect label. Given the error tolerance α, we use Hoeffding's inequality (Hoeffding, 1994) to compute a lower bound $\underline{p_A}$ on the probability $H_s^{c_A}(w; x_{test})$ and a upper bound $\overline{p_B}$ on the probability $H_s^{c_B}(w; x_{test})$ according to $\underline{p_A} = \hat{p}_A - \sqrt{\frac{\log(1/\alpha)}{2N}}$, $\overline{p_B} = \hat{p}_B + \sqrt{\frac{\log(1/\alpha)}{2N}}$. We leave the function CalculateRadius to be defined with our main results in later sections and we will analyze the robustness properties of the model trained and tested under our framework CRFL.

Comparison with Certifiably Robust Models in Centralized Setting Our method is different from previous certifiably robust models in centralized learning against evasion attacks (Cohen et al., 2019) and backdoors (Weber et al., 2020). Once the M noisy models (at round T, with σ_T) are generated, they are fixed and used for every test sample during test time, just like RAB (Weber et al., 2020) in the centralized setting. However, RAB actually trains M models using M noise-corrupted datasets, while we just train one model through FL and finally generated M noise-corrupted copies of it. For every test sample, randomized smoothing (Cohen et al., 2019) generates M noisy samples. Suppose the test set size is m. Then during testing, there are $m \cdot M$ times noise addition on test samples for randomized smoothing, and M times noises addition on

Algorithm 2 Certification of parameters smoothing

Input: a test sample x_{test} with true label y_{test}, the global model parameters $\text{Clip}_{\rho_T}(w_T)$, the classifier $h(\cdot,\cdot)$
for $k = 0, 1, \ldots M$ **do**
  $\epsilon_T^k \leftarrow$ a sample drawn from $\mathcal{N}(0, \sigma_T^2\mathbf{I})$
  $\widetilde{w}_T^k = \text{Clip}_{\rho_T}(w_T) + \epsilon_T^k$
end for
Calculate empirical estimation of p_A, p_B for x_{test}
$\texttt{counts} \leftarrow \text{GetCounts}(x_{test}, \{\widetilde{w}_T^1, \ldots, \widetilde{w}_T^M\})$
$\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in $\texttt{counts}$
$\hat{p}_A, \hat{p}_B \leftarrow \texttt{counts}[\hat{c}_A]/M, \texttt{counts}[\hat{c}_B]/M$
Calculate lower and upper bounds of p_A, p_B
$\underline{p_A}, \overline{p_B} \leftarrow \text{CalculateBound}(\hat{p}_A, \hat{p}_B, N, \alpha)$
if $\underline{p_A} > \overline{p_B}$ **then**
  $\mathsf{RAD} = \text{CalculateRadius}(\underline{p_A}, \overline{p_B})$
  Output: Prediction $\hat{c}_A$ and certified radius RAD
else
  Output: `ABSTAIN` and 0
end if

trained model for CRFL. To our best knowledge, this is the first work to study *parameter* smoothing rather than input smoothing, which is an open problem motivated by the FL scenario, since the sever directly aggregates over the model parameters.

5. Certified Robustness of CRFL

5.1. Pointwise Certified Robustness

Goal of Certification In the context of data poisoning in federated learning, the goal is to protect the global model against adversarial data modification made to the local training sets of distributed clients. Thus, the goal of certifiable robustness in federated learning is for each test point, to return a prediction as well as a certificate that the prediction would not change had some features in (part of) local training data of certain clients been modified.

Following our threat model in Section 3.2 and our training protocol in Algorithm 1, we define the trained global model $\mathcal{M}(D') := \text{Clip}_{\rho_T}(w'_T)$. For the FL training process that is exposed to model replacement attack, when the distance between D' (backdoored dataset) and D (clean dataset) is under certain threshold (i.e., the magnitude of $\{\{\delta_i\}_{j=1}^{q_i}\}_{i=1}^{R}$ is bounded), we can certify that $\mathcal{M}(D')$ is "close" to $\mathcal{M}(D)$ and thus is robust to backdoors. The rationale lies in the fact that we perform clipping and noise perturbation on the model parameters to control the global model deviation during training. During testing, intuitively, under the Gaussian smoothing measures μ as described in Algorithm 2, for two close distribution $\mu(\mathcal{M}(D'))$ and $\mu(\mathcal{M}(D))$, we would expect that even though the probabilities for each class c, i.e., $H_s^c(\mathcal{M}(D'); x_{test})$ and $H_s^c(\mathcal{M}(D); x_{test})$, may not be equal, the returned most likely label $h_s(\mathcal{M}(D'); x_{test})$ and $h_s(\mathcal{M}(D); x_{test})$ should be consistent.

