CAN ENSEMBLES DEFEND AGAINST BACKDOOR POISONING ATTACKS?

Pramodh Gopalan
Northeastern University
Indian Institute of Technology Kanpur
{pramodh}@iitk.ac.in

Alina Oprea
Northeastern University
Khoury College of Computer Science
{a.oprea}@northeastern.edu

ABSTRACT

Modern Machine Learning has achieved state-of-the-art accuracies on a wide range of Visual and Text based tasks. However, when such systems are deployed, they are susceptible to attacks during train and test time, affecting their ability to infer precisely. In particular, Poisoning attacks on Machine Learning incorporate adversarially manipulated points during train time that aim to alter their predictions on test data selectively. In this report, we introduce a novel method to defend against poisoning attacks, using a combination of feature selected models. First, we formalize the problem and derive a theoretical bound on the worst possible accuracy of the poisoned ensemble. Next, we demonstrate that the defense works on malware and elementary vision datasets when tested against poisoning attacks.

1 Introduction

Modern day Industries are increasingly incorporating Machine Learning (ML) models in their services and production pipelines. Models are becoming more complex as each day progresses, and seem to require larger amount of train data as a result. Thus, training data is often scraped from public sources on the internet in substantial quantities; as a result, these sources are often not verified. In such a case, an adversary can add maliciously crafted points to the train set, which influences the training process of these models. Often referred to as *Poisoning Attacks*, have been shown to be extremely effective when applied to vision datasets. In particular, *backdoor poisoning attacks* are being applied in various use cases of Machine Learning, such as Malware Identification, Language Models, and Segmentation models.

Backdoor poisoning attacks attempt to add a pattern to the feature space, such that the model learns to associate the backdoor pattern with a particular label of the attackers choice. One distinctive feature of backdoor attacks is that they manipulate only a small subset of feature space, which might prove to be a loophole on which a defense can be constructed. If we are able to guess which features were being manipulated by the adversary generate poisoned samples, we can simply discard those features, and train a model which would be clean.

Unfortunately, as of now, there are no methods to do so, and hence we turn to sampling features from the dataset at random. Since the number of backdoors are small, there is a good chance that we discard a few of them while sampling features at random. If we were to do this multiple times, and create an ensemble of models with different sets of features, it might be more robust to backdoor poisoning attacks. In this report, we build on this intuition and try validate it through empirical and theoretical analysis.

2 Background

Machine Learning models, especially neural networks, require a large amount data to excel at the learning task. Typically, the data for such use cases is gathered from a variety of sources, all of which cannot be trusted and verified. In such cases, the model is vulnerable to poisoning attacks, wherein an adversary can add malicious data in the train set.

Formally, the attacker adds m *poisoned* points $\mathcal{D}_p = \{x_i, y_i\}_{i=1}^{m}$ to the original dataset $\mathcal{D}$. Thus, the *poisoned* model $\mathcal{M}$ with parameters θ minimizes its loss $l(\mathcal{D} \cup \mathcal{D}_p, \mathcal{M})$, rather than $l(\mathcal{D}, \mathcal{M})$. The poison points are chosen such

that the performance of the model is degraded on a target distribution. Existing poisoning attacks can be classified into three classes based on the target distribution they choose.

Availability Attacks In an availability attack, the adversary aims to reduce the model's classification performance impartially, without targeting a subset of the attack.

Targeted Attacks In a targeted attack, the adversary has a set of points which they wish to misclassify.

Backdoor Attacks A Backdoor attack is one in which the adversary has the ability to manipulate certain features of the dataset referred to as *backdoors*. These backdoors usually consist of patterns in data that the ML model is able to pick up on. The adversary can also choose not to manipulate the labels of the poisoned dataset, thereby leading to *Clean Label Attacks*. In test time, the adversary supplants the same backdoors in the samples, and is able to get predictable misclassification on them.

In our report, we will be focusing mostly on backdoor attacks and how to defend against them.

