Provably Robust Defenses against Data Poisoning Attacks via Ensemble Methods

Jinyuan Jia
Duke University
jinyuan.jia@duke.edu

Xiaoyu Cao
Duke University
xiaoyu.cao@duke.edu

Neil Zhenqiang Gong
Duke University
neil.gong@duke.edu

Abstract

In a *data poisoning attack*, an attacker modifies, deletes, and/or inserts some training examples to corrupt the learnt machine learning model. *Bootstrap Aggregating (bagging)* is a well-known ensemble learning method, which trains multiple base models on random subsamples of a training dataset using a base learning algorithm and uses majority vote to predict labels of testing examples. We prove the intrinsic certified robustness of bagging against data poisoning attacks. Specifically, we show that bagging with an arbitrary base learning algorithm provably predicts the same label for a testing example when the number of modified, deleted, and/or inserted training examples is bounded by a threshold. Moreover, we show that our derived threshold is tight if no assumptions on the base learning algorithm are made. We evaluate our method on MNIST and CIFAR10. For instance, our method achieves a certified accuracy of 91.1% on MNIST when arbitrarily modifying, deleting, and/or inserting 100 training examples.

1 Introduction

Data poisoning attacks aim to carefully poison (i.e., modify, delete, and/or insert) some training examples such that the corrupted model makes incorrect predictions for testing examples as an attacker desires. To mitigate data poisoning attacks, several *certified defenses* [6, 7] were recently proposed. We say a learning algorithm is certifiably robust against data poisoning attacks if it can learn a classifier that provably predicts the same label for a testing example when the number of poisoned training examples is bounded. For instance, Ma et al. [6] showed that a classifier trained with differential privacy certifies robustness against data poisoning attacks. Rosenfeld et al. [7] leveraged *randomized smoothing* [2] to certify robustness against data poisoning attacks. However, these certified defenses suffer from two major limitations. First, they are only applicable to limited scenarios, i.e., Ma et al. [6] is limited to learning algorithms that can be differentially private, while Rosenfeld et al. [7] is limited to data poisoning attacks that only modify existing training examples. Second, their certified robustness guarantees are loose, meaning that a learning algorithm is certifiably more robust than their guarantees indicate.

We aim to address these limitations in this work. Our approach is based on a well-known ensemble learning method called *Bootstrap Aggregating (bagging)* [1]. Bagging first generates N subsamples by sampling from the training dataset with replacement uniformly at random, where each subsample includes k training examples. Then, bagging uses a base learning algorithm to train a base classifier on each subsample. Given a testing example, bagging uses each base classifier to predict its label and takes majority vote among the predicted labels as the final predicted label. Our first major theoretical result is that we prove the ensemble classifier in bagging predicts the same label for a testing example when the number of poisoned training examples is no larger than a threshold (called *certified poisoning size*). Our second major theoretical result is that we prove our derived certified poisoning size is tight if no assumptions on the base learning algorithm are made. Note that the certified poisoning sizes may be different for different testing examples. Moreover, we design an

NeurIPS 2020 Workshop on Dataset Curation and Security

efficient algorithm to compute our certified poisoning size. We note that a concurrent work [5] proposed to certify robustness against data poisoning attacks via partitioning the training dataset using a hash function. However, their results are only applicable to deterministic learning algorithms. We empirically evaluate our method on MNIST and CIFAR10. For instance, our method can achieve a certified accuracy of 91.1% on MNIST when 100 training examples are arbitrarily poisoned, where $k = 100$ and $N = 1,000$. Under the same attack setting, Ma et al. [6] and Rosenfeld et al. [7] achieve 0 certified accuracy on a simpler MNIST 1/7 dataset.

All our proofs can be found in our technical report [3].

2 Certified Robustness of Bagging

Assuming we have a training dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$ with n examples. Moreover, we are given an arbitrary deterministic or randomized base learning algorithm $\mathcal{A}$. For convenience, we jointly represent the training and testing processes as $\mathcal{A}(\mathcal{D}, \mathbf{x})$, which is $\mathbf{x}$'s label predicted by a classifier that is trained using algorithm $\mathcal{A}$ and training dataset $\mathcal{D}$.