In summary, we aim to develop a robustness certificate by studying under what condition for $\{\{\delta_i\}_{j=1}^{q_i}\}_{i=1}^{R}$ that the prediction for a test sample is consistent between the smoothed FL models trained from D and D' separately, i.e., $h_s(\mathcal{M}(D'); x_{test}) = h_s(\mathcal{M}(D); x_{test})$. To put forth our certified robustness analysis, we make the following assumptions on the loss function of all clients. Then we present our main theorem and explain its derivation through *model closeness* and *parameter smoothing*. Throughout this paper, we denote $\nabla_w \ell(w; z)$ as $\nabla \ell(w; z)$ for simplicity.

Assumption 1 (Convexity and Smoothness)**.** *The loss function $\ell(w; z)$ is β-smoothness, i.e, $\forall w_1, w_2$,*

$$\|\nabla \ell(w_1; z) - \nabla \ell(w_2; z)\| \leq \beta \|w_1 - w_2\|.$$

In addition, the loss function $\ell(w; z)$ is convex. Then co-coercivity of the gradient states:

$$\|\nabla \ell(w_1; z) - \nabla \ell(w_2; z)\|^2 \\ \leq \beta \langle w_1 - w_2, \nabla \ell(w_1; z) - \nabla \ell(w_2; z)\rangle.$$

Assumption 2 (Lipschitz Gradient w.r.t. Data)**.** *The gradient $\nabla_w l(z; w)$ is $L_{\mathcal{Z}}$ Lipschitz with respect to the argument z and norm distance $\|\cdot\|$, i.e, $\forall z_1, z_2$,*

$$\|\nabla \ell(w; z_1) - \nabla \ell(w; z_2)\| \leq L_{\mathcal{Z}} \|z_1 - z_2\|.$$

Assumption 3. *The whole FL system follows Algorithm 1 to train and Algorithm 2 to test.*

The assumptions on convexity and smoothness are common in the analysis of distributed SGD (Li et al., 2020; Wang & Joshi, 2019). We also make assumption on the Lipschitz gradient w.r.t. data, which is used in (Fallah et al., 2020; Reisizadeh et al., 2020) for analyzing the heterogeneous data distribution across clients.

Main Results

Theorem 1 (General Robustness Condition)**.** *Let h_s be defined as in Eq. 1. When $\eta_i \leq \frac{1}{\beta}$ and Assumptions 1, 2, and 3 hold, suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy*

$$H_s^{c_A}(\mathcal{M}(D'); x_{test}) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} H_s^{c}(\mathcal{M}(D'); x_{test}),$$

then if

$$R \sum_{i=1}^{R} (p_i \gamma_i \tau_i \eta_i \frac{q_{B_i}}{n_{B_i}} \|\delta_i\|)^2 \quad \leq \frac{-\log\left(1 - (\sqrt{\underline{p_A}} - \sqrt{\overline{p_B}})^2\right) \sigma_{\mathsf{t_{adv}}}^2}{2L_{\mathcal{Z}}^2 \prod_{t=\mathsf{t_{adv}}+1}^{T} \left(2\Phi\left(\frac{\rho_t}{\sigma_t}\right) - 1\right)},$$

it is guaranteed that

$$h_s(\mathcal{M}(D'); x_{test}) = h_s(\mathcal{M}(D); x_{test}) = c_A,$$

where Φ is standard Gaussian's cumulative density function (CDF) and the other parameters are defined in Section 3.

In practice, since the server does not know the global model in the current FL system is poisoned or not, we assume the model is already backdoored and derive the condition when its prediction will be certifiably consistent with the prediction of the clean model. Our certification is on three levels: *feature*, *sample*, and *client*. If the magnitude of the backdoor is upper bounded for every attackers, then we can re-write the Theorem 1 as the following corollary.

Corollary 1 (Robustness Condition in Feature Level)**.** *Using the same setting as in Theorem 1 but further assume identical backdoor magnitude $\|\delta\| = \|\delta_i\|$ for $i = 1, \ldots, R$. Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy*

$$H_s^{c_A}(\mathcal{M}(D'); x_{test}) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} H_s^{c}(\mathcal{M}(D'); x_{test}),$$

then $h_s(\mathcal{M}(D'); x_{test}) = h_s(\mathcal{M}(D); x_{test}) = c_A$ for all $\|\delta\| < \mathsf{RAD}$, where

$$\mathsf{RAD} = \sqrt{\frac{-\log\left(1 - (\sqrt{\underline{p_A}} - \sqrt{\overline{p_B}})^2\right) \sigma_{\mathsf{t_{adv}}}^2}{2RL_{\mathcal{Z}}^2 \sum_{i=1}^{R} (p_i \gamma_i \tau_i \eta_i \frac{q_{B_i}}{n_{B_i}})^2 \prod_{t=\mathsf{t_{adv}}+1}^{T} \left(2\Phi\left(\frac{\rho_t}{\sigma_t}\right) - 1\right)}} \tag{2}$$

The function $\mathrm{CalculateRadius}$ in our Algorithm 2 can calculate the certified radius RAD according to Corollary 1.

