An investigation into the effect of using Data Poisoning Models and PGDAttack during Adversarial Training

Suya, Fnu
fs5xz@virginia.edu

Villca-Rocha, Andrew
av3dj@virginia.edu

Ma, Rui
rm9ke@virginia.edu

Liu, Jerry Y
jyl3xf@virginia.edu

August 2020

1 Introduction

In this investigation, we explored the effects of using two popular machine learning attacks, Data Poisoning Models and PGDAttack, in adversarial training. We wish to discover if the two attacks will amplify each other when adversarially training the model or if they will hinder each other. Also, we attempt to find whether an adversarially trained model using PGDAttack will generally increase its accuracy against perturbed (PGDAttack) MNIST 1/7 images.

2 Preliminaries

2.1 Linear SVM

The model attacked in this investigation is a non-kernel linear SVM model implemented in Tensorflow. The loss function used is the Hinge Loss. Training is then conducted using Batch Gradient Descent optimized using the Gradient-DescentOptimizer from Tensorflow.

2.2 MNIST 1/7

The MNIST 1/7 dataset is used in this investigation. This data consists of the regular MNIST dataset but strictly using the 1 and 7 labels. A binary classification dataset is used to more easily identify the effects of the attack.

2.3 Data Poisoning Attack

One attack used is the Data Poisoning Attack. This consists of poisoning a subset of the data set and then appending this poisoned subset into the original data set. This poisoning's objective is to maximize the loss of the model by choosing an input and label which will maximize the loss function and update the loss function in the direction of this "attack point". The poisoned data set produced will be used in the later stage of implementing a dual-vector adversarialy trained model.

2.4 PGDAttack

This attack involves perturbing a given image in the direction that will result in maximizing the loss, causing the model to misclassify the image. These perturbations are usually small which would result in easier evasion of defenses.

2.5 Adversarial Training

Adversarial Training is used in the testing to evaluate the robustness of the model. In this investigation, we will use two forms of adversarial training. One is a single-vector and the other is dual-vector.

The single-vector adversarial training steps are as follows. At each batch gradient descent step, the batch is perturbed using the PGDAttack and then the linear SVM model is trained on this perturbed batch. The result should be a model that is more resilient against PGDAttacks.

The dual-vector adversarial training steps is the same as the single-vector except the MNIST 1/7 dataset is replaced with a mixed version, which includes poisoned points produced by the Data Poisoning Attack.

3 Test Plan

Here we will discuss the testing environment and what measurements were taken. A test is conducted for varying PGDAttack epsilon values.

3.1 Models

A linear SVM was used in each test but were trained differently. The "Normal SVM" is the SVM model trained on the MNIST 1/7 dataset. The "Normal SVM w/ Mixed" is the SVM model trained on a poisoned MNIST 1/7 dataset using the Data Poisoning Attack. The "Adv SVM w/ Normal" is a single-vector adversarially trained model. Finally, the "Adv SVM w/ Mixed" is the double-vector adversarially trained model.

3.2 Measurements

The measurements taken in each test are the clean accuracy and robust accuracy. The clean accuracy is the resulting model's performance in correctly classifying 1/7 images. The robust accuracy is the model's performance in correctly classifying perturbed (PGDAttack) 1/7 images.

4 Results

The tables below show the results of the testing. The PGDAttack used in testing had the following parameters: Learning Rate = 0.02 and Iteration Steps = 250. The table shows the results for varying epsilons which is the limiter to how much a pixel can change for each image.