Data poisoning attacks: In a data poisoning attack, an attacker can carefully *modify*, *delete*, and/or *insert* some training examples in $\mathcal{D}$ such that $\mathcal{A}(\mathcal{D}, \mathbf{x}) \neq \mathcal{A}(\mathcal{D}', \mathbf{x})$ for many testing examples $\mathbf{x}$ or some attacker-chosen $\mathbf{x}$, where $\mathcal{D}'$ is the poisoned training dataset. We denote the set of poisoned training datasets with at most r poisoned training examples as $B(\mathcal{D}, r) = \{\mathcal{D}' | \max\{|\mathcal{D}|, |\mathcal{D}'|\} - |\mathcal{D} \cap \mathcal{D}'| \leq r\}$. Intuitively, $\max\{|\mathcal{D}|, |\mathcal{D}'|\} - |\mathcal{D} \cap \mathcal{D}'|$ is the minimum number of modified/deleted/inserted training examples that can change $\mathcal{D}$ to $\mathcal{D}'$.

Bootstrap aggregating (Bagging) [1]: Bagging is a well-known ensemble learning method. We describe a probabilistic view of bagging, which makes it possible to theoretically analyze its certified robustness against data poisoning attacks. Specifically, we denote by $g(\mathcal{D})$ a random subsample, which is a list of k examples that are sampled from $\mathcal{D}$ with replacement uniformly at random. We use the base learning algorithm $\mathcal{A}$ to learn a base classifier on $g(\mathcal{D})$. Due to the randomness in sampling the subsample $g(\mathcal{D})$ and the (randomized) base learning algorithm $\mathcal{A}$, the label $\mathcal{A}(g(\mathcal{D}), \mathbf{x})$ predicted by the base classifier learnt on $g(\mathcal{D})$ for $\mathbf{x}$ is random. We denote by $p_j = \Pr(\mathcal{A}(g(\mathcal{D}), \mathbf{x}) = j)$ the probability that the learnt base classifier predicts label j for $\mathbf{x}$, where $j = 1, 2, \cdots, c$. We call p_j *label probability*. The *ensemble classifier* h in bagging essentially predicts the label with the largest label probability for $\mathbf{x}$, i.e., we have $h(\mathcal{D}, \mathbf{x}) = \operatorname{argmax}_{j \in \{1,2,\cdots,c\}} p_j$, where $h(\mathcal{D}, \mathbf{x})$ is the predicted label for $\mathbf{x}$ when the ensemble classifier h is trained on $\mathcal{D}$.

Certified robustness of bagging: We prove the certified robustness of bagging against data poisoning attacks. In particular, we show that the ensemble classifier in bagging predicts the same label for a testing example when the number of poisoned training examples is no larger than some threshold (called *certified poisoning size*). Moreover, we prove our derived certified poisoning size is tight. Formally, we have the following two theorems.

Theorem 1 (Certified Poisoning Size of Bagging)**.** *Given a training dataset $\mathcal{D}$, a deterministic or randomized base learning algorithm $\mathcal{A}$, a testing input $\mathbf{x}$, and the ensemble classifier h in bagging. Suppose l and s respectively are the labels with the largest and second largest label probabilities predicted by h for $\mathbf{x}$. Moreover, the probability bounds $\underline{p_l}$ and $\overline{p}_s$ satisfy the following:*

$$p_l \geq \underline{p_l} \geq \overline{p}_s \geq p_s = \max_{j \neq l} p_j. \tag{1}$$

Then, we have $h(\mathcal{D}', \mathbf{x}) = l, \forall \mathcal{D}' \in B(\mathcal{D}, r^*)$, where r^* is the solution to the following problem:

$$r^* = \operatorname*{argmax}_r r, \textit{s.t.} \max_{|n'-n| \leq r} (\frac{n'}{n})^k - 2 \cdot (\frac{\max(n, n') - r}{n})^k + 1 - (\underline{p_l} - \overline{p}_s - \delta_l - \delta_s) < 0, \tag{2}$$

where $n = |\mathcal{D}|$, $n' = |\mathcal{D}'|$, $\delta_l = \underline{p_l} - (\lfloor \underline{p_l} \cdot n^k \rfloor)/n^k$, and $\delta_s = (\lceil \overline{p}_s \cdot n^k \rceil)/n^k - \overline{p}_s$.

