Overview of Provable Robustness and Defenses against Data Poisoning Attacks: Techniques and Bounds

Ezgi Ozyilkan, Gustavo Sandoval, Chingun Khasar

Abstract

Data poisoning attacks aim to manipulate a learning-based model by distorting the training data samples in order to corrupt its test time performance. It has been shown that deep learning-based models are especially vulnerable against such attacks, whose comprehensive scrutiny is more timely than ever. On the defense side, although there has been much interest on investigating both empirical and provable robustness, provable defenses against data poisoning attacks (including backdoor attacks) remain broadly unexplored. Within the scope of this project, we will analyze provable robustness against such attacks from the lenses of a) randomized smoothing, b) partition aggregation and c) differential privacy. In particular, we will 1) provide a taxonomy for the provable robustness techniques under consideration, as well as to summarize their methodologies, 2) reveal their characteristics, strengths, limitations as well as fundamental connections they have to some other well-established techniques (e.g. randomized smoothing).

1 Motivation

As deep learning-based techniques achieve state-of-the-art performance on a wide variety of tasks such as image recognition, malware classification etc., they will be increasingly used for consequential decision making in the near future. Recently, security concerns have attracted considerable attention both from practical and theoretical research communities. It has become clear that deep neural networks (DNNs) are vulnerable against a broad range of attacks. Among these attacks, the most well-known and studied ones are *adversarial examples*, where the ill-intentioned adversary finds small perturbations to the correctly classified inputs in order to fool the DNN so that it produces an incorrect prediction during the inference phase.

Another equally important and yet understudied types of attack are *data poisoning* attacks, where the adversary manipulates dataset samples during the training phase so that the learned model becomes corrupted. A specific subset of *data poisoning* attacks, called *backdoor* attacks, are of special interest. At these attacks, the adversary inserts a *backdoor* into a DNN by adding small triggers to a subset of training instances in order to bias the trained model towards test instances with the same exact patterns. Backdoor attacks are especially stealthy since backdoored DNN models behave correctly on benign (clean) test data, making them particularly challenging to identify.

As deep learning-based models are typically highly computationally expensive to train, many engineers and businesses opt for *outsourcing* the training procedure (e.g. to the cloud computing providers). Another popular strategy to reduce the training costs is to employ *transfer learning*, where an already existing DNN is solely fine-tuned for the new task under consideration. We highlight that both of these practical outsourcing scenarios come with unique security concerns, where it should be noted that an adversary can, for example, interfere with the training procedure by corrupting the training dataset in order to create a maliciously trained network. Motivated by such a realistic attack scenario, we explore the provable robustness and defenses against data poisoning attacks within the scope of this project.

We highlight that although there have been many theoretical studies on certified defenses and robustness against test time (i.e. evasion) attacks, there has been little investigation of such for the training time (i.e. data poisoning) attacks. We hope that this project will motivate the urgent need to investigate provable robustness as well as defense techniques against data poisoning attacks in deep learning context.

2 Problem Description

For a more formal definition, let f be an arbitrary classifier that maps d-dimensional vectors (e.g. images in d-dimensional space), $x \in \mathbb{R}^d$, to C-classes. Furthermore, let the set of labels be $\mathcal{C}$ for a C-multiclass classification problem given by $\mathcal{C} = \{1.2, ..., C\}$, and let the training set be $\mathcal{D}$ consisting of (feature, label)-pairs such that $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. Formally, given clean a dataset $T \subseteq \mathcal{D}$ and a test instance x, a (reasonable) classifier's prediction $h_{prediction}(x, T)$ can be defined to be the class whose probable prediction by f is the highest, i.e.

$$h_{prediction}(x, T) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, \mathbb{P}(f(x) = c). \tag{1}$$

Similar to Weber et al. [2021], Levine and Feizi [2021], we consider the base classifier to be a deterministic function. Also following the style in Weber et al. [2021], we will occassionally omit the dependence on some parameters whenever the context is clear. In the context of *robust* classifiers $g(x)$, that are constructed from a base classifier $f(x)$ and are trained on potentially poisoned dataset U, we can define the class prediction as follows

$$j_{prediction}(x, U) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, \mathbb{P}(g(x) = c). \tag{2}$$

On the defense side, we are not only interested in the class prediction of $f(x)$ being correct, but also the robust classifier $g(x)$ outputting the same classification decision as $f(x)$ up to some poison level injected onto U, i.e.

$$h_{prediction}(x, T) = j_{prediction}(x, U) \;\; \forall U : ||T - U|| \leq \rho(x). \tag{3}$$

As shown in Fig. 1, the goal of provable robustness techniques is to guarantee that a test instance x, which may contain a poison as well as a backdoor pattern, would be predicted in the same way as the one predicted by a model trained on *clean* data (i.e. without any poisons) as long as the poisoned dataset U is within some vicinity of the original clean training set T, which is (sloppily) denoted as $||T - U|| \leq \rho(x)$ in Eq. (3).

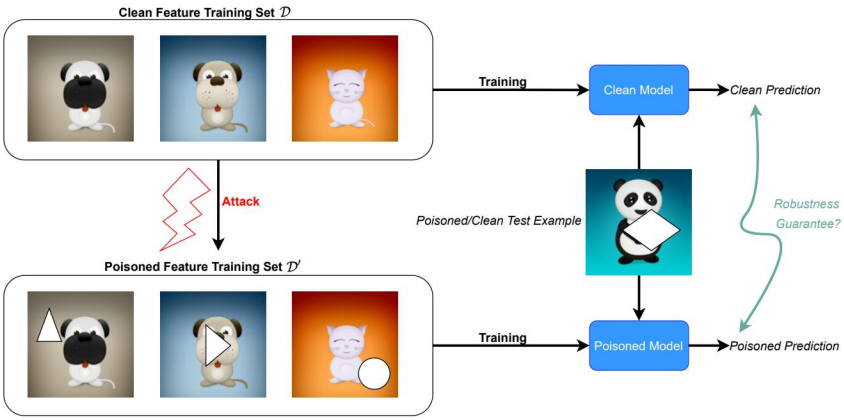

Figure 1: Illustration of the data poisoning (or backdoor) attacks problem under consideration. Figure is inspired from Weber et al. [2021].