3 Definitions and Defense Framework

3.1 Definitions

We consider a dataset $\mathcal{D}$ with feature set $\mathcal{F}$ where $|\mathcal{F}| = d$. Let $\mathcal{D}_p$ denote the set of poisoned points, with the set of poisoned features $\mathcal{B} \subseteq \mathcal{F}$, and $|\mathcal{B}| = b$. We denote $\mathcal{D}' = \mathcal{D} \cup \mathcal{D}_p$. Here, $\mathcal{D}_p$ makes up $p\%$ of $\mathcal{D}'$.

Let $\mathcal{M}_p$ and $\mathcal{M}$ denote models trained using the same algorithm $\mathcal{A}$, but with train datasets $\mathcal{D}'$ and $\mathcal{D}$. Let $\mathcal{T}$ and $\mathcal{T}_p$ be clean and poisoned testing points, with $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}_p$. We define ***Acc(M, D)*** as the accuracy of model *M* on the set of samples *D*.

Let the ensemble E consist of T models labeled with an index $i \in \{1, 2 \dots T\}$, based on the same learning algorithm $\mathcal{A}$. For each model i , we randomly sample $\mathcal{K}_i \subseteq \mathcal{F}$, with $|\mathcal{K}_i| = k$. We then train model i on feature set $\mathcal{K}_i$. During test time, model i evaluates test samples based on feature set $\mathcal{K}_i$, and the final prediction on the sample is realized through a hard voting scheme. We make two important assumptions while formulating a mathematical model:

- For any model i, if $\mathcal{K}_i \cap B \neq \phi$, then model i is considered to be *poisoned*, and the poisoning attack has full efficacy i.e, all poisoned test points will be mispredicted.
- For any model i, if $\mathcal{K}_i \cap B = \phi$, then model i is considered to be *clean*, and the poisoning attack has no effect i.e, all poisoned test points will be predicted correctly.

We acknowledge that these assumptions are not practical: The accuracy of the ensemble on any sample depends on the dataset, the model, the training method and its hyperparameters, the value of k chosen, the attack used and the parameters of the attack itself. We argue that these assumptions give a reliable baseline, to which other frameworks can be compared against.

3.2 Metrics for testing efficiency of Backdoor Attacks

To measure the efficiency of the attacks we impose on the models, we propose three metrics and their interpretations:

***Acc($\mathcal{M}$, $\mathcal{T}_p$)*:** We require that the clean model have high accuracy on the poisoned test samples.

Poisoning attacks make an attempt to change how a few *backdoor* features are perceived by model, by affecting it during train time. Therefore a model $\mathcal{M}$, which has been trained on clean dataset $\mathcal{D}$ should not be affected by poisoned test samples, and should be able to classify them correctly. If **Acc**($\mathcal{M}$, $\mathcal{T}_p$) is low, it means that the poisoning samples are *evading* the model, akin to an adversarial attack. While there is no necessity for **Acc**($\mathcal{M}$, $\mathcal{T}_p$) be extremely high, it's lack thereof could be a potential indicator of attack failure.

***Acc($\mathcal{M}_p$, $\mathcal{T}_p$)*:** We require that the poisoned model have low accuracy on the poisoned test samples.

***Acc($\mathcal{M}_p$, $\mathcal{T}$)*:** We require that the poisoned model have high accuracy on the clean test samples.

3.3 A Lower Bound on Worse case Test Error

Theorem 3.1. *Given the definitions above, the worst case test error on samples has a lower bound of* $1 - e^{-\frac{T}{2\alpha}(\alpha - \frac{1}{2})^2}$, *where* $\alpha = \frac{\binom{d-b}{k}}{\binom{d}{k}}$