We now make several remarks about Corollary 1 and will verify them in our experiments: 1) The noise level σ_t and the parameter norm clipping threshold ρ_t are hyper-parameters that can be adjusted to control the robustness-accuracy trade-off. For instance, the certified radius RAD would be large when: σ_t is high; ρ_t is small; the margin between $\underline{p_A}$ and $\overline{p_B}$ is large; the number of attackers R is small; the poison ratio $\frac{q_{B_i}}{n_{B_i}}$ is small; the scale factor γ_i is small; the aggregation weights for attackers p_i is small; the local iteration τ_i is small; and the local learning rate η_i small. 2) Since $0 \leq 2\Phi(\cdot) - 1 \leq 1$, the certified radius RAD goes to ∞ as $T \to \infty$ when $\Phi(\cdot) < 1$. Ituitively, the benign fine-tuning after backdoor injection round $\mathsf{t_{adv}}$ would mitegate the poisoning effect. Thus, with infinite rounds of such fine-tuning, the model is able to tolerate backdoors with arbitrarily large magnitude. In practice, we note that the continued multiplication in the denominator may not approach 0 due to numerical issues, which we will verify in the experiments section. 4) Large number of clients N will decrease the aggregation weights p_i of attackers, thus it can tolerate backdoors with large magnitude, resulting in higher RAD. 5) For general neural networks, efficient computation of Lipschitz gradient constant (w.r.t. data input) is an open question, especially when the data dimension is high. We will provide a closed-form expression for $L_{\mathcal{Z}}$ under some constraints next.

As mentioned in Section 3.2, the backdoor for data sample z_j^i includes both the backdoor pattern δ_{i_x} and adversarial target label flipping δ_{i_y}. In Assumption 2 we define $L_{\mathcal{Z}}$ with $z = \{x, y\}$ (concatenation of x and y) to certify against both backdoor patterns and label-flipping effects. Without loss of generality, here we focus on backdoor patterns considering bounded model parameters in Lemma 1, which provides a closed-form expression for $L_{\mathcal{Z}}$ in the case of multi-class logistic regression. By applying $L_{\mathcal{Z}}$ from Lemma 1 to Theorem 1, it indicates that the prediction for a test sample is independent with the backdoor pattern so the backdoor pattern is disentangled from the adversarial target label.

Lemma 1. *Given the upper bound on model parameters norm, i.e., $\|w\| \le \rho$, and two data samples z_1 and z_2 with $x_1 \neq x_2$ ($y_1 = y_2$), for multi-class logistic regression (i.e., one linear layer followed by a softmax function and trained by cross-entropy loss), its Lipschitz gradient constant w.r.t data is $L_{\mathcal{Z}} = \sqrt{2 + 2\rho + \rho^2}$. That is,*

$$\|\nabla \ell(w; z_1) - \nabla \ell(w; z_2)\| \le \sqrt{2 + 2\rho + \rho^2}\|z_1 - z_2\|.$$

Proof for Lemma 1 is provided in the Appendix B.6.

In order to formally derive the main theorem, there are two key results. We first quantify the closeness between the FL trained models $\mathcal{M}(D')$ and $\mathcal{M}(D)$ using Markov Kernel, and then connect the model closeness to the prediction consistency through parameter smoothing.

5.2. Model Closeness

As described in Algorithm 1, owing to the Gaussian noise perturbation mechanism, in each iteration the global model can be viewed as a random vector with the Gaussian smoothing measure μ. We use the f-divergence between $\mu(\mathcal{M}(D'))$ and $\mu(\mathcal{M}(D))$ as a statistical distance for measuring model closeness of the final FL model. Based on the data post-processing inequality, when we interpret each round of CRFL as a probability transition kernel, i.e., a Markov Kernel, the contraction coefficient of Markov Kernel can help bound the divergence over multiple training rounds of FL.