Let's elaborate more on the threat model. Remember that we defined the clean dataset to be $T = \{(x_i, y_i)\}$. Depending on whether it is a backdoor attack and more generally, a data poisoning attack that is being considered, the attacker will have the capability to insert, for example, some

patterns δ_i as well as *poisoned* target classes $\tilde{y}_i$. For now, let's suppose that attacker *cannot* remove any samples from the training set, but only is capable of inserting some distortions onto already existing training samples. Then, without loss of generality, we can assume that the attacker can replace r training instances (x_i, y_i) by poisoned instances $(x_i + \Omega_x, \tilde{y}_i)$ which can be formalized as a poisoning attack as the transformation from $(T, \Omega_x, \tilde{y}) \mapsto U(\Omega_x, \tilde{y})$, where

$$U(\Omega_x, \tilde{y}) = \{(x_i + \delta_i, \tilde{y}_i)\}_{i=1}^{r} \cup \{(x_i, y_i)\}_{i=r+1}^{n}, \tag{4}$$

where $\Delta(\Omega_x) = \{\delta_1, \delta_2, ...\delta_r, 0, ..., 0\}$ is the collection of the poison patterns Weber et al. [2021]. Being more formal than the case in Eq. (3), we can then re-iterate the goal of defense to be *independent* of the poisoning pattern Ω_x, i.e.

$$j_{prediction}(x, U(\Omega_x, \tilde{y})) = j_{prediction}(x, U(\emptyset, y)) = h_{prediction}(x, T), \tag{5}$$

where $\mathcal{D}'(\emptyset, y)$ denotes the dataset without any embedded poissons (i.e. $\delta_i = 0 \; \forall i = 1, ..., n$).

3 Related Works

We will investigate provable defenses and robustness techniques in the literature from the following angles: randomized smoothing, partition aggregation and differential privacy.

3.1 Randomized Smoothing

Originally proposed in the influential paper Cohen et al. [2019], randomized smoothing is a method for constructing a new *smoothed* classifier $s(x)$ from a base classifier $b(x)$. When queried at x, the smoothed classifier s returns whichever class the base classifier b is most likely to return when x is perturbed by isotropic Gaussian noise, i.e.

$$s(x) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, \mathbb{P}(b(x + \epsilon) = c) \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, \, \sigma^2 I)$. Although the influential work in Cohen et al. [2019] is designated for *evasion* attacks, we can still make use of some of their results. Specifically, under some circumstances, the authors in Cohen et al. [2019] identify that they can establish a provable robustness margin (more specifically, a *radius*) of the following form

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})), \tag{7}$$

where $\underline{p_A}$ is the lower bound on the probability of top class whereas $\overline{p_B}$ is the upper bound on the probability of each other class, and where σ is referring to the standard deviation magnitude of isotropic noise of ϵ in Eq. (6). Formally, they argue that $s(x + \delta) = c_A$, for all $||\delta||_2 < R$ as long as "runner-up" class's probability is not too close to that of the top class (see Theorem 1 in Cohen et al. [2019] for the formal argument) [1].

There are few other things to note in Eq. (6) and (7). Firstly, these equations assume nothing about the base classifier b. Secondly, the certified radius gets larger when:

- the noise level σ is greater,
- the probability of the top class p_A is higher,
- the probability of the "runner-up" class p_B is low.

In Cohen et al. [2019], they use the result from Eq. (6) and (7) to train a classifier on ImageNet that is certifiably L_2-robust with provable top-1 accuracy scoring 49% on CIFAR-10.

In a sequel work by in Weber et al. [2021], they use the results from Cohen et al. [2019] to craft defenses against data poisoning, and more specifically *backdoor* attacks. Their goal is that a test instance which may contain backdoor patterns would be classified the same way, independent of whether the models where trained on data with or without embedded backdoor patterns, as long as the backdoor patterns are within an L_p-ball of radius R.

[1] Recently, in Salman et al. [2019] the certified radius indicated in Eq. (7) was proved in an alternative way, where they exploited the Lipschitz smoothness of the $\Phi^{-1}(\cdot)$ function.

In Weber et al. [2021], they aim to obtain a bound on R that such that whenever the L_2 norm of the distortion on the pixel level across the entire training dataset is below R, the class prediction of the poisoned classifier would be the same as the classifier trained on completely benign (clean) data, i.e.

$$\sqrt{\sum_{i=1}^{r} \|\delta_i\|_2^2} < R, \tag{8}$$

where δ_i is referring to a poison pattern on some image, and where the attacker incurred r different poisons onto the clean dataset (see Eq. (4)).

The provable robustness against poisoning attacks in Weber et al. [2021] consists of adding noise samples from a smoothing distribution to the given training instances in order to ultimately obtain a collection of of "smoothed" version of the training dataset. They train a unique classifier model on each smoothed version of the training dataset and aggregate the final output together as the final smoothed prediction. Evoking Neyman Pearson lemma, they derive a robustness condition for this smoothed training process (see Theorem 1 in Weber et al. [2021]), whose proof of concept is similar to the that of in Cohen et al. [2019]. Let's stick with the notation employed in Weber et al. [2021], and look at their approach. More specifically, we will be outlining the main proof (i.e. Theorem 1) in Weber et al. [2021].

3.1.1 Theorem 1 in Weber et al. [2021]

Preliminaries. In Weber et al. [2021], they propose a unified theoretical framework for certified robustness against evasion and data poisoning attacks by randomizing the prediction, more specifically by *smoothing* the final prediction, via employing the technique in Cohen et al. [2019].

In a nutshell, they try to obtain a smoothed classifier g from a base classifier f by introducing additive Gaussian noise to the input (similar to Eq. (6)). The main reasoning behind is that such smoothing operation would reduce the occurences of regions with high curvature, which would render the classifier to be less vulnerable against attacks[2].