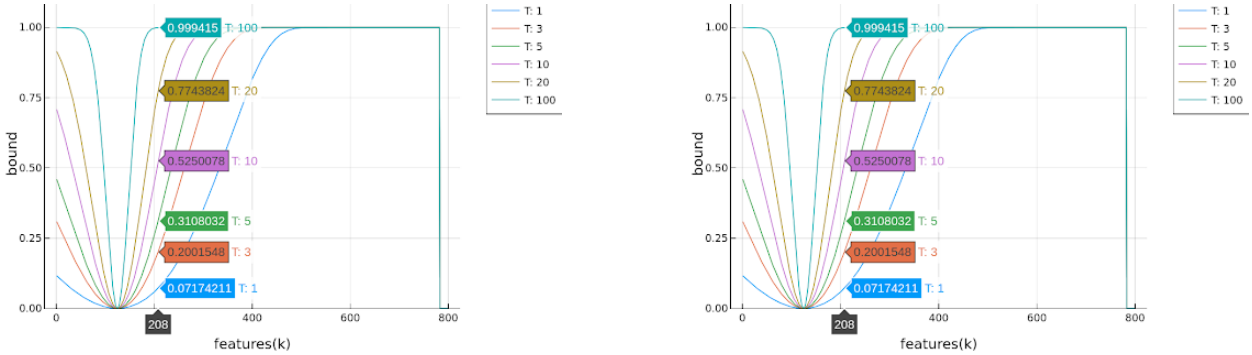
Proof. Let X_i be a bernoulli random variable denoting the event $\mathcal{K}_i \cap \mathcal{B} = \phi$. We then have,

$$P(X_i = 1) = \alpha = \frac{\binom{d-b}{k}}{\binom{d}{k}}$$

We denote $X = \sum_{i=1}^{T} X_i$, and through the chernoff bound, we get the following:

$$P(X > T/2) \geq 1 - e^{-\frac{T}{2\alpha}(\alpha - \frac{1}{2})^2}$$

If there are more than half the models in E whose feature sets $\mathcal{K}_i$ don't have any intersection with the backdoor set $\mathcal{B}$, then by our assumptions, the above bound functions as a lower bound for our test error. □



(a) The Chernoff bound as we vary k

(b) The Chernoff bound as we vary T

Figure 1: A Plot of Chernoff bound as a function of T and k, for the MNIST dataset with $b = 4$, and $d = 784$

4 Experiments

We conduct experiments on three datasets: **Drebin**, **Drebin-991**(A feature selected version of the original drebin dataset), and a subset of the **MNIST** dataset consisting of only two classes, 0 and 1. We choose two attacks to test our defense on, the **Explanation based backdoor** attack proposed by Severi et.al and the **BadNets** attack proposed by Gu et.al

Explanation based backdoors[1] is a *Clean Label Attack* which uses SHAPley values to find and alter backdoor features. It has been shown to be highly effective in poisoning malware classifiers, and can bypass recent defenses proposed such as Activation Clustering [2], and Spectral Signatures [3].

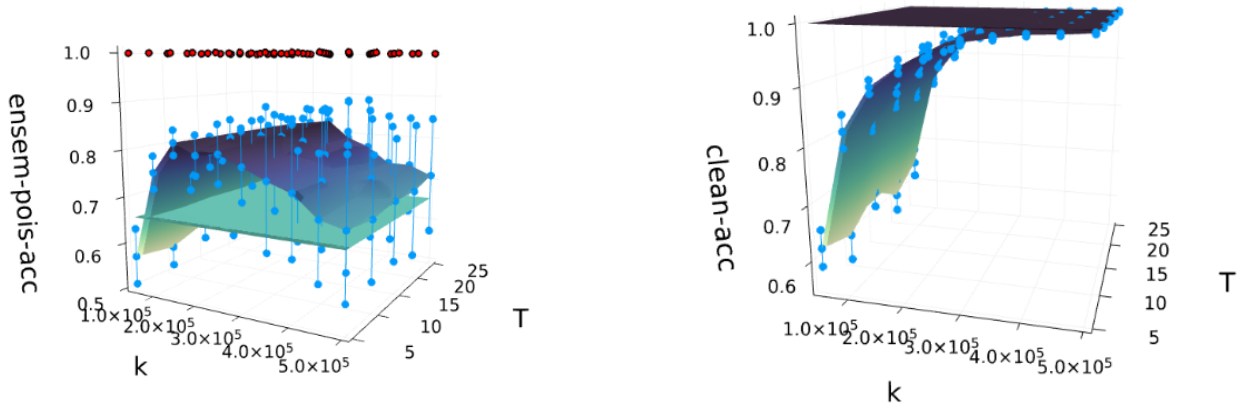
Badnets[4] is backdoor attack on predominantly vision datasets, where in an adversary adds in a pattern to the input image, and changes the label correspondingly. Although simple in design, it works well in practice.

