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 method to defend against backdoor 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 Related Work

We provide some notes on literature closely related to certified defenses against evasion and poisoning attacks.

Certified Adversarial Robustness: Randomized Smoothing [1] They repeatedly add gaussian random noise to image x, to create a large set of noisy images. A *base classifier* is then used to classify these images robustly by taking majority vote. The main insight is that given a clean and adversarial image x and x', the expected number of votes for each class can only differ between x and x' by a bounded amount. We can also bound the 'gap' between number of votes given to the top class and the number of votes given to any other class. If this gap is large, then we can guarantee with high probability that robust classification at x' is the same as x. Hence, we get a radius under which there does not exist any adversarial example.

PatchGuard: Smaller Convolution kernels and Activation Masking [2] This paper provides a provably robust defense against patch adversarial attacks. It leverages two insights:

1. Large convolution kernels/ bigger image lead to the adversarial patch affecting most of the extracted features, leading to increased attack success. Hence, using smaller convolution kernel/ splitting image into patches and using ensembles helps immensely.
2. The smaller receptive fields would force the attacker to use bigger perturbations to affect the output. In this case, we can mask the activations so that it does not affect the decision of the Neural net.

Using this intuition, we can flag suspicious activations and remove them through masking. Choosing an appropriate value of the threshold to flag such activations leads to proving that this defense correctly classifies adversarial images.

Deep Partition Aggregation [3] This paper provides a provable defense against general poisoning attacks. We do the following:

1. Split the dataset into k disjoint partitions. Train one model on each parition.
2. During test time, we use k models as an ensemble and use hard voting to output the "correct" class.

They prove that, not changing the training set too much (through adding or removing poisoning samples) (please refer to the paper for upper bound) implies that the ensemble does not change it's predicted class.

Here are my views on this paper. Please do correct me if I'm wrong.

1. If the initial training set is already poisoned ($p\%$ poison samples) and splitting it equally into k disjoint sets still renders each individual dataset to be $p\%$ poisonous(in expectation). This would still render the k base classifiers of the ensemble to be poisoned and output the wrong result.
2. Note that the claim they make says: "The prediction does not change on adding/removing samples from training set", it does not say that "The *correct* prediction does not change". Such a certification might be misleading, since it does not guarantee that the ensemble initially(before changing the train dataset) outputs correct labels.

Having said this, If we assume the initial dataset that the models use is not poisoned, it would lead to robust results, and the certification proves to be extremely useful.

RAB: Provable defense to backdoors[4] The premise of this paper is very similar to the randomized smoothing paper explained above. Instead of a norm on the pixel space, they provide bounds on the sum of magnitude of backdoors. As an anologue, this requires them to instantiate multiple smoothed training sets, and train a model on each of these datasets. During test time, we use the empirical majority vote as the prediction of the classifier.

Yes, Malware Classifiers can be more secure [5] This paper uses the intuition that evenly distributed weights in an SVM increases robustness. The L_0 norm provides an upper bound on the "unevenness", which is used as a penalty function to train the function. The weight parameters are also constrained to lie inbetween two vectors. This provides an SVM with robustness guarantees.

4 Definitions and Defense Framework

4.1 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, $|\mathcal{D}_p| = p$, $|\mathcal{D}| = N$

Let $\mathcal{M}_p$ and $\mathcal{M}$ denote models(possibly ensemble of 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.

4.2 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.

4.3 A Lower Bound on Worse case Test Error

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

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

We denote $X = \sum_{i=1}^{T} X_i$, where $X \sim \text{Binomial}(T, \alpha)$ and observe that the quantity we need is $P(X \geq \frac{T}{2})$. This, by our assumptions outlined in section 4.1, would be an estimate of the test error when the model is poisoned. To estimate this, we use three methods, the derivations for which can be found in section 10.