Let $f : (0, \infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$, μ and ν be two probability distributions. Then the f-divergence is defined as $D_f(\mu||\nu) = E_{W\sim\nu}[f(\frac{\mu(W)}{\nu(W)})]$. Common choices of f-divergence include total variation ($f(x) = \frac{1}{2}\|x - 1\|$) and Kullback-Leibler (KL) divergence ($f(x) = x \log x$). The data processing inequality (Raginsky, 2016; Polyanskiy & Wu, 2015; 2017) for the relative entropy states that, for any convex function f and any probability transition kernel (Markov Kernel), $D_f(\mu K||\nu K) \le D_f(\mu||\nu)$, where μK denotes the push-forward of μ by K, i.e., $\mu K = \int \mu(dW)K(W)$. In other words, $D_f(\mu||\nu)$ decreases by post-processing via K. (Asoodeh & Calmon, 2020) extend it to analyze SGD.

In our setting, all the operations in one round of our CRFL, including SGD, clipping and noise perturbations, are incorporated as a Markov Kernel. We note that in the single-round attack setting, the adversarial clients use clean datasets to train the local models after $\mathsf{t_{adv}}$, so the Markov operator is the same as the one in the benign training process. Therefore the f-divergence of the two global models (backdoored and benign) of interest decreases over rounds, which is characterized by a contraction coefficient defined in Appendix B. We quantify such contraction property of Markov Kernel for each round with the help of two hyperparameters in the server side: model parameter norm clipping threshold ρ_t and the noise level σ_t, and finally bound f-divergence of global models in round T. Although our analysis can be adopted to general f-divergence, we here use KL divergence as an instantiation to measure the model closeness.

Theorem 2. *When $\eta_i \le \frac{1}{\beta}$ and Assumptions 1, 2, and 3 hold, the KL divergence between $\mu(\mathcal{M}(D))$ and $\mu(\mathcal{M}(D'))$ with $\mu(w) = \mathcal{N}(w, {\sigma_T}^2\mathbf{I})$ is bounded as:*

$$D_{KL}(\mu(\mathcal{M}(D))||\mu(\mathcal{M}(D'))) \le \frac{2R\sum_{i=1}^{R}\left(p_i\gamma_i\tau_i\eta_i\frac{q_{B\,i}}{n_{B\,i}}L_{\mathcal{Z}}\|\delta_i\|\right)^2}{\sigma^2_{\mathsf{t_{adv}}}} \prod_{t=\mathsf{t_{adv}}+1}^{T}\left(2\Phi\left(\frac{\rho_t}{\sigma_t}\right) - 1\right)$$

The proof is provided in the Appendix B.

5.3. Parameter Smoothing

We connect the model closeness to the prediction consistency by the following theorem. The smoothed classifier h_s is robustly certified at $\mu(w')$ with respect to the bounded KL divergence, $D_{KL}(\mu(w), \mu(w')) \le \epsilon$.

Theorem 3. *Let h_s be defined as in Eq. 1. Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy*

$$H_s^{c_A}(w'; x_{test}) \ge \underline{p_A} \ge \overline{p_B} \ge \max_{c \neq c_A} H_s^c(w'; x_{test}),$$

then $h_s(w'; x_{test}) = h_s(w; x_{test}) = c_A$ for all w such that $D_{KL}(\mu(w), \mu(w')) \le \epsilon$, where

$$\epsilon = -\log\left(1 - (\sqrt{\underline{p_A}} - \sqrt{\overline{p_B}})^2\right)$$

The proof is provided in the Appendix C.

Finally, combining Theorem 2 and 3 leads to our main Theorem 1. In detail, Theorem 2 states that $D_{KL}(\mu(\mathcal{M}(D))||\mu(\mathcal{M}(D')))$ under our CRFL framework is bounded by certain value that depends on the difference between D and D'. Theorem 3 states that for a test sample x_{test}, as long as the KL divergence is smaller than $-\log(1 - (\sqrt{\underline{p_A}} - \sqrt{\overline{p_B}})^2)$, the prediction from the poisoned smoothed classifier h_s that is built upon the base classifier with model parameter $\mathcal{M}(D')$ will be consistent with the prediction from h_s that is built upon $\mathcal{M}(D)$. Therefore, we derive the condition for D and D' in Theorem 1, under which $D_{KL}(\mu(\mathcal{M}(D))||\mu(\mathcal{M}(D'))) \leq -\log(1 - (\sqrt{p_A} - \sqrt{p_B})^2)$. This condition also indicates that h_s built upon the model parameter $\mathcal{M}(D')$ is certifiably robust.

