Cybersecurity project work

Anna Fabris

Experimenter Attitude

Abstract

This report describes the project work done on the "Poisoning the Unlabeled Dataset of Semi-Supervised Learning" paper [1].

The paper investigate a new class of attacks that could be done to a Semi-supervised learning model and what are the possible defenses. Semi-supervised learning is particularly vulnerable to specif types of attacks as the training data can be changed without human supervision.

Following the review of the paper I built a semi-supervised learning model that recognizes the digits in the MNIST database and implemented various attacks.

Introduction

In recent years there has been an unparalleled surge of interest in the topic of machine learning and the performance of models has been steadily increasing. This can be partly attributed to the availability of enormous labeled data sets that allowed developers to train ever larger models. Collecting labeled data for a learning problem often requires the use of a skilled human agent. As a result, the cost of labeling may make it impossible for large, fully labeled training datasets to exist. For this reason, semi-supervised learning can be used to reduce the demand of machine learning models for labeled data.

Semi-supervised learning is a type of learning that falls between supervised and unsupervised learning. It uses a small amount of labeled data and a large amount of unlabeled data, which can yield a significant boost in learning accuracy. The latest models are a few percentage points better than fully-supervised training, but require 100% less labeled input [1].

1 Attacks

Semi-supervised learning models are vulnerable to different types of attacks compared to fully supervised models. Datasets are built by gathering information from anonymous, unverified web sources without regard to quality to get as much data as possible. The lack of human supervision over this process exposes security risks as training data can be modified to influence the behavior of learned models. The only thing the attacker has to do is to upload an image on the internet and it will be scanned and included in the dataset.

There are two typical attack objectives:

- **indiscriminate poisoning attack** aimed at lowering the accuracy of the classifier
- **targeted poisoning** intended to cause a certain misprediction in a specific case

In this project work the second type of attack will be performed. Given the appropriate decision boundary, we want the model to learn a slightly imprecise decision boundary so that the model still remains essentially accurate, but a chosen imprecision is classified erroneously. The attack presented is called Interpolation Consistency Poisoning and consist in inserting a chain of poisoned examples connecting the two classes (rather than simply adding one poisoned example). In this way the nearest neighbors will be labeled the same manner, and be misclassified.

The paper tested the attack on three datasets (CIFAR-10, SVHN and STL-10), using the three most accurate Semi-Supervised techniques (MixMatch, UDA and FixMatch) on the ResNet-28 model. For each experimental trial 8 attack were attempted, on average the attack has a 91% success rate when poisoning 0.5% of the unlabeled dataset.

2 Defenses

While there are many protections against indiscriminate poisoning attacks, there are far fewer defenses against targeted poisoning attacks.

To prevent this kind of attack human inspection could be added. Paying a human to manually label the dataset would be too much (if this were acceptable, the entire dataset would be labeled), but human inspections could be used to quickly filter out obviously wrong samples.

Another alternative is to utilize Agglomerative Clustering. This generates clusters of similar examples, routinely merging the two sets with the minimum distance to form a single set until the minimum distance exceeds a certain threshold. Since our poisoned examples are all similar to each other in pixel space, it is likely that they are all located in the same cluster. As a result, by deleting the largest cluster, we can entirely prevent this attack. This can only work if the defender is able to build a useful distance function, which the attacker can easily avoid.

To avoid this we can monitor the training dynamics. It can be assumed that numerous unlabeled instances will impact benign examples at the same time, while our poisoned instances, are constructed to primarily influence the prediction of the other poisoned examples. We can thus eliminate examples that are influenced by only a few other examples.

3 Experiments

Several experiments have been conducted to determine what is the best strategy for poisoning a semi-supervised model and how much poisoned data is needed.

The project has been published and is available on GitHub at the link: https://github.com/annafabris/Poisoning-unlabeled-Dataset-for-Semi-Supervised-Learning

All the following results were obtained after 10 epoch of training, it would clearly be preferable to run each experiment for more epoch and average the results of a few trials, but it wasn't possible as the model is slow to train.

Dataset

The dataset used to test the attack is the MNIST database of handwritten digits. It has a training set of 60,000 examples, and a test set of 10,000 examples [3]. A small sample of which can be seen on Figure 1.

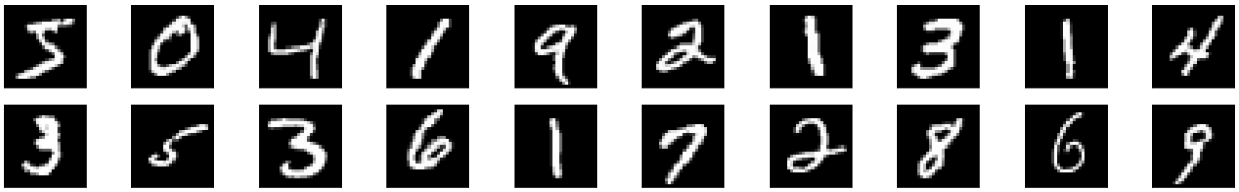

Figure 1: An example of MNIST test data

As MNIST was intended for fully-supervised training, it is used in semi-supervised learning by deleting the labels from all but a few examples.

Model

The model chosen for the task is a Ladder Network and was implemented from [4]. The first test on the model was to run it using all the possible labeled dataset (as if it was a fully supervised model). The supervised learning model obtains a test accuracy of 98.88%. This is far from being state-of-the-art as near-human performance is possible (0.23% test error rate) [2]. This should only make our model more resistant to poisoning attacks as better model would just be easier to poison [1].