Numerical Computation Since $X \sim \text{Binomial}(T, \alpha)$, we can directly compute $P(X \geq \frac{T}{2})$ as

$$P(X \geq T/2) = \sum_{i=T/2}^{T} \binom{T}{i} \alpha^i (1-\alpha)^{T-i}$$

Gaussian approximation of the Binomial A binomial distribution $\text{Binomial}(T, \alpha)$ can be approximated by the Gaussian distribution $\text{Normal}(T\alpha, T\alpha(1-\alpha)$. we compute $P(X \geq \frac{T}{2})$ as

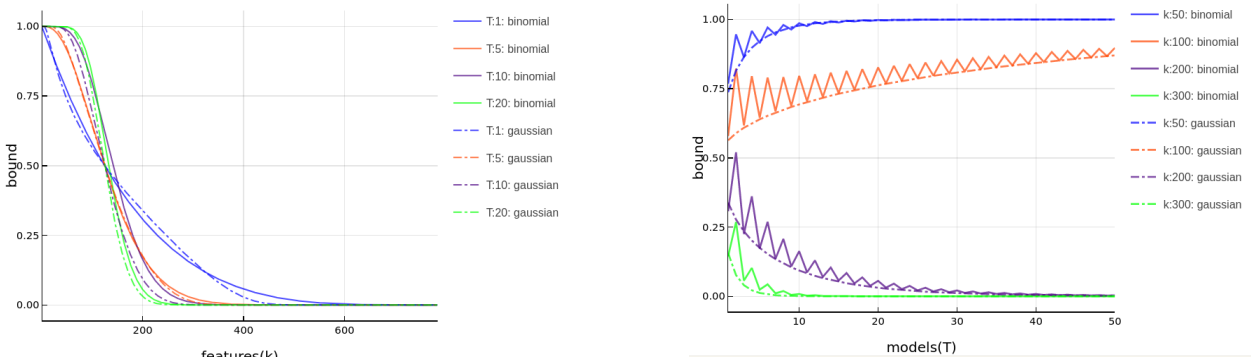
$$P(X \geq T/2) \approx 1 - cdf(T/2)$$

where cdf returns the cumulative distribution function of the Normal approximation defined above.

Chernoff Bound We use the Chernoff inequality to propose an lower bound on $P(X \geq \frac{T}{2})$, giving us:

$$P(X \geq \frac{T}{2}) \geq \begin{cases} 0 & \text{if } \alpha < \frac{1}{2} \\ 1 - e^{-\frac{T}{2\alpha}(\alpha - \frac{1}{2})^2} & \alpha \in \left[\frac{1}{2}, 1\right] \end{cases}$$

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 bound functions derived as a lower bound for our test error.



(a) The Binomial bound and it's gaussian approximation as we vary k

(b) The Binomial bound and it's gaussian approximation as we vary T

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

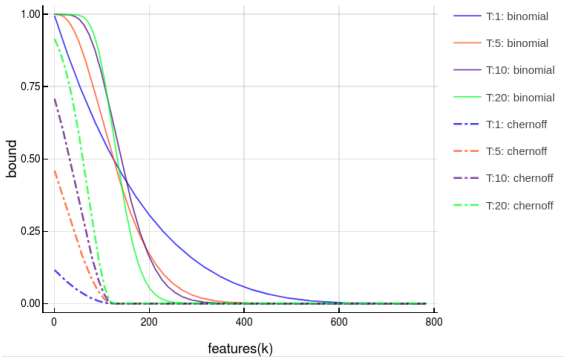
Given that d and b are constant, we observe that α is a non-increasing function of k. In Figure 1a, as we vary k, α varies, which causes the bound to change. In the gaussian approximation, $N(T\alpha, T\alpha(1-\alpha))$, we note that the mean of the gaussian goes from T when k is zero, to 0 when $k \geq d - b$. This would mean that $cdf(T/2)$, which denotes the cumulative distribution function varies from 0 to 1. Then $P(X \geq T/2) = 1 - cdf(T/2)$ varies from 1 to zero as we progress.