Given attack A, we compare it's effectiveness on Ensemble E, and normal model $\mathcal{M}$ using the metrics highlighted in section 3.2.

4.1 Drebin/Explanation based backdoors

Drebin is a dataset based on android consisting of statically extracted features from around 6000 malware and 123,000 goodware android apps. It has around 540,000 features, all of them binary. We run the attack with 1-3% poisoning percentage and 30-60 backdoor features. We choose ensembles with k ranging from 100,000-500,000, and T ranging from 5-25. On all these tests, for chosen values of T and k, we observe that the $Acc(E_p, \mathcal{D}_p)$ is well above that of $Acc(\mathcal{M}_p, \mathcal{D}_p)$, where E_p and $\mathcal{M}_p$ denote the poisoned ensemble and the poisoned model.

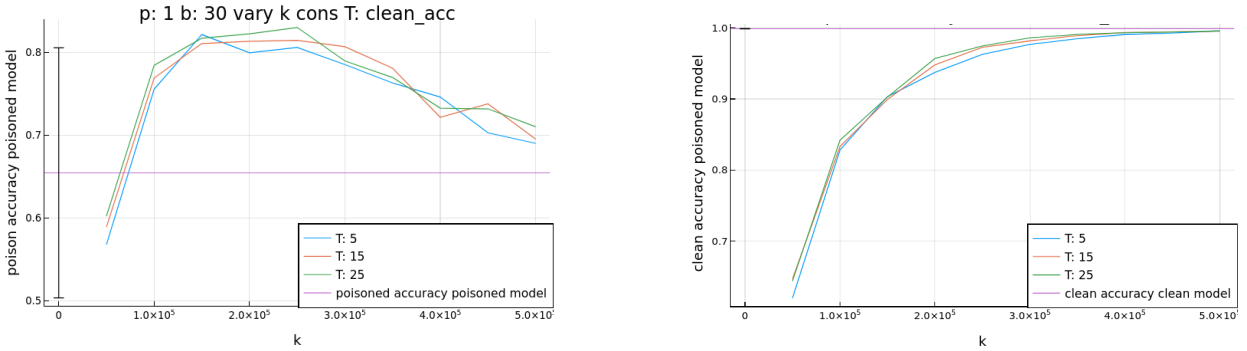
We make a few observations in Figure 3. When k is low, the ensemble fails to capture any meaningful features, and hence $Acc(E_p, \mathcal{D}_p)$ is low. When $k \approx d$, a majority of models in the ensemble have intersections with the backdoor features, thus bringing $\text{Acc}(E_p, \mathcal{D}_p) \approx \text{Acc}(E_p, \mathcal{D}_p)$. But when k, is aptly chosen, the poisoned ensemble performs about 60% when compared to the poisoned normal model. We also observe that the parameter T affects $Acc(E_p, \mathcal{D}_p)$, but not much in comparison to k. Hence, to bring some diversity into each individual model, we decided to use bagging in the subsequent experiments.



(a) $\text{Acc}(E_p, \mathcal{T}_p)$ as we vary both k and T

(b) $\text{Acc}(E_p, \mathcal{T})$ as we vary both k and T

Figure 2: A plot of poison and clean accuracies of ensemble E compared against normal model $\mathcal{M}$ as a baseline, evaluated on the drebin dataset and explanation based backdoor attack. The red dots on each of the figures show the chernoff bound.