Defend against Other Potential Attack Here we discuss the potentials to generalize our method against other training-time attacks. 1) Our method can naturally extend to *fixed-frequency* attack by applying our analysis for each attack period. In particular, we can repeatedly apply our Theorem 2 to analyze model closeness for each attack period, and the different initializations of each period can be bounded based on its last period. Then Theorem 3 can be applied to connect model closeness to certify the prediction consistency. 2) (Wang et al., 2020) introduce edge-case adversarial training samples to enforce the model to misclassify inputs on the tail of input distribution. The edge-case attack essentially conducts a special semantic attack (Bagdasaryan et al., 2020) by selecting rare images instead of directly adding backdoor patterns. It is possible to apply our framework against such attack by viewing it as the whole *sample* manipulation.

Comparison with Differentially Private Federated Learning In order to protect the privacy of each client, differentially private federated learning (DPFL) mechanisms are proposed (Geyer et al., 2017; McMahan et al., 2018; Agarwal et al., 2018) to ensure that the learned FL model is essentially unchanged when one individual client is modified. Compared with DPFL, our method has several fundamental differences and addresses additional challenges: 1) Mechanisms: DPFL approaches add training-time noise to provide privacy guarantee, while ours add smoothing noise during training and testing to provide certified robustness against data poisoning. In general, the added noise in CRFL does not need to be as large as that in DPFL to provide *strong* privacy guarantee, and therefore preserve higher model utility. 2) Certification goals: DPFL approaches provide client-level privacy guarantee for the learned model parameters, while in CRFL the robustness guarantee is derived for certified pointwise prediction which could be on the feature, samples and clients levels. 3) Technical contributions: DPFL approaches derive DP guarantee via DP composition theorems (Dwork et al., 2014; Abadi et al., 2016), while we quantify the global model deviation via Markov Kernel and verify the robustness properties of the smoothed model via parameter smoothing.

6. Experiments

In our experiments, the attackers perform the model replacement attack at round $\mathsf{t}_{\mathsf{adv}}$ during our CRFL training, and the server performs parameter smoothing on a possibly backdoored FL model at round T to calculate the certified radius RAD for each test sample based on Corollary 1. Specifically, we evaluate the effect of the training time noise σ_t, the attacker's ability which includes the number of attackers R, the poison ratio $\frac{q_{B_i}}{n_{B_i}}$ and the scale factor γ_i, robust aggregation protocol, the number of total clients N and the number of training rounds T. Moreover, we evaluate the model closeness empirically to justify Theorem 2.

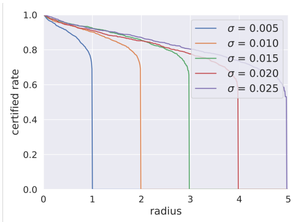
6.1. Experiment Setup

We focus on multi-class logistic regression (one linear layer with softmax function and cross-entropy loss), which is a convex classification problem. We train the FL system following our CRFL framework with three datasets: Lending Club Loan Data (LOAN) (Kan, 2019), MNIST (LeCun & Cortes, 2010), and EMNIST (Cohen et al., 2017). We refer the readers to Appendix A for more details about the datasets, parameter setups and attack setting. We train the FL global model until convergence and then use our certification in Algorithm 2 for robustness evaluation.

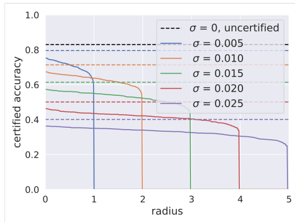
The metrics of interest are *certified rate* and *certified accuracy*. Given a test set of size m, for i-th test sample, the ground truth label is y_i, and the output prediction is either c_i with the certified radius RAD_i or $c_i = \texttt{ABSTAIN}$ with $\mathsf{RAD}_i = 0$. Then we calculate **certified rate** at r as $\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\{\mathsf{RAD}_i \geq r\}$, and **certified accuracy** at r as $\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\{c_i = y_i \text{ and } \mathsf{RAD}_i \geq r\}$. The certified rate is the fraction of the test set that can be certified at radius $\mathsf{RAD} \geq r$, which reveals how consistent the possibly backdoored classifier's prediction with the clean classifier's prediction. The certified accuracy is the fraction of the test set for which the possibly backdoored classifier makes correct and consistent predictions with the clean model. In the displayed figures, there is a critical radius beyond which the certified accuracy and certified rate are dropped to zero. Since each test sample has its own calculated certified radius RAD_i, this critical value is a threshold that none of them have a larger radius than it, similar to the findings in (Cohen et al., 2019). We certified 10000/5000/10000 samples from the LOAN/MNIST/EMNIST test sets. In all experiments, unless otherwise stated, we use $\sigma_T = 0.01$ to generate $M = 1000$ noisy models in parameter smoothing procedure, and use the error tolerance $\alpha = 0.001$. In our experiments, we adopt the expression of $L_{\mathcal{Z}}$ in Lemma 1. $L_{\mathcal{Z}}$ can be generalized to other poisoning settings by specifying z_1, z_2 in Assumption 2 under the case of "$x_1 \neq x_2$ and $y_1 \neq y_2$" or "$x_1 = x_2$ and $y_1 \neq y_2$".