In Figure 1b, α is constant, rather T varies. In the gaussian approximations given by $N(T\alpha, T\alpha(1-\alpha))$. When $\alpha < \frac{1}{2}$, the mean of the gaussian is less than $\frac{T}{2}$, and hence $cdf(T/2)$ increases to 1 as we increase T. Similarly, $\alpha > \frac{1}{2}$ that $cdf(T/2)$ decreases to 0 as T increases, thus explaining the trends.

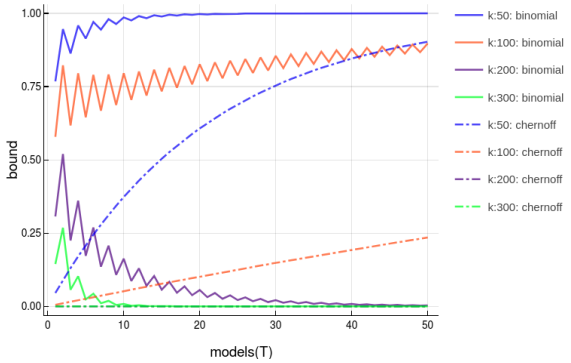
In Figure 2a, all the chernoff bounds drop to zero after some time as in this range $\alpha < \frac{1}{2}$. Then, as α moves to 0 as k increases. The bound becomes more reliable as T which is a constant is increased.

In Figure 2b, We can clearly see that our lower bounds are completely weak.

Overall, taking into consideration the relative error of the bounds proposed, and the numerical computation required to calculate them, the Gaussian approximation seems to be best suited to our use.



(a) The Binomial bound and it's chernoff bound as we vary k



(b) The Binomial bound and it's chernoff bound as we vary T

Figure 2: A Plot of two bounds as a function of T and k, for the MNIST dataset with $b = 4$, and $d = 784$

We now present a theorem that certifies the accuracy of the ensemble under poisoning attacks constrained on the number of features used in the backdoor.

Theorem 4.1. *Given a training dataset $\mathcal{D}$ with N samples and d features, we construct an ensemble E, consisting of T models. Each model in the ensemble is trained on all the samples in $\mathcal{D}$, using a random subset of the feature set of size k. We assume that the adversary is allowed to manipulate at most b features in the backdoor poisoned samples, and the assumptions listed in Section 4.1 are true.*

If $\alpha > \frac{1}{2}$ and $T \geq -\log(\epsilon)\frac{2\alpha}{\left(\alpha-\frac{1}{2}\right)^2}$, then the probability of E classifying test time examples correctly is at least $1-\epsilon$, for ϵ, a small error probability.

Note that the above theorem assumes that the classifiers have 100% accuracy. However, in real word, this is rarely the case. We propose another theorem, where each model i in the ensemble has an accuracy γ_i.

Theorem 4.2. *Given a training dataset $\mathcal{D}$ with N samples and d features, we construct an ensemble E, consisting of T models. Each model in the ensemble is trained on all the samples in $\mathcal{D}$, using a random subset of the feature set of size k, and achieves an accuracy of γ_i on clean test samples. We assume that the adversary is allowed to manipulate at most b features in the backdoor poisoned samples, and the assumptions listed in Section 4.1 are true.*

Define $\mu = \alpha\left(\sum_{i=1}^{T}\gamma_i\right)$.

If $\alpha > \frac{T}{2\left(\sum_{i=1}^{T}\gamma_i\right)}$ and $T \leq 2\left(\mu - \sqrt{-2\mu\log(\epsilon)}\right)$, then the probability of E classifying test time examples correctly is at least $1-\epsilon$, for ϵ, a small error probability.

We provide proofs of the statements in the appendix.

5 Extending Theory to give better bounds

Note that we have not yet used the fact that the number of poisoning points $p << n$, where n is the number of poisoning points. We also introduce some extra notation here. Let each model i of the ensemble be denoted by M_i. Here, we split the dataset $\mathcal{D}'$ into T parts, each denoted by $\mathcal{D}_i$, with $|\mathcal{D}_i| = n_i$, and the number of poisoning points in each partition by p_i. These datasets are pairwise disjoint from one another. Now, we train each model M_i on each dataset $\mathcal{D}_i$ instead of the whole dataset $\mathcal{D}'$.