Slightly modifying the notation introduced in Eq. (1), we can argue that the base classifier can be defined as $h(x, T) = \operatorname{argmax}_c p(c \mid x, T)$, where the conditional probability distribution p over classes c is learned from some dataset T. In Weber et al. [2021], they define a *smoothed* classifier as

$$q(c \mid x, T) = \mathbb{P}_{X,D}(h(x + X, T + D) = c), \tag{9}$$

where they specifically introduce random variables $X \sim \mathbb{P}_X$ and $D \sim \mathbb{P}_D$, which act as *smoothing* distributions and are assumed to be independent. One can think of D as a collection of n independent and identically distributed (i.i.d.) random variables $D^{(i)}$ that are added onto T. Revising the previous definition established in Eq. (2), one can define the smoothed classifier on some test instance x as follows

$$j(x, T) = \operatorname*{argmax}_c q(c \mid x, T). \tag{10}$$

Observe that as seen in Eq. (9), the smoothed classifier g depends on the choice of smoothing distributions $\mathbb{P}_X$ and $\mathbb{P}_D$.

Statistical hypothesis testing. Hypothesis testing is a standard problem in statistics that is concerned with deciding between two alternative explanations for the data observed Cover and Thomas [2006]. In the simplest setup, the problem boils down to deciding between two i.i.d. distributions. Formally, the decision is based on some realization x from a random variable X, whose distribution is either $\mathbb{P}_0$ (also called the null hypothesis) or $\mathbb{P}_1$ (also called the alternative hypothesis). Given a sample $x \in \mathcal{X}$, some randomized test ϕ can be formally represented as a function $\phi : \mathcal{X} \to [0, 1]$ which rejects the null hypothesis or accepts it with probability $\phi(x)$ and $1 - \phi(x)$, respectively.

We can define two probabilities of error: type I error, denoted as $\alpha(\phi)$, and type II error, denoted as $\beta(\phi)$. Formally, we can define them as:

$$\alpha(\phi) = \mathbb{E}_{x \sim \mathbb{P}_0}[\phi(X)], \quad \beta(\phi) = \mathbb{E}_{x \sim \mathbb{P}_1}[1 - \phi(X)]. \tag{11}$$

[2]This was also the main argument for employing randomized smoothing in the seminal work of Cohen et al. [2019].

The binary version of the hypothesis testing problem then reduces to selecting the test ϕ^* which minimizes the probability of type II error subject to $\alpha(\phi^*) \leq \alpha_0$. The Neyman Pearson lemma states that a likelihood ratio test is an optimal candidate for such ϕ^*, i.e. $\alpha(\phi_{NP}) = \alpha_0$ and where

$$\beta(\phi_{NP}) = \inf_{\phi:\alpha(\phi)\leq\alpha_0} \beta(\phi). \tag{12}$$

The likelihood ratio tests which decides whether a sample x originates from a distribution $\mathbb{P}_0$ or $\mathbb{P}_1$ can be formally defined as

$$\phi(x) = \begin{cases} 1 \text{ if } \Lambda(x) > t \\ q \text{ if } \Lambda(x) = t, \\ 0 \text{ if } \Lambda(x) < t. \end{cases}, \text{ with } \Lambda(x) = \frac{f_{X_1}(x)}{f_{X_0}(x)}, \tag{13}$$

where f_{X_0} and f_{X_1} are probability densities of with respect to a measure μ, and where the values q and t are chosen to satisfy that the test ϕ has *significance level* α_0, i.e. $\alpha(\phi) = \mathbb{P}_0(\Lambda(X) > t) + q \cdot \mathbb{P}_0(\Lambda(X) = t) = \alpha_0$.

Lemma 1. *Let X_0 and X_1 be random variables with probability density functions f_1 and f_0 with respect to a measure μ. Furthermore, let ϕ^* be some likelihood ratio test for testing the null hypothesis X_0 against the alternative one X_1. Then, for any other likelihood ratio test ϕ, the following conditions would hold:*
i) $\alpha(\phi) \geq 1 - \alpha(\phi^*) \implies 1 - \beta(\phi) \geq \beta(\phi^*)$
ii) $\alpha(\phi) \leq \alpha(\phi^*) \implies \beta(\phi) \geq \beta(\phi^*)$.

Proof. See Cover and Thomas [2006].

Theorem 1 in Weber et al. [2021]. *Let q be the smoothed classifier defined as in Eq. (9) with smoothing distribution $Z = (X, D)$, where $X \in \mathbb{R}^d$ and D being a collection of i.i.d. $\mathbb{R}^d$-valued random variables. Let $\delta \in \mathbb{R}^d$ be backdoor patterns, and let $\Delta = (\delta_1, \delta_2, .., \delta_n)$ be the collection of the backdoor patterns (defined as in Section 2). Let $c_A \in \mathcal{C}$ and let $p_A, p_B \in [0, 1]$ be the probability of top class and of the runner-up one, respectively, such that $c_A = g(x, T)$ and*

$$q(c_A \mid x, T) \geq p_A > p_B \geq \max_{c \neq c_A} q(c \mid x, T). \tag{14}$$

If the optimal type II errors, for testing the null hypothesis $Z \sim \mathbb{P}_0$ against the alternative one $Z + (\delta, \Delta) \sim \mathbb{P}_1$ satisfy the following condition

$$\beta^*(1 - p_A) + \beta^*(p_B) > 1, \tag{15}$$

then it is guaranteed that $c_A = \operatorname{argmax}_c q(c \mid x + \delta, T + \Delta)$.

Proof sketch. Firstly, we will explicitly construct likelihood ratio tests ϕ_A and ϕ_B as defined in Eq. (13) for testing the null hypothesis against the alternative one, scoring type I errors of $\alpha(\phi_A) = 1 - p_A$ and $\alpha(\phi_B) = p_B$, respectively. A line of reasoning similar to Neyman-Pearson lemma can then be used to show that the class probability for c_A given by conditional distribution q on the perturbed input is lower bounded by $\beta(\phi_A) = \beta^*(1 - p_A)$. A similar argument also leads to the conclusion that the prediction score for $c \neq c_A$ on the perturbed input is upper bounded by $1 - \beta(\phi_B) = 1 - \beta^*(\phi_B)$. Combining these two observations, one can conclude Eq. (15).