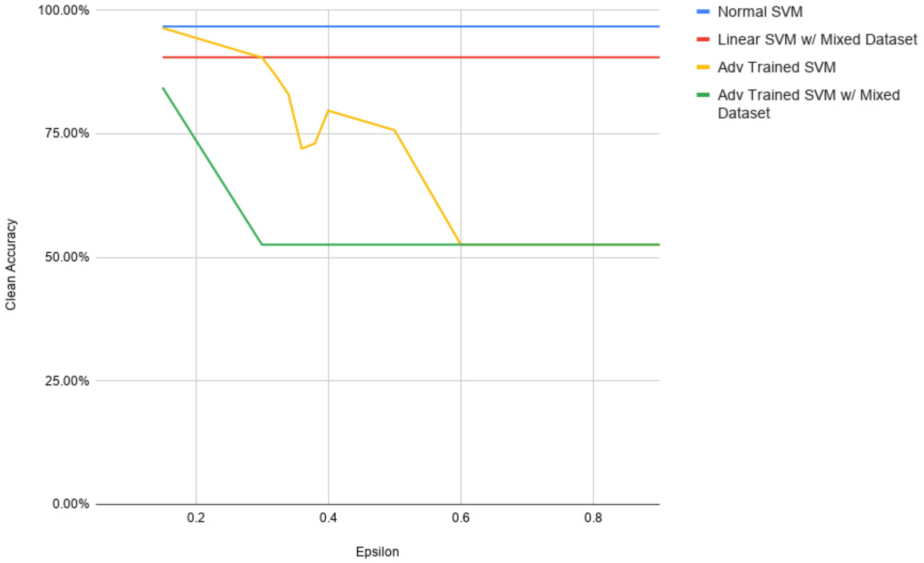

Figure 1: Clean Accuracy vs Epsilon (of PGDAttack) for each model

5 Analysis

5.1 Testing Analysis

From Figure 1. we can see that the combination of the PGDAttack and the Data Poisoning Attack in adversarilly training magnifies the detrimental effect on the model's clean accuracy. The model's clean accuracy collapses quickly

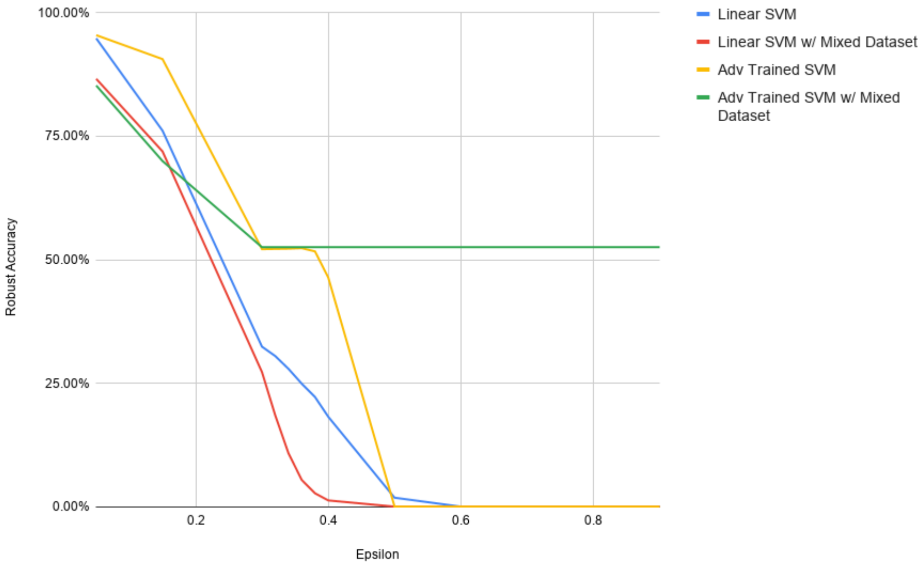

Figure 2: Robust Accuracy vs Epsilon (of PGDAttack) for each model

when compared to one adversarially trained with PGDAttack only, reaching a flat-line at 52.55%.

From Figure 2. we notice that the non-adversarilly trained models plummet to 0%. Even a Normal SVM trained on a Poisoned Dataset does little to combat the PGDAttack itself. The model trained adversarially with PGDAttack initially performs well (epsilon less than 0.4). But once the epsilon values reach enormous sizes, its performance plummets to that of the non-adversarially trained models. The adversarially trained SVM with the poisoned dataset reaches the same valley that is seen in the clean accuracy. This makes it perform better than the adversarially trained SVM on epsilon values greater than 0.4.