5.1 Bringing poisoning percentage into the picture

Given our setup above, we have:

$$\sum_{i=1}^{T} p_i = p \qquad \text{with } p_i \geq 0 \ \forall i \tag{1}$$

Note that this famous diophantine equation has $\binom{p+T-1}{T-1}$ solutions. When we constrain the p_i's to be strictly greater than 0, equation 1 has $\binom{p-1}{T-1}$ solutions. We further analyse the case when some of these p_i's are zero.

Let z of these p_i's be zero. Note that we did not specify which partitions contain no poisoning points. Hence, in this case, the number of solutions to equation 1 has $\binom{T}{z}\binom{p-1}{T-1-z}$. This is explained as follows: we first choose z quantities out of T to be zero, and now equation 1 becomes equivalent to:

$$\sum_{i=1}^{T-z} y_i = p \qquad \text{with } y_i > 0 \ \forall i \tag{2}$$

We apply the strict version in our case because we have already fixed which quantities we want to be zero. Now, 2 has $\binom{p-1}{T-1-z}$ solutions, thus giving us the counting result. We can define a random variable Z, which takes value z when z partitions don't contain any poisoning points(hereby referred to as "clean" partitions). As a natural extension, we have:

$$P(Z=z) = \frac{\binom{T}{z}\binom{p-1}{T-1-z}}{\binom{p+T-1}{T-1}} \tag{3}$$

5.2 Combining it with analysis done in Section 4.3

Now, we know that the probability of the backdoor intersecting with a feature set K_i is given by

$$\alpha = \frac{\binom{d-b}{k}}{\binom{f}{k}}$$

and that we need to find the value $P(\sum_{i=1}^{T} X_i \geq \frac{T}{2})$. Let $X = \sum_{i=1}^{T} X_i$

5.2.1 How does equation 3 change things here?

Let us say, out of T partitioned datasets, z of them are already clean. We make an assumption here that having a clean partition implies that the model is not poisoned, and will produce correct predictions. Now, instead of needing $P(\sum_{i=1}^{T} X_i \geq \frac{T}{2})$, we would only need $P(\sum_{i=1}^{T-z} X_i \geq \frac{T}{2} - z)$. An explanation for this is that out of the $T-z$ poisoned partitions, we would need $\frac{T}{2} - z$ models to be clean. This adds more power to our analysis, as we now require lesser than half the models to be feature-clean.

We repeat the analysis done earlier. Denote $X^z = \sum_{i=1}^{T-z} X_i$. Note $X^z \sim \text{Binomial}(T-z, \alpha)$. The binomial computation becomes:

$$P(X^z \geq T/2 - z | z) = \sum_{i=T/2-z}^{T-z} \binom{T}{i} \alpha^i (1-\alpha)^{T-i} \tag{4}$$

and the gaussian approximation to this binomial approximation becomes $\mathcal{N}((T-z)\alpha, (T-z)\alpha(1-\alpha))$

$$P(X^z \geq T/2 - z|z) = 1 - cdf(T/2 - z) \tag{5}$$

5.2.2 Getting a better estimate of probability of correct classification

Note that given random variables Z and X_z, we can define a probability distribution of two variables, defined by equations 3, and 4. Now, we use the law of conditional probability, to write down $P(X \geq T/2)$ as

$$P(X \geq T/2) = \sum_{z=0}^{T} P(Z = z)P(X^z > (T/2 - z)|Z = z) \tag{6}$$

Wherein we can use any approximation of $P(X^z > (T/2 - z)|Z = z)$ as we need.

6 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[6] 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 [7], and Spectral Signatures [8].

Badnets[9] 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 4.2.

6.1 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.