Proof. For some $p \in [0, 1]$, let ϕ_p be a likelihood ratio test as defined in Eq. (13), having *significance level* of $1 - p$, i.e. $\alpha(\phi) = 1 - p$. Set some q and t values to match this significance level of $1 - p$. Therefore, we can argue that the likelihood ratio test $\phi_A \equiv \phi_{p_A}$ then satisfies $\alpha(\phi_A) = 1 - p_A$.

As per the assumption in Eq. (14) (and as per the original definition of the smoothed classifier provided in Eq. (9)), we have $\mathbb{E}(p(c_A \mid x + X, T + D)) = q(c_A \mid x, T) \geq 1 - \alpha(\phi_A)$. Applying the first remark in Lemma 1 by setting $\phi \equiv p(c_A \mid x + X, T + D)$ and $\phi^* \equiv \phi_A$ and using the same line of argument as in Neyman-Pearson lemma, one can argue that

$$q(c_A \mid x + \delta, T + \Delta) = 1 - \beta(\phi) \geq \beta(\phi_A). \tag{16}$$

Similarly, let the likelihood ratio test be $\phi_B \equiv \phi_{1-p_B}$ satisfying the significance level of $\alpha(\phi_B) = p_B$. Following the assumption in Eq. (14), for $c \neq c_A$ we have $\mathbb{E}(p(c \mid x + X, T + D)) = q(c \mid x, T) \leq p_B \leq \alpha(\phi_B)$. Symmetrically, applying the second remark in Lemma 1 by setting $\phi \equiv p(c \mid x + X, T + D)$ and $\phi^* \equiv \phi_B$, we can conclude that

$$q(c \mid x + \delta, T + \Delta) = 1 - \beta(\phi) \leq \beta(\phi_B). \tag{17}$$

Combining the relationships in Eq. (16) and (17), one can observe that if $\beta(\phi_A) + \beta(\phi_B) > 1$, then it is guaranteed to satisfy the following condition, which completes the proof,

$$q(c_A \mid x + \delta, T + \Delta) > \max_{c \neq c_A} q(c \mid x + \delta, T + \Delta). \tag{18}$$

Using Theorem 1 in Weber et al. [2021] and the alternative proof of the robustness guarantee of the radius R in Cohen et al. [2019], which is provided in Salman et al. [2019] that makes use of Lipschitz smoothness of the $\Phi^{-1}(\cdot)$ function, one can easily show the following corollary, which constitutes the main result as the provable robustness against backdoor attacks in Weber et al. [2021].

Corollary 1. *Let $\Delta = (\delta_1, \delta_2, ..., \delta_n)$, where $\delta_j \in R^d$ some backdoor pattern and let T be the training dataset. For each i, let the smoothing noise on the training features be i.i.d. Gaussian one, i.e. $D^{(i)} \sim^{i.i.d.} \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$. Let $c_A \in \mathcal{C}$ such that $c_A = g(x + \delta, T + \Delta)$, where class predictions satisfy the following condition,*

$$q(c_A \mid x + \delta, T + \Delta) \geq p_A > p_B \geq \max_{c \neq c_A} q(c \mid x + \delta, T + \Delta). \tag{19}$$

If the backdoor patterns δ_i satisfy the condition

$$\sqrt{\sum_{i=1}^{r} \|\delta_i\|_2^2} < \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)), \tag{20}$$

it is then guaranteed that $q(c_A \mid x + \delta, T + \Delta) > \max_{c \neq c_A} q(c \mid x + \delta, T + \Delta)$.

Remarks. Observe that the robustness margin in Cohen et al. [2019], provided in Eq. (7), matches the right-hand side expression in Eq. (20). This result also shows that given that the norms of the backdoor patterns are below some certain threshold, one can guarantee that the robust classifier will output the same class prediction with the classifier that is trained on the dataset without any embedded patterns.

3.1.2 Training and Inference

Training. As briefly detailed in Section 3.1, we draw N i.i.d. samples $d_1, d_2, ..., d_n$ from the smoothing distribution $D \sim \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ for some σ value chosen. Given these N samples of training noise, we consequently train N DNN models on the smoothened versions of the datasets $T + d_k$ for $k = 1, ..., N$ in order to obtain classifiers $h_1, ..., h_N$. To reduce the mismatch between training and test distributions, a random noise u_k is also drawn from $\mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ with a random seed based on the hash value of the trained model file. This random noise vector value is stored as it will be used during inference time.

Inference. To get the prediction of the smoothed classifier on a test instance x, the empirical majority vote in the form of an unbiased estimate of the class probabilities is computed as follows

$$\hat{q}(c \mid x, T) = \frac{|\{k : h_k(x + u_k, T + d_k) = c\}|}{N}, \tag{21}$$

where u_k is the deterministic noise vector sampled during training. Next, p_A and p_B values are computed for a given tolerance (significance) level α. Invoking Corollary 1, one can compute the robust radius according to Eq. (20) based on p_A, p_B, the smoothing noise standard deviation parameter σ and the number of poisoned training sample amount r. Should the resulting radius R be larger than the magnitude of the backdoor samples δ_i, we can conclude that the class prediction is *certified*, meaning that the backdoor attack has failed to succeed on this particular test instance x.

3.2 Partition Aggregation

Deep Partition Aggregation (DPA) is proposed by Levine and Feizi Levine and Feizi [2021]. This method leads to robustness certifications against *general* poisoning attacks (i.e. insertion or deletion of a bounded number of training samples). Authors in Levine and Feizi [2021] claim that their results are the current state-of-the-art provable defenses against general data poisoning attacks.

Firstly, let's define some notation related to this work. $A \ominus B$ represents the set symmetric difference, i.e. $A \ominus B = (A \setminus B) \cup (B \setminus A)$. The number of elements in A is denoted as $|A|$. $[n]$ is the set of

integers 1 through n, and $\lfloor z \rfloor$ is the largest integer less than or equal to z. $\mathbb{1}$ represents the indicator function, i.e. $\mathbb{1}_{Prop}$ = 1 if Prop is true; $\mathbb{1}_{Prop}$ = 0 otherwise.