(a) $\text{Acc}(E_p, \mathcal{T}_p)$ as we vary k, with a fixed T

(b) $\text{Acc}(E_p, \mathcal{T})$ as we vary k, with a fixed T

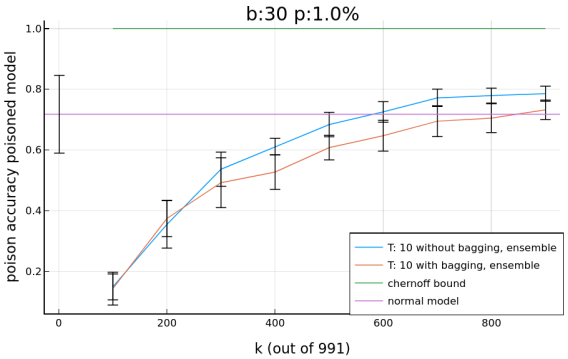
Figure 3: Plots of clean and poison accuracies of ensemble E compared against normal model $\mathcal{M}$ on the drebin dataset, with the backdoor explanation attack.

4.2 Drebin - 991/Explanation based backdoors

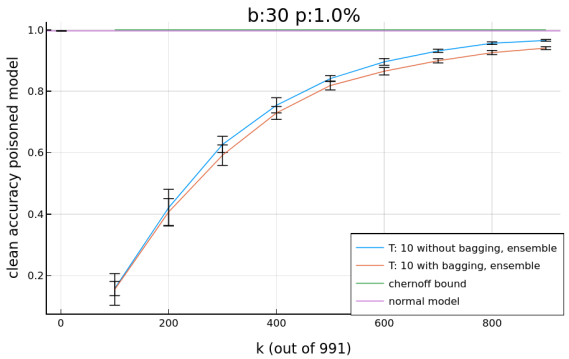
Drebin-991 is a compressed version of the Drebin dataset with 991 features, instead of $500,000$ features. The 991 features were selected using Lasso Regression. The dataset was then filtered to remove duplicates, which brought down the count of of malware to 800, and $40,000$ goodware. In this case, the poisoned ensemble was only able to perform slightly better than the poisoned normal model. We believe that this is due to the imbalanced dataset, which would lead to the model requiring all of the features before it can distinguish between malware and goodware correctly. We also compare if bagging has any effect on the overall results.

4.3 MNIST/Explanation based backdoors

We use a binary MNIST dataset with just two classes 0 and 1. Here we use the pixels themselves as individual features. On running the attacks, we see that $\text{Acc}(\mathcal{M}, \mathcal{D}_p)$ is around 40%, which indicates that the attack is not working as we



(a) Acc$(E_p, \mathcal{T}_p)$ as we vary k for a fixed T

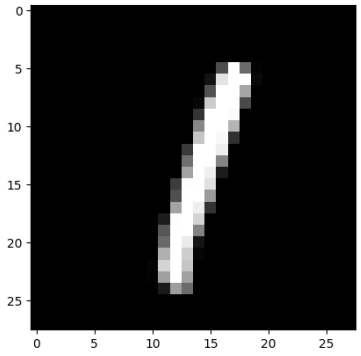


(b) Acc$(E_p, \mathcal{T})$ as we vary k, for a fixed T

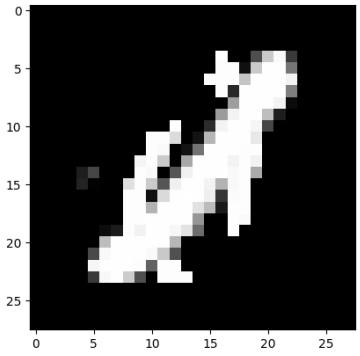
Figure 4: A plot of clean and poison accuracies of ensemble E compared against normal model $\mathcal{M}$ as a baseline, evaluated on the drebin-991 dataset and explanation based backdoor attack. The green lines on each of the figures denote the chernoff bound

.

expected to. From image 5, we can clearly see that the images are similar to adversarial attacks with large L_2 norm, which evade classification at test time, rather than at train-time. Hence, we don't evaluate results from this experiment.