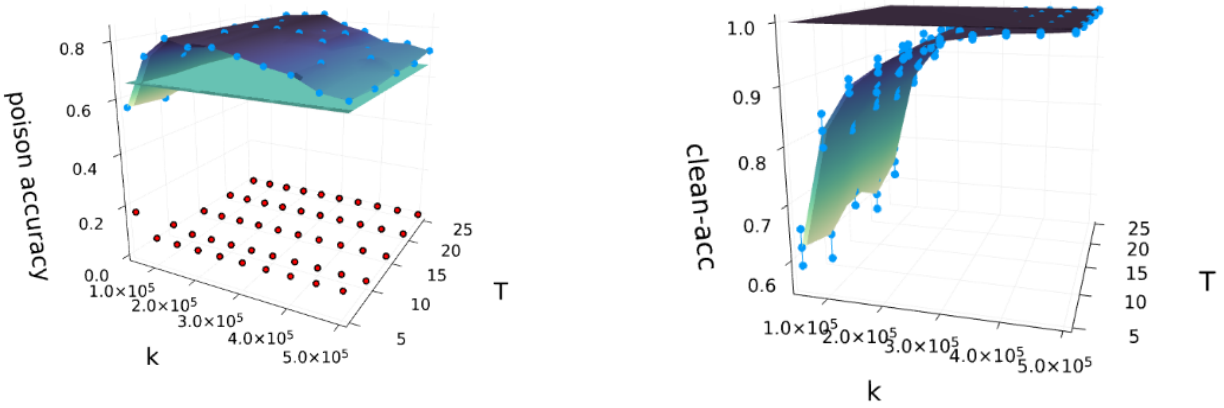
We make a few observations in Figure 3. When k is low, the ensemble fails to capture any meaningful features, and hence $Acc(E_p, \mathcal{D}_p)$ is low. When $k \approx d$, a majority of models in the ensemble have intersections with the backdoor features, thus bringing $\text{Acc}(E_p, \mathcal{D}_p) \approx \text{Acc}(E_p, \mathcal{D}_p)$. 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.

6.2 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.

6.3 MNIST/Explanation based backdoors

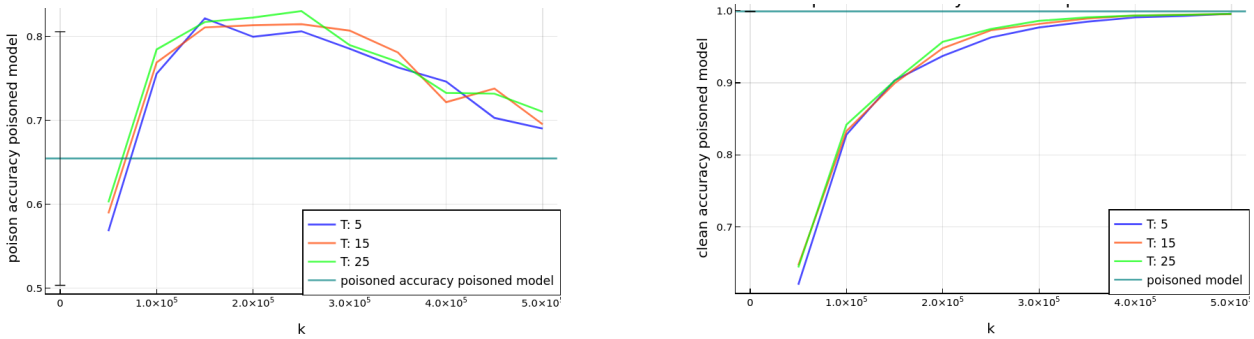
We use a binary MNIST dataset with just two classes 0 and 1. Here we use the pixels themselves as individual features. On running the attacks, we see that $\text{Acc}(\mathcal{M}, \mathcal{D}_p)$ is around 40%, which indicates that the attack is not working as we