Let $\mathcal{S}$ be the space of all possible unlabeled samples. Assuming that we can sort elements of $\mathcal{S}$ in a deterministic and unambiguous way, we denote lexicographical sorted version of $\mathcal{S}$ as $\mathcal{S}_L$. We represent labels with integers; therefore, the set of all possible labeled samples is:

$$\mathcal{S}_L = \{(x, c) \mid x \in S, c \in \mathbb{N}\}. \tag{22}$$

A training set for a classifier can then be denoted as $T \in \mathcal{P}(\mathcal{S}_\mathcal{L})$ where $\mathcal{P}(\mathcal{S}_L)$ corresponds to the power set of $\mathcal{S}_L$. Similar to the previous work, a classifier can be defined to be a function from the training set *and* the sample to a label, i.e. $f : \mathcal{P}(\mathcal{S}_L) \times \mathcal{S} \to \mathbb{N}$. Being consistent with the earlier notation, we note $f(\cdot)$ and $g(\cdot)$ to be the base and robust classifiers, respectively.

Training. DPA divides the the training set T into "k" partitions, where a hash function "h" determines the partition assignment of a training sample. Defining partitions as $P_1, ..., P_K \subseteq T$ from the training set, we have

$$P_i := \{t \in T \mid h(t) \equiv i \ (mod\ k)\}, \tag{23}$$

where $k \in \mathbb{N}$ is a hyperparameter denoting the number of base classifiers to be used in the ensemble. There is no particular restriction on hash function h as long as it is a deterministic function to form some partitions of similar sizes, and it can be defined as $h : \mathcal{S}_L \to \mathbb{N}$. A separate (unique) base classifier is trained on each partition,

$$f_i(x) := f(P_i, x), \tag{24}$$

and we can define the trained base classifiers as $f_i : \mathcal{S} \to \mathbb{N}$. Since the hash value depends only on the value of the training sample, neither poisoning other samples, nor changing the total number of samples, nor reordering the samples can change which the partition t is assigned to. The key insight is that removing a training sample or adding a new sample will only change the contents of *one* partition and therefore, will only affect the classification of one of the k base classifiers.

Inference. Given a test instance x, we evaluate it on each base classifier f_i, and we count the number of classifiers that return each unique class c,

$$n_c(x) := \mid \{i \in [k] \mid f_i(\mathbf{x}) = c\} \mid. \tag{25}$$

This lets us define the robust classifier g_{dpa} which returns the consensus output of all the base classifiers (i.e. the ensemble)

$$g_{dpa}(T, x) := \underset{c}{argmax}\ n_c(x). \tag{26}$$

Note that when taking the argmax in Eq. (26), we break ties *deterministically* by returning the smaller class index[3]. The resulting classifier g_{dpa} leads to some robustness guarantees against general poisoning attacks, which is the main theorem provided in Levine and Feizi [2021].

Theorem 2 (i.e. Theorem 1 in Levine and Feizi [2021]). *For a fixed deterministic base classifier f, hash function h, ensemble size k, training set T, and a test instance x, let*

$$c := g_{dpa}(T, x), \tag{27}$$

$$\rho(x) := \left\lfloor \frac{n_c - max_{\ c' \neq c} \left(n_{c'}(x) + \mathbb{1}_{c' < c}\right)}{2} \right\rfloor. \tag{28}$$

Then, for any poisoned (in the general sense) training set U, if $|T \ominus U| \leq \rho(x)$, then it is guaranteed that $g_{dpa}(U, x) = c$.

As seen in Eq. (28), given that the *poison amount* inserted onto dataset T is lower than $\rho(x)$, g_{dpa} is then guaranteed to classify the test instance x correctly although it is trained on the poisoned dataset U.

Proof. Let's define the partitions, the trained classifiers, the class counts both for the clean dataset T as well as for the poisoned one U. Being consistent with the introduced notation in Section 3.2, we will be using k partitions and the deterministic hash function h.

$$P_i^T := \{t \in T \ \mid \ h(t) \equiv i \ (mod\ k)\}, \tag{29}$$

$$P_i^U := \{t \in T \ \mid \ h(t) \equiv i \ (mod\ k)\}, \tag{30}$$

[3]Breaking the ties *deterministically* will play a key role in the proof of Theorem 2.

$$f_i^T := f(P_i^T,\ x), \tag{31}$$
$$f_i^U := f(P_i^U,\ x), \tag{32}$$

$$n_c^T := \{i \in [k] \mid\ f(P_i^T, x) = c\}, \tag{33}$$
$$n_c^U := \{i \in [k] \mid\ f(P_i^U, x) = c\}, \tag{34}$$

$$g_{dpa}(T, x) := \underset{c}{argmax}\ n_c^T(x), \tag{35}$$
$$g_{dpa}(U, x) := \underset{c}{argmax}\ n_c^U(x). \tag{36}$$

Note that we are using superscripts to distinguish dependencies to the clean training set T and to the poisoned dataset U. When computing the argmax values in Eq. (35) and (36), should there be a tie, we break it *deterministically* by returning the class index having the smaller integer value.

Few observations are in order. Note that we have $P_i^T = P_i^U$ as long as there is no t belonging to $T \ominus U$ such that $h(t) \equiv i\ (mod\ k)$. Remembering that the hash function h is a deterministic one and also noticing the fact that the number of partitions i for which we would have $P_i^T \neq P_i^U$ can be at most $|T \ominus U|$, implying that $f_i^T(x) = f_i^U(x)$, we can conclude that the number of classifiers for which we would have $f_i^T(x) \neq f_i^U(x)$ is at most $\rho(x)$. Looking at Eq. (33) and (34), we can conclude that this remark formally corresponds to

$$\forall c' :\ |n_{c'}^T - n_{c'}^U| \leq \rho(x). \tag{37}$$

Next, let $c := g_{dpa}(T, x)$. We would also have $g_{dpa}(U, x) = c$ if the following conditions are satisfied

$$\forall c' < c\ : n_c^U(x) > n_{c'}^U, \tag{38}$$
$$\forall c' > c\ : n_c^U(x) \geq n_{c'}^U, \tag{39}$$

where the separate cases stem from the nature of Eq. (35) and (36), where we deterministically choose the smaller indices in cases of ties in argmax computations. Eq. (38) and (39) can be succintly written as $\forall c' \neq c\ : n_c^U(x) \geq n_{c'}^U(x) + \mathbb{1}_{c'<c}$. Evoking triangle inequality, in combination with the relationship established in Eq. (37), we can conclude that we would have $g_{dpa}(U, x) = c$ if $\forall c' \neq c\ : n_c^T(x) \geq n_{c'}^T(x) + 2\rho(x) + \mathbb{1}_{c'<c}$. Arranging this equivalent expression, we can get an upperbound on the maximum symmetric difference amount $\rho(x)$ that DPA algorithm can sustain, i.e. $g_{dpa}(U, x) = g_{dpa}(T, x)$, which completes the proof.

