Bagging model against Data Poisoning Attacks

Manita Ngarmpaiboonsombat
manita_ngarmpaiboomsombat@student.uml.edu

***Abstract*— This study is inspired by the paper "Intrinsic Certified Robustness of Bagging against Data Poisoning Attacks" [1]. This paper shows the bound of data poisoning attacks that any arbitrary bagging model can still predict the same labels for the testing examples. The poisoning attacks include modifying, deleting, and inserting training examples. Using the provided code, we tested the original code with MNIST dataset to compare performance with the real-world situation. Our results show differences from the paper's results. However, we observed a similar trend as the paper for large subsample size. It shows higher certified accuracy when there are no data poisoning attacks, but it significantly drops as the number of poisoned training examples increases. Then, we experimented with modifying the training examples in the data poisoning attack. We evaluate our study based on the accuracy of bagging model of the same dataset when training with the different size of the poisoning attack and different network. The result shows a different trend from the paper's result. We observed the accuracy is significantly lower at higher size of poisoning.**

I. Introduction

Nowadays machine learning models based on neural networks are the powerful methods in the industry. However, they are also vulnerable to adversarial attacks. One of most popular attack method is the data poisoning attack which pose severe security threats to machine learning systems. Certified defenses were proposed to derive the boundaries of the number of poisoning training examples. Therefore, the networks are certifiably robust against data poisoning attacks if it can predict the same label for the testing examples when the poisoning training examples size is within the boundary. Knowing the certified robustness benefits to defender because they can design more secure models, while attacker benefits because they can construct adversarial examples that can be misclassified without detection.

II. Literature Review

The paper aims to address the previous major limitations. First, they were only applicable to limited scenarios such as data poisoning attacks that only modify existing training examples [2], [3]. Second, their certified robustness could not certify that the learnt model predicts the same label for a testing example [4]. To address these limitations the paper introduced the certified robustness of bagging against data poisoning attacks and proved the tightness of their robustness guarantee by developing the algorithms to compute the certified poisoning size and then evaluating on MNIST and CIFAR10.

A. Certified Robustness of Bagging

The goal is to find the maximal poisoning size r such that the network still predicts label l for x when trained on the poisoned training dataset with at most r poisoned training examples. Theoretically, the following inequality has to be satisfied where D' is the poisoned training dataset, l is predicted label, and j is the second largest probability label.

$$h(\mathcal{D}', \mathrm{x}) = l \Longleftrightarrow p_l' > \max_{j \neq l} p_j'.$$

B. Computing the certified poisoning size

The certified poisoning size is computed based on a lower bound P_l of the largest label probability and an upper bound P_s of the second largest label probability. The algorithm was designed with TrainUnderSample function which randomly separates the N subsamples and trains N base classifiers, SimuEM function which estimates the probability bounds of P_l and P_s and BinarySearch function which solves the optimization problem using the estimated probability bounds P_l and P_s to get the certified poisoning size r.

III. Reference Code Testing

The paper's code consists of 4 files: dataaug.py, training_bagging_mnist.py, compute_certified_poisoning_size.py and run.py. We ran it with the MNIST dataset and the original CNN architecture. Then, we set up the required parameters which are k = [30, 50, 100], end = 1000 and α = 0.001.

- k is the number of subsamples from the training examples with replacement, with default 30.
- end is the number of base classifiers, with default 1
- α is the confidence level, with default 0.001

Evaluation metric is the certified accuracy which is the lower bound of testing accuracy. It is defined from the number of the testing labels that are correctly predicted and the size of certified poisoning are at least r.

The run.py file started working by running training_bagging_mnist.py on the clean dataset. Before starting to train the model, dataaug.py is called for data augmentation and then train with bagging method with CNN as base algorithm. The result collects all label frequencies and the predicted label. After that, it continues running the compute_certified_poisoning_size.py to compute the probability bounds of P_l and P_s, and then the certified accuracy.

A. Result

Although our certified accuracies (Fig. 1) are not similar to their results (Fig. 2), we observed that our certified accuracy is equal to 0 at a similar poisoning size. Our results show the larger the poisoning size, the lower the accuracy.