6.2. Experiment Results

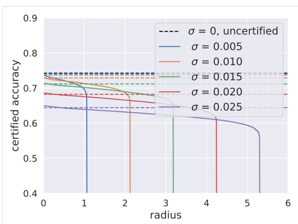
We only change one factor in each experiment and keep others the same as the experiment setup. We plot the certified accuracy and certified rate on the clean test set, and report the results on the backdoored test set in Appendix A.



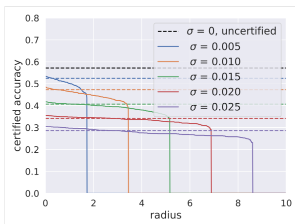
(a) Certified rate on MNIST



(b) Certified acc. on MNIST



(c) Certified acc. on LOAN



(d) Certified acc. on EMNIST

Figure 2. Certified accuracy and certified rate on MNIST, LOAN, and EMNIST with different training-time noise σ. Solid lines represent certified accuracy; dashed lines of the same color show the accuracy of base classifier trained with σ; black dashed line presents the accuracy of the classifier trained without noise.

Effect of Training Time Noise Since we aim to defend against backdoor attack, the training time noise σ ($\sigma = \sigma_t, t < T$) in our Algorithm 1 is more essential than σ_T in parameter smoothing (Algorithm 2). The reason is that σ can nullify the malicious model updates at early stage. Figure 2 plots the certified accuracy and certified rate attained by training FL system with different σ. In Figure 2(a), when σ is high, certified rate is high at every r and large radius can be certified. Figure 2(b)(c)(d) show that large radius is certified but at a low accuracy, so the parameter noise σ controls the trade-off between certifiability and accuracy, which echoes the property of evasion-attack certification (Cohen et al., 2019). Comparing the solid line with the dashed line for each color, we can see that the parameter smoothing with σ_T does not hurt the accuracy much.

Effect of Attacker Ability From the perspective of attackers, the larger number of attackers R, the larger poison ratio $\frac{q_{B_i}}{n_{B_i}}$ and the larger scale factor γ_i result in the stronger attack. Figure 3, Figure 4, and Figure 5 show that in the three datasets, the stronger the attack, the smaller radius can be certified. After training sufficient number of rounds with clean datasets after $\mathsf{t}_{\mathsf{adv}}$, we show that the certified radius is not sensitive to the attack timing $\mathsf{t}_{\mathsf{adv}}$ in Appendix A.2.

Effect of Robust Aggregation Our CRFL can be used to assess different robust aggregation rules. Figure 6 presents the certified accuracy on MNIST and EMNIST as R is varied, when our CRFL adopts the robust aggregation algorithm RFA (Pillutla et al., 2019), which detects outliers

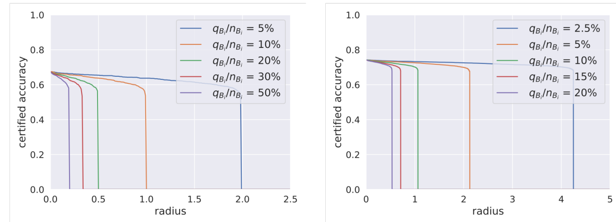


Figure 3. MNIST (left) and LOAN (right) test set certified accuracy as the poison ratio q_{B_i}/n_{B_i} is varied.

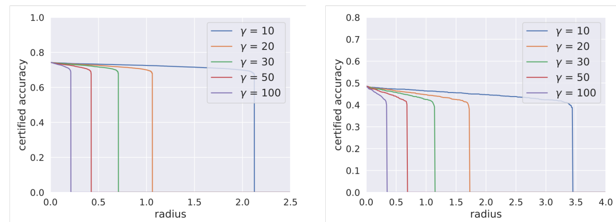


Figure 4. Certified accuracy with different scaling factor γ on LOAN (left) and EMNIST (right).

and down-weights the malicious updates during aggregation. Comparing FedAvg in Figure 5 with RFA in Figure 6 (the magnitude of x-axis is different), we observe that very large radius can be certified under RFA. This is because that the attacker is assigned with very low aggregation weights p_i, which is part of our bound in Eq. 2. Our certified radius reveals that RFA is much robust than FedAvg, which shows the potential usage of our certified radius as an evaluation metric for the robustness of other robust aggregation rules.