3.3 Differential Privacy

Although differential privacy (DP) was primarily established for protecting users' privacy Dwork et al. [2006], it has been shown that differentially-private learners might have some ingrained protection against data poisoning attacks. This mainly stems from the fact that a DP algorithm should satisfy the requirement that no adversary can learn much about a single data point in the database. DP ensures this by introducing randomness in a way that changing a single entry cannot change the output distribution of the database very much (see classical definition of (ϵ, δ)-DP). The connection between DP and robustness was formally established by Lecuyer et al. [2019] for the first time, where the authors propose certified defenses against *evasion* attacks. Recently, Ma et al. [2019] showed the theoretical bounds on how much the adversary can change the distribution by introducing a fixed number of poisoned entries into the training set. Later, Hong et al. [2020] used DP-SGD, as a means of *gradient shaping*, to boost the model's robustness against data poisoning attacks. Although later Jagielski and Oprea [2021][4], argued DP cannot be directly considered as a defense against poisoning attacks, the intuitive ramifications of DP for training time attacks need deeper investigation.

[4]In this work, by appealing for a more thorough quantification to asses the defenses, they introduce multiple metrics that a defense against data poisoning should satisfy. They empirically show that DP algorithms do not necessarily perform well across all metrics and therefore, argue that the core component of defense mechanism of DP originates from *robust training* algorithms, rather than formal *privacy* properties of DP.

4 Experimental Results

In this section, we present extensive experimental evaluation of the approaches discussed in detail in Section 3. In particular, we evaluate a distinct type of DNN on CIFAR-10 dataset. To compare different approaches of Weber et al. [2021] and Levine and Feizi [2021], we consider a unique type of a backdoor attack (see Fig. 2).

4.1 Experiment Setup

Using CIFAR-10 dataset consisting of coloured images of size 32×32, we opted for train and test split of $50,000 : 10,000$. The dataset consists of 10 classes, where each class has 6000 images. For both approaches of Weber et al. [2021], Levine and Feizi [2021], we follow the backdoor attack setting in Schwarzschild et al. [2020]. We train a DNN model to classify $(image, class)$ pairs on each sample from the dataset.

In our experiments, we vary the amount of poison that the attacker can inject onto the training dataset. Specifically, we consider poisoning amount of 2%, 5% and 10% of the training set (see Section 4.2). For all experimental evaluations, we opt for four-pixel backdoor pattern, where the poisoning location on the image is pre-determined and fixed across all the backdoored images (see Fig. 2).

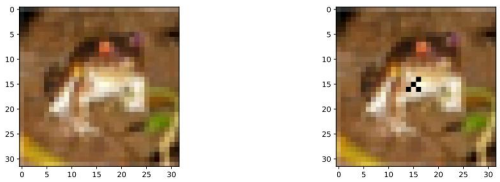

Figure 2: A comparison of an original CIFAR-10 image (left) and its backdoored version (right).

To benchmark the performance of the approaches of RAB and DPA, we choose the ResNet architecture used in Cohen et al. [2019] for both. For experimental evaluations, we use the pretrained version of the model, and consequently, fine tune it with a subset of training data containing backdoored samples. We fix the number of sampled noise vectors to be $N \in \{100, 1000\}$, leading to an ensemble of $\{100, 1000\}$ classifiers (see Eq. (21)). The added smoothing noise is obtained from the Gaussian distribution with $\mu = 0$ and $\sigma \in \{0.5, 1.0, 2.0\}$. We also set the significance level used in obtaining p_A and p_B as $\alpha = 0.001$.

For experimenting with the approach proposed in Levine and Feizi [2021], we again use the same attack style illustrated in Fig. 2 on CIFAR-10 dataset. We only consider the vanilla DPA algorithm proposed in Levine and Feizi [2021]. As discussed in Section 3.2, one needs to carefully consider the hyperparameter k for this robustness method. Note that this hyperparameter k is equivalent to the number of classifiers in the ensemble. Since each sample of the training set is seen by exactly one of the k classifiers, one can easily see that the number of samples used in the training of each of the classifier k is therefore *inversely* proportional to k. Thus, there is an ingrained trade-off between robustness and accuracy in Levine and Feizi [2021]: the higher k value is, the larger certified robustness DPA can provide as well as the smaller number of training samples each of the k classifiers gets to use.

4.2 Empirical Results

First, let's discuss the experiments related to Weber et al. [2021]. As seen in Fig. 3, one can observe that the certified accuracy on the non-poisoned test data deteriorates as the ratio of the poison in the training set increases (i.e. the x-axis values) across all σ values considered. Although higher σ value would mean a larger certifiable robust margin as seen in Eq. (20), one can observe that the median of the certified accuracy actually drops as σ ranges from 0.5 to 2.0. We argue that this is because higher σ value corresponds to higher variance in the sampled smoothing noise, which renders the classifiers to be less accurate although the interquartile range of N unique classifiers becomes smaller (see Fig. 3).