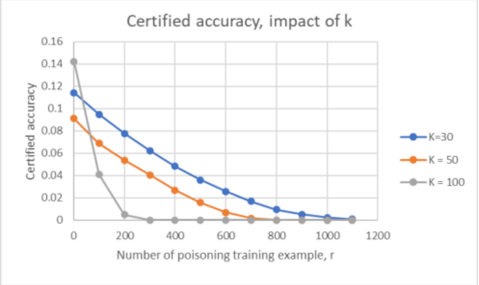

Fig. 1 Our certified accuracy

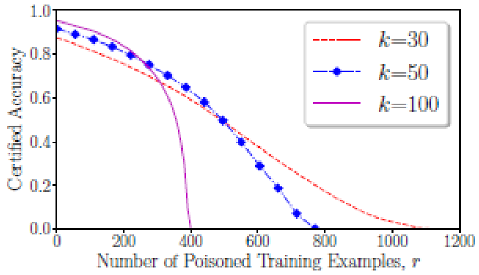

Fig. 2 Paper's certified accuracy

Moreover, the same trend is observed when k is larger. It shows the higher accuracy when there are no data poisoning attacks but the certified accuracy significantly drops 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. In the future we could investigate why these results are different from the paper.

IV. EXPERIMENT

After we studied the paper's code and observed that they used the Adadelta optimizer which is not often used with CNN models and provides lower accuracy for the MNSIT dataset (around 20%). We decided to experiment with the network by adjusting the optimizers, and then continued training with bagging method. In addition, we tested those networks with different data poisoning sizes.

We poisoned the training network by changing the label. First, we created poisoning_label function to change the label to 0. If the label was equal to 0, we changed it to 1 instead. This function required two inputs which are label array and poisoning size. Secondly, we created count_label function to check each digit label before and after we poisoned the training label. Finally, we created bagging_accuracy function to calculate the model accuracy at the end.

A. *Experiment setup*

Due to limited time and computational resources, we set N equal to only 100 and k = 30. We experimented with three optimizers which were Adadelta, Adam and SGD and a range of poisoning sizes [0, 1000, 2000, 3000, 4000].

B. *Experiment results*

The results show different trend from the paper's result. Since our method is classified as the modification attack, the accuracy should significantly drop when the poisoning size is 1000 (Fig. 4, right).

We observed the expected trend that the larger the poisoning size, the lower the accuracy. However, our graphs (Fig. 3 and Fig. 4, left) show the accuracy markedly decrease after poisoning size of 3000 and then the accuracy suddenly shift back at the poisoning size of 4000 on all optimizers. Our suspicion is that it might come from either the effect of lower N value or the attack method that we targeted to change labels to be only 0 and 1. This trend needs to be investigated further in a future work.

R	Adadelta	Adam	SGD
0	16.18	93.08	84.36
1000	14.59	92.86	84.11
2000	14.14	92.78	83.7
3000	10.18	92.47	82.38
4000	12.41	92.64	83.21

Table 1 Model accuracy with label poisoning attack

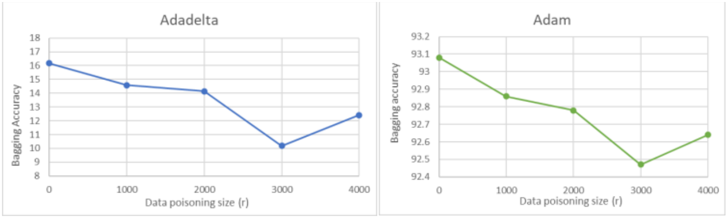

Fig. 3 Adadelta optimizer (left) and Adam optimizer (right) with label poisoning attack

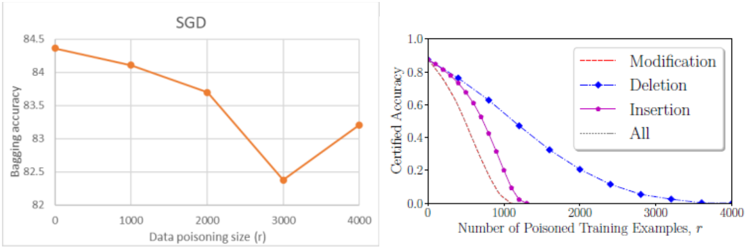

Fig. 4 SGD optimizer with label poisoning attack (left) and Adadelta optimizer with data poisoning attack from the paper result (right)

V. CONCULSION

In this project, we studied the paper which introduced the bound of data poisoning attack that the CNN network with bagging ensemble method still can predict the same label for testing examples. In this work, we have tested with the code that they provided from the paper. Even though our results are not the same as their paper, we observed the same trend on poisoning size that shows the zero certified accuracy and the k value. When k is large, the certified accuracy is high and then falls significantly when the poisoning size rises. Moreover, we conducted the label poisoning attack with slight adjustments to the CNN network by changing the model's optimizer. The results show a different trend from the paper's result. We observed our accuracy markedly drops and then suddenly increases after the same poisoning size which we need to further investigate in the future.

REFERENCES

[1] J. Jinyuan, C. Xiao yu, G. Neil Zhenqiang, "Intrinsic Certified Robustness of Bagging against Data Poisoning Attacks", December 2020.

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

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

[4] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In Advances in neural information processing systems, pages 3517–3529, 2017.