An initial hypothesis to explain the sudden collapse in the single-vector adversarially trained model is the high epsilon PGDAttack destroys the linear seperation between the 1's and 7's. Thus, the linear SVM model can no longer learn from the dataset due to its non-linearity. We will test this by inspecting the robust loss over training iterations and by providing some statistics that describe the relationship between 1s data and 7s data.

5.2 Hypothesis Analysis

Figure 3 shows the robust loss of the single-vector adversarially trained model. Here, we see oscillating trends in the robust loss function, and on average, the loss does not decrease by a significant amount. This confirms that the model

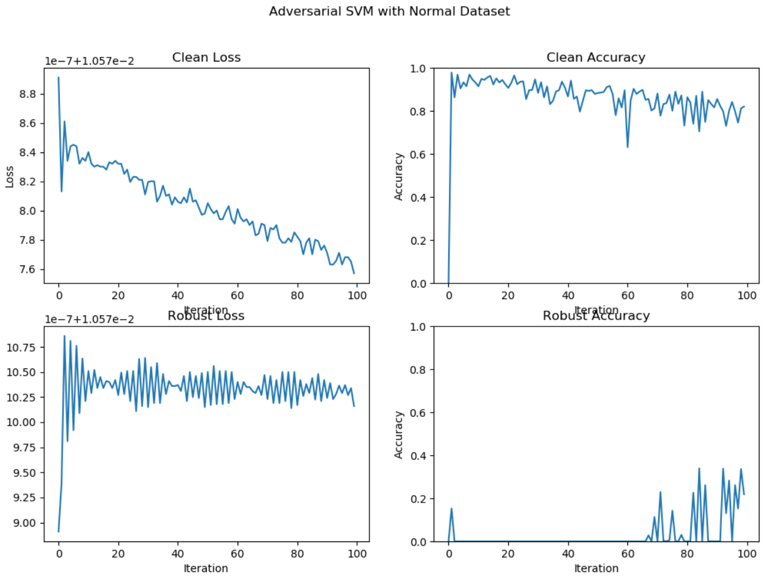

Figure 3: A single-vector adversarially trained model with epsilon 0.5

Table 1: Average Percentage of Pixels within 0.5 of different class

Image Types	Percentage of pixels within 0.5 of different class
Normal 1s and 7s	86.29%
Perturbed 1s and 7s	100.0%

can no longer learn from the data set leading to a collapse in accuracy.

A justification for why it can't learn can be explained with Table 1. Here we sampled the average percentage of pixels from perturbed 1s and perturbed 7s that are within 0.5 of each other. We picked 0.5 because this is an epsilon value where we experienced the collapse. We can see that nearly 100% of the pixels are indeed within 0.5 of each other. In comparison, a clean MNIST 1/7 dataset has 86% of the pixels within 0.5 of each other. Additionally in Table 2, we took the average L2 norm of the distance between every perturbed 1 and perturbed 7. This resulted in a l2 norm of 4.2. In comparison, the average l2 norm for every normal 1 and normal 7 is 9.2. This shows a dramatic decrease in the distance between two.

6 Conclusion

In this write-up we explored the effect of the dual-vector adversarial training on the accuracy and robustness of the model. We found that the dual-vector

Table 2: Average Distance (L2 Norm) between 1s and 7s

Image Types	Average distance between pixels of different class
Normal 1s and 7s	9.34
Perturbed 1s and 7s	4.58

adversarial training (PGDAttack and Data Poisoning) further decreased the model's clean accuracy. However, in the robustness of the model we found that using the dual-vector adversarial training did not drastically improve its robustness. It did improve the normal models, but did not exceed the model adversarially trained with PGDAttack only. Furthermore, we found that the PGDAttack adversarially trained model fails to learn at epsilons greater than 0.4, which we argue is due to the destruction of the linear seperation in the dataset. In the future, we should conduct more testing to identify the weird flat-line behavior in both the clean accuracy and robust accuracy. This may be a byproduct of using a simple linear SVM model. So, further testing should be conducted on more complex models to achieve more concrete results.