Effect of Client Number Distributed learning across a large number of clients is an important property of FL. Figure 7 shows that large radius can be certified when N is large (i.e., more clients can tolerant larger backdoor magnitude), because it decreases the aggregation weights p_i of attackers. Moreover, the backdoor effect could be mitigated by more benign model updates during training.

Effect of Training Rounds According to Figure 8, the certified accuracy is higher when T is larger. However, the largest radius that can be certified for the test set does not increase. We note that this is due to numerical issues of the standard Gaussian CDF $\Phi(\cdot)$. As we mentioned in Section 5.1, the continued multiplication in the denominator of Eq. 2 will not achieve 0 in practice. Otherwise the certified radius RAD goes to ∞ as $T \to \infty$ since $0 \leq 2\Phi(\rho/\sigma) - 1 \leq 1$.

To verify our argument, we fix $\underline{p_A}$ and $\overline{p_B}$ to be 0.7 and 0.1, use default values for other parameters, and study the relationship between ρ/σ, T and RAD in Figure 9(b). When ρ/σ is larger than certain threshold, the certified radius RAD does not change much when T increases. If one wishes to increase T for improving certified radius, then we suggest

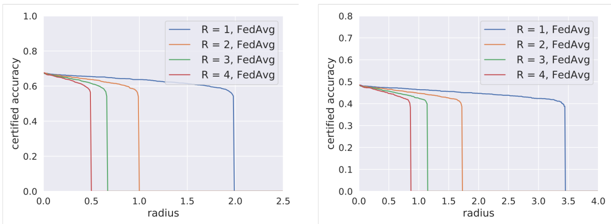


Figure 5. MNIST (left) and EMNIST (right) test set certified accuracy as the number of adversarial clients R is varied.

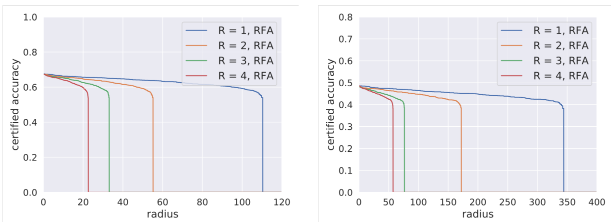


Figure 6. Certified accuracy on MNIST (left) and EMNIST (right) with different R when FL is trained under the robust aggregation RFA (Pillutla et al., 2019).

to keep ρ/σ smaller than the threshold to make effect. The increased certified accuracy when T is large in Figure 8 could be attributed to improved model performance up to convergence, so the margin between $\underline{p_A} - \overline{p_B}$ is widened.

We also study the error tolerance α and the number of noisy models M in Appendix A.2. Larger M yields larger certified radius, and the certified radius is not very sensitive to α.

Empirical Evaluation on Model Closeness Our theorems are derived based on the analysis in comparison to a "virtual" benign training process. Empirically, we train such FL global model under the benign training process and compare the ℓ_2 distance between the clean global model and the backdoored global model at every round. In Figure 9(a), one attacker performs model replacement attack on MNIST at round $\mathsf{t}_{\mathsf{adv}}= \{20, 40, 60\}$ respectively. We can observe that the plotted ℓ_2 distance over the FL training rounds after $\mathsf{t}_{\mathsf{adv}}$ is decreasing, which echos our assumption that because all clients behave normal and use their clean local datasets to purify the global model after $\mathsf{t}_{\mathsf{adv}}$, the global models between two training process become close. This observation also can justify the model closeness statement in Theorem 2.

7. Conclusion

This paper establishes the first framework (CRFL) on certifiably robust federated learning against backdoor attacks. CRFL employs model parameter clipping and perturbing during training, and uses model parameter smoothing during testing, to certify conditions under which a backdoored model will give consistent predictions with an oracle clean model. Our theoretical analysis characterizes the relation between certified robustness and federated learning parameters, which are empirically verified on three different datasets.

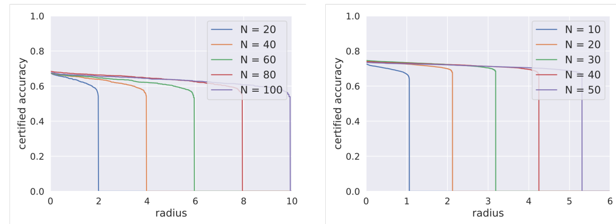


Figure 7. Certified accuracy on MNIST (left) and LOAN (right) as the number of total clients N is varied.