Theorem 2 (Tightness of the Certified Poisoning Size)**.** *Assuming we have $\underline{p_l} + \overline{p}_s \leq 1$, $\underline{p_l} + (c-1) \cdot \overline{p}_s \geq 1$, and $\delta_l = \delta_s = 0$. Then, for any $r > r^*$, there exist a base learning algorithm $\overline{\mathcal{A}^*}$ consistent with (1) and a poisoned training dataset $\mathcal{D}'$ with r poisoned training examples such that $h(\mathcal{D}', \mathbf{x}) \neq l$ or there exist ties.*

Algorithm 1 CERTIFY

Input: $\mathcal{A}$, $\mathcal{D}$, k, N, $\mathcal{D}_e$, α.
Output: Predicted label and certified poisoning size for each testing example.
$f_1, f_2, \cdots, f_N \leftarrow$ TRAINUNDERSAMPLE$(\mathcal{A}, \mathcal{D}, k, N)$
for $\mathbf{x}_i$ **in** $\mathcal{D}_e$ **do**
  counts$[j] \leftarrow \sum_{o=1}^{N} \mathbb{I}(f_o(\mathbf{x}_i) = j), j \in \{1, 2, \cdots, c\}$
  $l_i, s_i \leftarrow$ top two indices in counts (ties are broken uniformly at random).
  $\underline{p_{l_i}}, \overline{p}_{s_i} \leftarrow$ SIMUEM(counts, $\frac{\alpha}{e}$)
  if $\underline{p_{l_i}} > \overline{p}_{s_i}$ **then**
    $r_i^* \leftarrow$ BINARYSEARCH$(\underline{p_{l_i}}, \overline{p}_{s_i}, k, |\mathcal{D}|)$
  else
    $l_i, r_i^* \leftarrow$ ABSTAIN, ABSTAIN
  end if
end for
return $l_1, l_2, \cdots, l_e$ and $r_1^*, r_2^*, \cdots, r_e^*$

3 Computing the Certified Poisoning Size

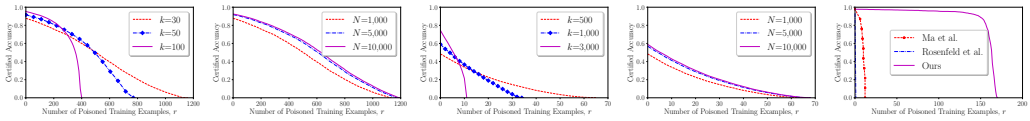
Given a base learning algorithm $\mathcal{A}$, a training dataset $\mathcal{D}$, subsampling size k, and e testing examples in $\mathcal{D}_e$, we aim to compute the label l_i predicted by the ensemble classifier and the corresponding certified poisoning size r_i^* for each testing input $\mathbf{x}_i$. For a testing input $\mathbf{x}_i$, our certified poisoning size relies on a lower bound $\underline{p_{l_i}}$ of the largest label probability and an upper bound $\overline{p}_{s_i}$ of the second largest label probability. We design a Monte-Carlo algorithm to estimate the probability bounds for the e testing examples simultaneously via training N base classifiers. Algorithm 1 shows our algorithm CERTIFY to estimate the predicted labels and certified poisoning sizes for e testing examples in $\mathcal{D}_e$. The function TRAINUNDERSAMPLE randomly samples N subsamples and trains N base classifiers. The function SIMUEM estimates the probability bounds $\underline{p_{l_i}}$ and $\overline{p}_{s_i}$ with confidence level $1 - \frac{\alpha}{e}$. In particular, we have $\underline{p_{l_i}} = \text{Beta}(\frac{\alpha}{c}; N_{l_i}, N - N_{l_i} + 1)$ and $\overline{p}_j = \text{Beta}(1 - \frac{\alpha}{c}; N_j, N - N_j + 1)$, $\forall j \neq l_i$, where $1 - \alpha$ is the confidence level and $\text{Beta}(\beta; \lambda, \theta)$ is the βth quantile of the Beta distribution with shape parameters λ and θ. Based on the *Bonferroni correction*, the simultaneous confidence level of estimating the probability bounds for the e testing examples is at least $1 - \alpha$ [4]. Moreover, we estimate $\overline{p}_{s_i}$ as $\overline{p}_{s_i} = \min(\max_{j \neq l_i} \overline{p}_j, 1 - \underline{p_{l_i}})$. The function BINARYSEARCH solves the optimization problem in (2) via binary search to obtain the certified poisoning size r_i^* for $\mathbf{x}_i$. Since the probability bounds are estimated using a Monte-Carlo algorithm, they may be estimated incorrectly. When they are estimated incorrectly, our algorithm CERTIFY may output an incorrect certified poisoning size. However, the following theorem shows that the probability that CERTIFY returns an incorrect certified poisoning size for at least one testing example is at most α.