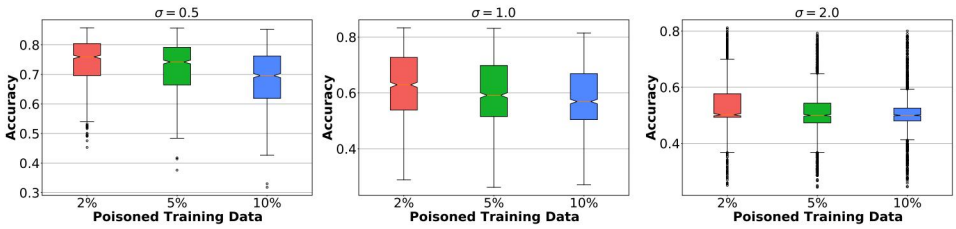

Figure 3: Certified accuracy performance on the *non-poisoned* test samples as a function of different ratios of poisoned training data on CIFAR-10 dataset employing the attack style illustrated in 2. We consider $\sigma \in \{0.5, 1.0, 2.0\}$ while keeping $N = 1000$ (see Eq. (20)).

The performance of certified accuracy on *poisoned* test samples is provided in Fig. 4. It is interesting to observe that as the ratio of poisoned training data increases, the certified accuracy *does* increase across all σ values chosen. We argue that this is because of the peculiar choice of the random noise samples u_k, which are using the same random seed based on the hash value of the training noise samples d_k in order to diminish the mismatch between training and test distributions, as explained in Section 3.1.2. This observation hints at the possibility that the DNN under consideration can actually be simply *memorizing* the backdoor pattern across poisoned training samples, which would explain why it is scoring higher certified accuracy value as the ratio of poisoned training data increases across all σ values investigated. The pattern that emerges from Fig. 3 and 4 related to the performance on the *non-poisoned* and *poisoned* test samples is also observed when we set $N = 100$ (see Fig. 5).

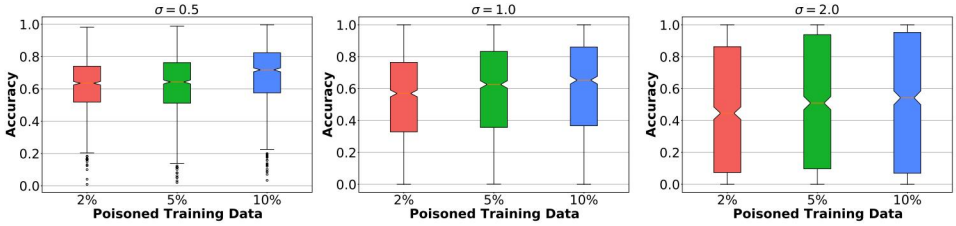

Figure 4: Certified accuracy performance on the *poisoned* test samples as a function of different ratios of poisoned training data on CIFAR-10 dataset employing the attack style illustrated in 2. We consider $\sigma \in \{0.5, 1.0, 2.0\}$ while keeping $N = 1000$ in RAB approach (see Eq. (20)).

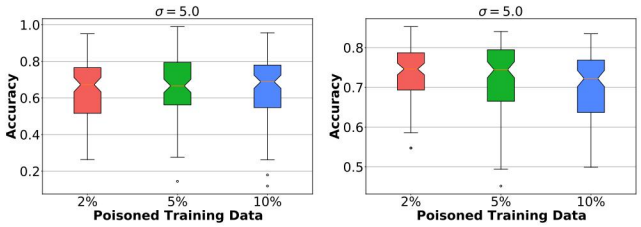

Figure 5: Certified accuracy performance on the *non-poisoned* (left) and *poisoned* (right) test samples as a function of different ratios of poisoned training data on CIFAR-10 dataset employing the attack style illustrated in 2. We only consider $\sigma = 5$ while keeping $N = 100$ in RAB approach (see Eq. (20)).

In Abadi et al. [2016a], the authors propose an algorithm to implement (ϵ, δ)-DP within the context of deep neural networks. Specifically, their implementation is based on training the neural networks under a modest privacy budget that is argued to be particularly efficient (see Algorithm 1). Observe that their approach is notably similar to DP-SGD Hong et al. [2020], where they provide privacy in

the form of *gradient shaping*. By varying the parameters clipping norm C and noise multiplier σ, one can obtain different budgets of privacy loss of ϵ. In our experiments, we fix $C = 1$ while we vary the noise multiplier as $\sigma \in \{0.5, 2.0, 5\}$. One can easily remark that by increasing the σ value, we obtain more *privacy* budget, corresponding to smaller ϵ value in the original definition of ϵ-DP. As another extension of the RAB method, after training *smooth* differentially private DNNs using Hong et al. [2020], we evaluate RAB approach on them. As seen in Fig. 6, we do not observe an improvement in certified accuracy metric as the DNN becomes more differentially private (corresponding to higher σ value). We can therefore argue that differential privacy in terms of gradient shaping by itself is not enough for further improving the robustness in terms of enhancing the certified accuracy. This observation is consistent with the one provided in Weber et al. [2021].

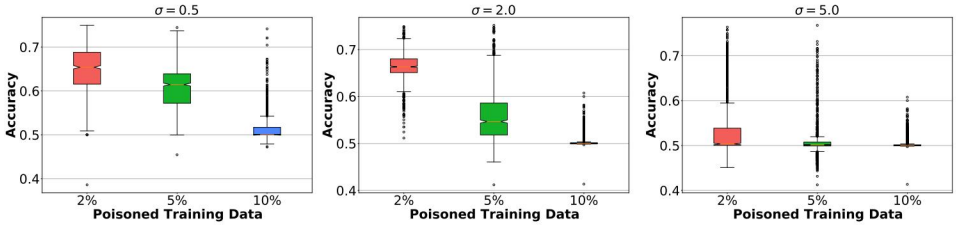

Figure 6: Certified accuracy performance on the *non-poisoned* test samples as a function of different ratios of poisoned training data on CIFAR-10 dataset employing the attack style illustrated in 2. We consider various noise multiplier such as $\sigma \in \{0.5, 1.0, 2.0\}$ while keeping $C = 1$ (see Algorithm 1). We also fix $\sigma = 0.5$ $N = 1000$ in terms of RAB parameters (see Eq. (20)).