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

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

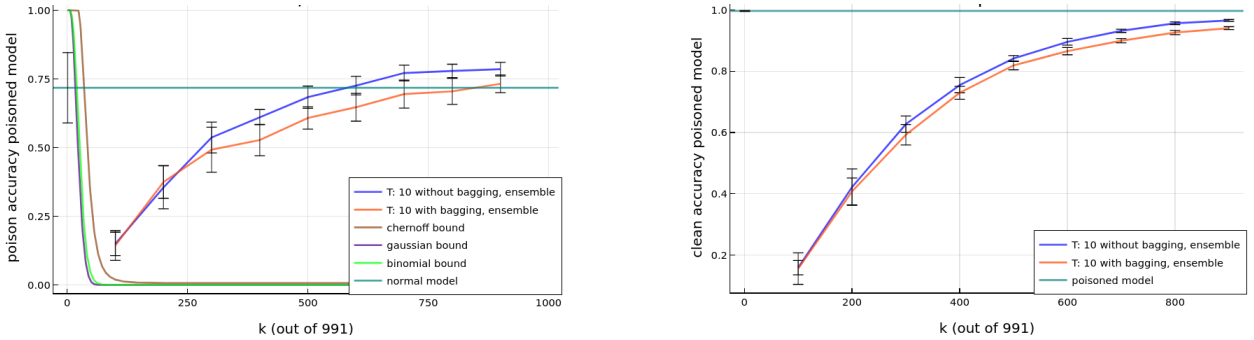
Figure 3: A plot of poison and clean accuracies of ensemble E compared against normal model $\mathcal{M}$ as a baseline, evaluated on the drebin dataset and explanation based backdoor attack. The red dots on the left figure show the chernoff bound. These results are obtained with a poisoning percentage of 1%, with 30 backdoors in the feature set.



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

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

Figure 4: Plots of clean and poison accuracies of ensemble E compared against normal model $\mathcal{M}$ on the drebin dataset, with the backdoor explanation attack. These results are obtained with a poisoning percentage of 1%, with 30 backdoors in the feature set.



(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 5: 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. These results are obtained with a poisoning percentage of 1%, with 30 backdoors in the feature set

expected to. From image 6, 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.

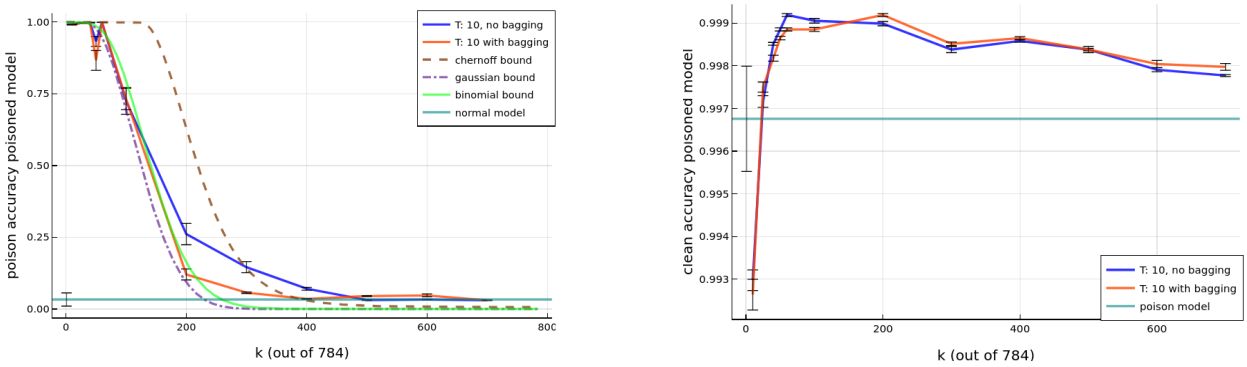
(a) Unpoisoned MNIST image from class 1

(b) Unpoisoned MNIST image from class 1

Figure 6: A comparison of MNIST Images generated from the explanation based backdoor attack

6.4 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 $\text{Acc}(\mathcal{M}_p, T_p)$ under 10%. A surprising result is that even when choosing $k = 10$, we are able to get high values(>99%) of $\text{Acc}(E_p, \mathcal{T}_p)$, with very loss in $\text{Acc}(E_p, \mathcal{T})$ (about 99%). Increasing T leads to results with less variance.