Theorem 3. *The probability that* CERTIFY *returns an incorrect certified poisoning size for at least one testing example in $\mathcal{D}_e$ is at most α, i.e., we have: Pr$(\cap_{\mathbf{x}_i \in \mathcal{D}_e}((\forall \mathcal{D}' \in B(\mathcal{D}, r_i^*), h(\mathcal{D}', \mathbf{x}_i) = l_i) | l_i \neq$ ABSTAIN$)) \geq 1 - \alpha$.*

4 Experiments

Datasets and classifiers: We use MNIST and CIFAR10. The base learning algorithm is neural network, and we use the example convolutional neural network architecture and ResNet20 in Keras for MNIST and CIFAR10, respectively. Both datasets have 10,000 testing examples, which are the $\mathcal{D}_e$ in our algorithm. When we train a base classifier, we adopt the example data augmentation in Keras for both datasets.

Evaluation metric: We use *certified accuracy* as our evaluation metric. Formally, we define the certified accuracy CA_r at r poisoned training examples as follows: $CA_r = \frac{\sum_{\mathbf{x}_i \in \mathcal{D}_e} \mathbb{I}(l_i = y_i) \cdot \mathbb{I}(r_i^* \geq r)}{|\mathcal{D}_e|}$, where y_i is the ground truth label for testing input $\mathbf{x}_i$, and l_i and r_i^* respectively are the label predicted by the classifier and the corresponding certified poisoning size for $\mathbf{x}_i$. Intuitively, CA_r of a classifier means that the classifier's testing accuracy for $\mathcal{D}_e$ is at least CA_r no matter how the attacker manipulates at most r poisoned training examples.



(a) (b) (c) (d) (e)

Figure 1: Impact of k and N on our method for MNIST ((a)-(b)) and CIFAR10 ((c)-(d)). (e) Comparing our method with existing methods.

Parameter setting: Unless otherwise mentioned, we adopt the following default parameter settings for our method: $\alpha = 0.001$, $N = 1,000$, $k = 30$ for MNIST, and $k = 500$ for CIFAR10.

Impact of k and N: Figure 1 shows the impact of k and N on the certified accuracy of our method. As the results show, k controls a tradeoff between accuracy under no poisoning and robustness. Specifically, when k is larger, our method has a higher accuracy when there are no data poisoning attacks (i.e., $r = 0$) but the certified accuracy drops more quickly as the number of poisoned training examples increases. The reason is that a larger k makes it more likely to sample poisoned training examples when creating the subsamples in bagging. The certified accuracy increases as N increases. The reason is that a larger N produces tighter estimated probability bounds.

Comparing with Ma et al. [6] and Rosenfeld et al. [7]: Since these methods are not scalable because they train N classifiers on the entire training dataset, we perform comparisons on the MNIST 1/7 dataset that just consists of the digits 1 and 7. This subset includes 13,007 training examples and 2,163 testing examples. Figure 1(e) shows the comparison results, where $k = 50$, $\alpha = 0.001$, and $N = 1,000$. To be consistent with previous work, we did not use data augmentation when training the base classifiers for all three methods in these experiments. Our method significantly outperforms existing methods. For example, our method can achieve 96.95% certified accuracy when the number of poisoned training examples is $r = 50$, while the certified accuracy is 0 under the same setting for the two existing methods. Ma et al. outperforms Rosenfeld et al. because differential privacy directly certifies robustness against modification/deletion/insertion of training examples while randomized smoothing was designed to certify robustness against modifications of features/labels.

5 Conclusion

In this work, we show the intrinsic certified robustness of bagging against data poisoning attacks. Specifically, we show that bagging predicts the same label for a testing example when the number of poisoned training examples is bounded. Moreover, we show that our derived bound is tight if no assumptions on the base learning algorithm are made. Our results on MNIST and CIFAR10 show that our method achieves much better certified robustness than existing certified defenses.

ACKNOWLEDGMENTS: We thank the anonymous reviewers for insightful reviews. This work was supported by NSF grant No. 1937786.

References

[1] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[2] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[3] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. *arXiv preprint arXiv:2008.04495*, 2020.

[4] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *ICLR*, 2020.

[5] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.

[6] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *International Joint Conference on Artificial Intelligence*, 2019.

[7] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and J Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *ICML*, 2020.