Empirical evaluation of DPA method is provided in Fig. 7. Looking at the left image in Fig. 7, we can immediately observe the trade-off between robustness and accuracy concerning the hyperparameter k (see Section 3.2 for the related discussion). We observe that although higher value of k provides a larger margin of robustness as the ratio of poisoned training data gets larger (i.e. the x-axis), it also leads to less number of training data seen by each unique classifier, which explains why the certified accuracy values observed for $k = 50$ and $k = 250$ are remarkably low.

We also compare RAB and DPA methods on the same exact poisoning attack in Fig. 7 (right). We can observe that RAB significantly outperforms DPA under this particular poisoning attack style. We argue that this is because of the particular choice of N value for the RAB approach (see Eq. (20)), which renders the failure probability of the randomized certificate of RAB to be significantly low. Although it is claimed that DPA is the current state-of-the-art approach for *general* poisoning attacks, we suggest that it is prone to be particularly vulnerable against backdoor attacks as empirically observed in Fig. 7.

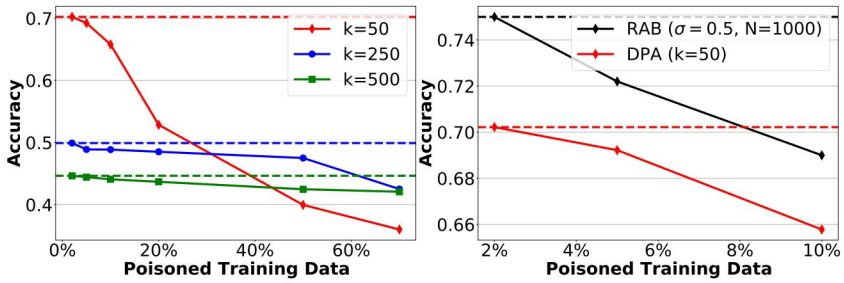

Figure 7: Certified accuracy performance on the *non-poisoned* test samples as a function of different ratios of poisoned training data on CIFAR-10 dataset employing the attack style illustrated in Fig. 2 (left). We vary the parameter k in DPA algorithm such that $k \in \{50, 250, 500\}$ (see Section 3.2). Certified accuracy performance on the *non-poisoned* test samples for the indicated RAB and DPA methods as a function of different ratios of the same poisoning attack (right).

5 Discussion and Conclusion

Limitations. Let's analyze Weber et al. [2021] first. Looking at Corollary 1, one can observe that this method can only indeed certify for *clean-label* attacks (i.e. where only the existing training samples are modified, not their corresponding labels). Furthermore, scrutinizing Corollary 1, one can also realize that the robustness guarantee it provides is bounded to L_2 norm of the distortion calculated across *all pixel values* in the entire training set. This turns out to be yielding certificates to only very small range of distortions of the training data. Moreover, the failure probability of this proposed randomized certificate goes to zero only at the limit where the number of trained classifiers, i.e. N in Eq. (21), goes to infinity. Finally, each of the N classifiers is trained on a specific noisy version of the *entire* training dataset, which renders RAB model to be computationally expensive compared to DPA.

Considering the DPA approach proposed in Levine and Feizi [2021], one can notice that unlike most of the previous certified robustness methods (e.g. Cohen et al. [2019]), it provides *deterministic* certificates against poisoning attacks. This hints at the possibility that it will potentially be vulnerable to some malicious *backdoor* attacks, where the adversary could easily exploit the non-stochastic aspect of this method.

As discussed in Section 4.2, leveraging differential privacy as a supplementary defense mechanism against poisoning attack does not seem to improve the certified accuracy on the non-poisoned test samples. This validates the claim of Jagielski and Oprea [2021], that further investigation is needed to consider whether DP is indeed an effective instrument for training time attacks.

Conclusion. In this project, we compare and contrast three approaches of robustness mechanisms against data poisoning attacks: randomized smoothing, partition aggregation and differential privacy (in terms of *gradient shaping*). For provable robustness, we provide the outline of the proofs of the main theorems in Weber et al. [2021] and Levine and Feizi [2021]. We evaluate and analyze these certified robustness approaches on a realistic backdoor attack setup. We identify the strengths and limitations of all methods considered. We conclude that there is plenty of important work left to be done in terms of designing provable robustness defenses against general data poisoning as well as backdoor attacks. The approaches discussed in this report represent a promising first step towards a more secure outsourced training methodology for deep neural networks, whose practical importance is indeed very timely. One straightforward future direction would to be to unify both methodologies of RAB and DPA (i.e. randomized smoothing and partition aggregation) and to prove the certified robustness margin achieved as such.

6 Appendix

This is the algorithm proposed by Abadi et al. [2016b] that is used for the DP implementation within the context of deep neural networks.

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, ..., x_N\}$, loss function $L(\theta) = \frac{1}{N}\sum_i L(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.
  Initialize θ_0 randomly
  for $t \in [T]$ **do**
    Take a random sample L_t with sampling probability L/N
    Compute gradient
    for all $i \in L_t$ **do**
      compute $g_t(x_i) \leftarrow \nabla L(\theta_t, x_i)$
      Clip gradient
      $\hat{g}_t \leftarrow g_t(x_i)/max(1, \frac{\|g_t(x_i)\|_2}{C})$
      Add noise
      $\hat{g}_t \leftarrow \frac{1}{L}(\sum_i \hat{g}_t(x_i) + N(0, \sigma^2 C^2 \mathbf{I}))$
      Descent
      $\theta_{t+1} \leftarrow \theta_t - \eta_t \hat{g}_t$
Output: θ_T and compute the overall privacy cost (ϵ, δ) using a privacy accounting method.

References

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*. ACM, oct 2016a. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145%2F2976749.2978318.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy, oct 2016b. URL https://doi.org/10.11452F2976749.2978318.

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping, 2020.

Matthew Jagielski and Aliana Oprea. Does differential privacy defeat data poisoning?, 2021.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy, 2019.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks, 2021.

Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses, 2019.

H. Salman, G. Yang, J. Li, P. Zhang, H. Zhang, I. Razenshteyn, and S. Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers, 2019.

Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks, 2020. URL https://arxiv.org/abs/2006.12557.

Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks, 2021.

1 See project report description for list of expected contents **30 / 30**

✓ **- 0 pts** **Correct**