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

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

Figure 7: 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. These results are obtained with a poisoning percentage of 1%, with 4 backdoors in the feature set

7 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.

- 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?

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

[2] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking, 2021.

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

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

[5] Ambra Demontis, Marco Melis, Battista Biggio, Davide Maiorca, Daniel Arp, Konrad Rieck, Igino Corona, Giorgio Giacinto, and Fabio Roli. Yes, machine learning can be more secure! a case study on android malware detection, 2017.

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

[7] 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.

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

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

9 Proofs

9.1 Proof of Theorem 4.1

We transform our inequality on T as follows: If $\alpha > \frac{1}{2}$ and $T \geq -\log(\epsilon)\frac{2\alpha}{\left(\alpha-\frac{1}{2}\right)^2}$

$$T \geq -\log(\epsilon)\frac{2\alpha}{\left(\alpha-\frac{1}{2}\right)^2}$$

$$-T\frac{\left(\alpha-\frac{1}{2}\right)^2}{2\alpha} \leq \log(\epsilon)$$

$$1 - e^{-T\frac{\left(\alpha-\frac{1}{2}\right)^2}{2\alpha}} \geq 1-\epsilon$$

Note, by the chernoff bound we propose (When $\alpha > \frac{1}{2}$), we get

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

$P(X > T/2)$ is the probability of the classifier E predicting test samples correctly because of the assumptions listed in the theorem. Hence, we are done.

9.2 Proof of Theorem 4.2

Note that we have our chernoff bound as follows, given a $\beta \geq 0$, $\mu = \alpha(\sum_{i=1}^{T} \gamma_i)$

$$P(X \leq (1-\beta)\,\mu) \leq \exp\left(-\frac{\beta^2}{2}\mu\right)$$

.

Now, we need

$$(1-\beta)\mu = \frac{T}{2}$$

Giving us

$$\beta = 1 - \frac{T}{2\mu} \geq 0$$

The inequality above gives us two important results,

$$\mu \geq \frac{T}{2}$$

and

$$\alpha \geq \frac{T}{2\sum_{i=1}^{T} \gamma_i}$$

Using the value of β so obtained, we get the bound as follows:

$$P(X \leq \frac{T}{2}) \leq \exp \frac{-(\mu - \frac{T}{2})^2}{2\mu}$$

Thus giving

$$P(X \geq \frac{T}{2}) \geq 1 - \exp \frac{-(\mu - \frac{T}{2})^2}{2\mu}$$

For the model to have atmost ϵ error, we need

$$P(X \geq \frac{T}{2}) \geq 1 - \exp \frac{-(\mu - \frac{T}{2})^2}{2\mu} \geq 1 - \epsilon$$

Simplyfing, we get the inequalities:

$$\begin{aligned}
\exp \frac{-(\mu - \frac{T}{2})^2}{2\mu} &\leq \epsilon \\
\frac{-(\mu - \frac{T}{2})^2}{2\mu} &\leq \log(\epsilon) \\
(\mu - \frac{T}{2})^2 &\geq -2\mu \log(\epsilon) \\
\mu - \frac{T}{2} &\geq \sqrt{-2\mu \log(\epsilon)} \\
2(\mu - \sqrt{-2\mu \log(\epsilon)}) &\geq T
\end{aligned}$$

Note, taking on taking square roots to both side, the LHS expanded to $\mu - \frac{T}{2}$, and not the other way around since, $\mu \geq \frac{T}{2}$ (described above)

10 Appendix

10.1 Problem

We are given a dataset with n data points, each having d features, out of which b are backdoored. There are T models labeled from $M_1, \ldots M_T$ given to us, which are trained on k features chosen at random from the dataset.