(a) Unpoisoned MNIST image from class 1



(b) Unpoisoned MNIST image from class 1

Figure 5: A comparison of MNIST Images generated from the explanation based backdoor attack

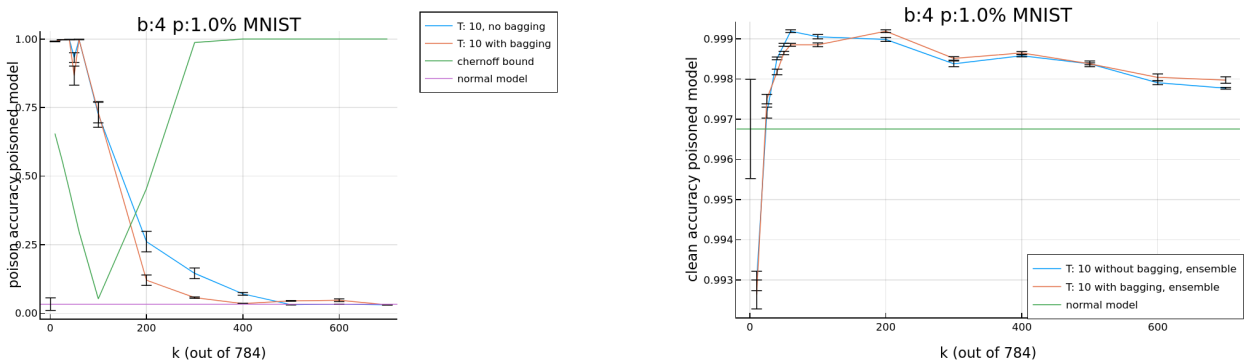
4.4 MNIST/BadNets

Similar to the previous experiment, we use a binary MNIST. We choose a backdoor size of four pixels, at 5% poisoning percentage to bring Acc$(\mathcal{M}_p, T_p)$ under 10%. A surprising result is that even when choosing $k = 10$, we are able to get high values(>99%) of Acc$(E_p, \mathcal{T}_p)$, with very loss in Acc$(E_p, \mathcal{T})$ (about 99%). Increasing T leads to results with less variance.

5 Conclusions and Future Work

In this report, we show that using Feature Selected Ensembles can help us mitigate backdoor attacks in Machine learning, provided that we pick and choose T and k carefully. There are many venues to improve upon the current work, here are some:

- We endeavour to extend this defense to CNNs, which are much more prevalent than Feedforward Neural Networks today.
- From the experiments, we observe that the chernoff bound derived by us was very loose, we hope to try and include more useful parameters in the mathematical model, and formulate better assumptions.



(a) Acc$(E_p, \mathcal{T}_p)$ as we vary k for a fixed value of $T = 10$

(b) Acc$(E_p, \mathcal{T})$ as we vary both k for $T = 10$

Figure 6: A plot of clean and poison accuracies of ensemble E compared against normal model $\mathcal{M}$ as a baseline, evaluated on the MNIST dataset and the BadNets attack. The green line on each of the figures show the chernoff bound.

- We aim to study how much of our original assumptions were actually true; i.e, what would be the minimum size of $\mathcal{K}_i \cap B$ to induce a misclassifcation?

6 Acknowledgements

I would like to thank Prof. Alina Oprea for giving me an opportunity to work under her remotely during the COVID-19 pandemic. I am grateful to have a mentor like her, she has taught me many things beyond the scope of the project which shall forever help me in my future research. Each meeting with her helped me learn something new about how research is done in academic settings. I would also like to thank Giorgio Severi and Matthew Jagielski for their invaluable feedback on the project and helping me with implementation issues as and when they rose. Finally, I would like to thank everyone in the NDS2 Lab at Northeastern University for all the discussions they involved me in, and for participating in my project presentation and posing crucial questions regarding the ideas and offering feedback to improve upon them.

References

[1] Giorgio Severi, Jim Meyer, Scott Coull, and Alina Oprea. Explanation-guided backdoor poisoning attacks against malware classifiers, 2021.

[2] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering, 2018.

[3] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks, 2018.

[4] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019.