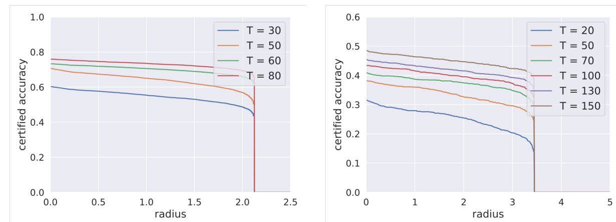


Figure 8. LOAN (left) and EMNIST (right) test set certified accuracy as T is varied.

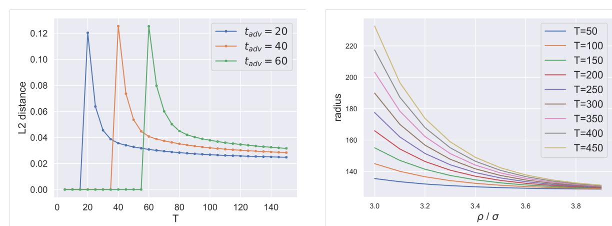


Figure 9. (a) The ℓ_2 distance of the global models between the backdoored training process and the benign training process. (b) Numerical analysis of the standard Gaussian CDF $\Phi(\cdot)$.

Acknowledgements

This work is partially supported by NSF grant No.1910100, NSF CNS 20-46726 CAR, Amazon Research Award, IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) – a research collaboration as part of the IBM AI Horizons Network.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Agarwal, N., Suresh, A. T., Yu, F., Kumar, S., and McMahan,

H. B. cpsgd: communication-efficient and differentially-private distributed sgd. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7575–7586, 2018.

Andreina, S., Marson, G. A., Möllering, H., and Karame, G. Baffle: Backdoor detection via feedback-based federated learning. *arXiv preprint arXiv:2011.02167*, 2020.

Asoodeh, S. and Calmon, F. Differentially private federated learning: An information-theoretic perspective. In *ICML Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2020.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pp. 634–643, 2019.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

Cao, X., Jia, J., and Gong, N. Z. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6885–6893, 2021.

Chen, Y., Su, L., and Xu, J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Dobrushin, R. L. Central limit theorem for nonstationary markov chains. i. *Theory of Probability & Its Applications*, 1(1):65–80, 1956.

Dvijotham, K. D., Hayes, J., Balle, B., Kolter, Z., Qin, C., György, A., Xiao, K., Gowal, S., and Kohli, P. A framework for robustness certification of smoothed classifiers using f-divergences. In *ICLR*, 2020.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

El Mhamdi, E. M., Guerraoui, R., and Rouault, S. L. A. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, number CONF, 2018.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach. *NeurIPS*, 2020.

Fu, S., Xie, C., Li, B., and Chen, Q. Attack-resistant federated learning with residual-based reweighting. *arXiv preprint arXiv:1912.11464*, 2019.

Fung, C., Yoon, C. J., and Beschastnikh, I. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, pp. 301–316, 2020.

Geyer, R. C., Klein, T., and Nabi, M. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Hoeffding, W. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pp. 409–426. Springer, 1994.

Kan, W. Lending club loan data, Mar 2019. URL https://www.kaggle.com/wendykan/lending-club-loan-data.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

Makur, A. *Information contraction and decomposition*. PhD thesis, Massachusetts Institute of Technology, 2019.

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

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017b.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Polyanskiy, Y. and Wu, Y. Dissipation of information in channels with input constraints. *IEEE Transactions on Information Theory*, 62(1):35–55, 2015.

Polyanskiy, Y. and Wu, Y. Strong data-processing inequalities for channels and bayesian networks. In *Convexity and Concentration*, pp. 211–249. Springer, 2017.

Raginsky, M. Strong data processing inequalities and *phi*-sobolev inequalities for discrete channels. *IEEE Transactions on Information Theory*, 62(6):3355–3389, 2016.

Reisizadeh, A., Farnia, F., Pedarsani, R., and Jadbabaie, A. Robust federated learning: The case of affine distribution shifts. *NeurIPS*, 2020.

Rudin, W. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976. ISBN 9780070856134. URL https://books.google.com.hk/books?id=kwqzPAAACAAJ.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 4424–4434, 2017.

Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., and Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. *NeurIPS*, 2020.

Wang, J. and Joshi, G. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. In *ICML Workshop on Coding Theory for Machine Learning*, 2019.

Weber, M., Xu, X., Karlas, B., Zhang, C., and Li, B. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018.

Zhang, J., Zheng, K., Mou, W., and Wang, L. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3922–3928, 2017.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.