10.2 Analysis

Given a single model M_i, let E_i be the event that there is no intersection in between the k features and any backdoor in the dataset. Let X_i be a random variable defined as follows.

$$X_i = \begin{cases} 1 & \text{if } E_i \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Clearly, we have

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

Let X be a random variable that is defined as the number of successes of E_i over all is. Verbosely, it denotes the number of models (out of T) that have no intersection with a backdoor. Trivially, X can be seen as the sum of T independent Bernoulli variables X_i.

Now, we require $P[X \geq \frac{T}{2}]$.

10.2.1 Chernoff Bound based analysis

One way to approach this is through the Chernoff's bound, For definiton and corollaries, please refer to Corollary 13.3 linked. Here is a derivation of the bound in our case. From the chernoff bound, we get this corollary:

Given $X_1, \ldots X_n$ with $X = \sum_{i=1}^{n} X_i$ be independent RV, not necessarily from the same distribution, and some real number $\epsilon \in [0, 1]$. Setting $\mu = E[\sum_{i=1}^{n} X_i]$, we have:

$$P(X \leq (1 - \epsilon)\,\mu) \leq \exp\left(-\frac{\epsilon^2}{2}\mu\right)$$

.

In our case, with $X_i s$ being i.i.d Bern(α), $\mu = T\alpha$. We require $P[X \geq \frac{T}{2}]$, so $(1-\epsilon)\,\mu = \frac{T}{2}$. thus giving the equations:

$$(1-\epsilon)\,T\alpha = \frac{T}{2}$$

Implying

$$\epsilon = 1 - \frac{1}{2\alpha}$$

Note that we also have the constraint $\epsilon \geq 0$, hence giving us:

$$1 \geq \frac{1}{2\alpha} \implies \alpha \geq \frac{1}{2}$$

The other constraint $\epsilon \leq 1$ is satisfied automatically since $\alpha > 0$:

$$\epsilon = 1 - \frac{1}{2\alpha} < 1$$

Substituting back into the bound, we get:

$$P[X \leq \frac{T}{2}] \leq e^{-\frac{T\alpha}{2}(1-\frac{1}{2\alpha})^2}$$

$$P[X \leq \frac{T}{2}] \leq e^{-\frac{T}{2\alpha}(\alpha-\frac{1}{2})^2}$$

Consequently, we get:

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

and

$$P[X \geq \frac{T}{2}] \geq 1 - e^{-\frac{T}{2\alpha}(\alpha-\frac{1}{2})^2} + P(X = \frac{T}{2})$$

The latter term in the above equation is 0, if T is odd. In all the following cases, we choose T to be odd. Note that this inequality holds only when $\alpha \geq \frac{1}{2}$. When $\alpha = \frac{1}{2}$, the bound is 0. As, a natural extension, we propose the lower bound of 0 when $\alpha < \frac{1}{2}$. In total, the bound becomes:

$$P[X \geq \frac{T}{2}] \geq \begin{cases} 1 - e^{-\frac{T}{2\alpha}(\alpha-\frac{1}{2})^2} & \alpha \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

10.2.2 Gaussian approximation to the Binomial distribution

Given a binomial distribuion $Binomial(n, p)$ with n trials, it can be approximated with normal distribution $\mathcal{N}(np, np(1-p))$, under the conditions that either $p \approx 1/2$ or n is large. So, in our case, the approximating normal distribution is given by $\mathcal{N}(T\alpha, T\alpha(1-\alpha))$. We approximate the quantity we need as the follows

$$P(X \geq T/2) \approx 1 - cdf(T/2)$$

where cdf returns the cumulative distribution function of the Normal approximation defined above. Even if this approximation might not be exact in some situations, it's calculation is much simpler than the exact numerical computation.

10.2.3 Exact numerical computation

Since we need to evaluate $P(X > T/2)$, we directly compute it as follows:

$$P(X \geq T/2) = \sum_{i=T/2}^{T} \binom{T}{i} \alpha^i (1-\alpha)^{T-i}$$

NOTE: This method can result in non-exact answers (due to floating point approximations) when $\alpha << 1$. It can also be computationally expensive, since it requires the calculation of $\binom{T}{i} \forall i \in [T/2, T]$