For the following test only 100 labels were kept. This is the semi-supervised learning model that will be poisoned later. From now on referred to as non-poisoned model and it obtained a test accuracy of 95.46%

Figure 2 and 3 represent the confusion matrix of the two model described. On the x-axis are the predicted labels, on the y-axis the true labels.

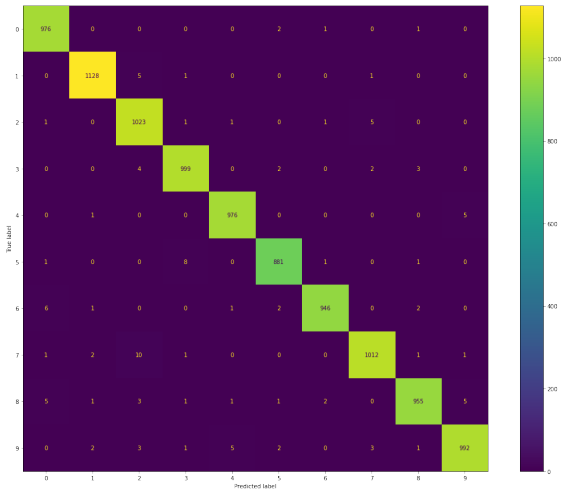

Figure 2: Supervised learning

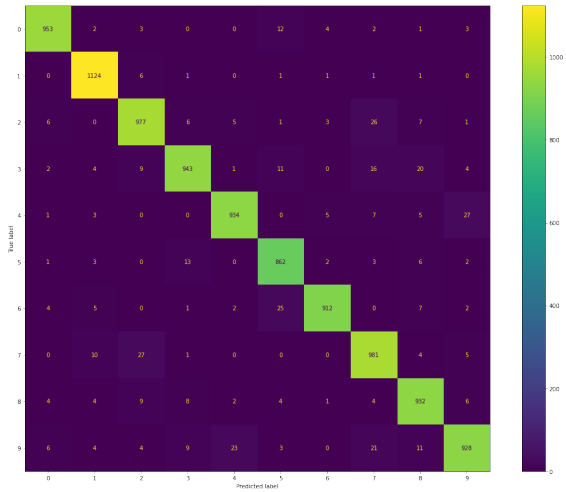

Figure 3: Non-poisoned model

Attacks

The following attacks attempt to misclassify 4s and 9s. To achieve this, the data was poisoned by introducing the interpolation of different pairs of 4-9's in the data. The interpolations were created using two different techniques, following [5]: latent space interpolation and image space interpolation.

Latent space interpolation

The first technique used to generate the interpolation was latent space interpolation, an example of an interpolation with this technique is shown on Figure 4.

Figure 4: Latent space interpolation between 4 and 9

I generated 86 interpolation of 21 images each for a total of around 3% of the dataset. The model was trained in exactly the same way as the non-poisoned model.

The test accuracy of the model is 92.90%, which is somewhat worse than the non-poisoned model. However, as seen in Figure 3, 9.3% of 4s were misclassified as 9s and 0.9% of 9s were misclassified as 4s. It is worth pointing out that this attack led to slightly worse performance of the model for other pairs, for example 6.9% of 7s were misclassified as 9. This is most likely due to the similarities between some 7s and the interpolation images.

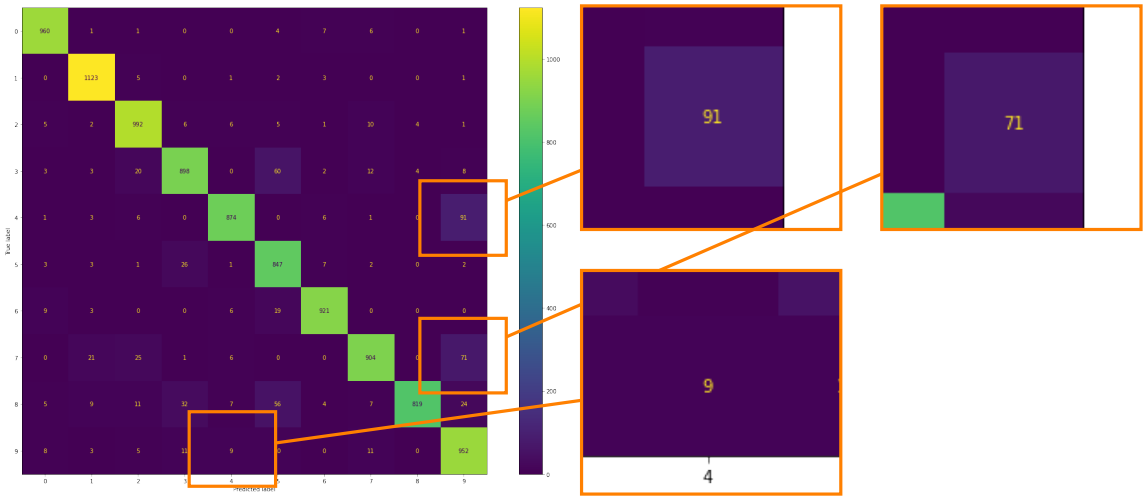

Figure 5: Latent space interpolation confusion matrix

Image space interpolation

The same experiments were then repeated using image space interpolation, an example of an interpolation with this technique is shown on Figure 6.

Figure 6: Image space interpolation between 4 and 9

I generated the same number of poisoned data, for a total of around 3% of the dataset. The model was trained in exactly the same way as the non-poisoned model.

The test accuracy of the model is 89.04%, which is significantly worse than the non-poisoned model. As seen in Figure 3, 47.6% of 4s were misclassified as 9s and 11.8% of 9s were misclassified as 4s. Compared to the latent space interpolation model the effect on other pairs is smaller, with

only 3.4% of 7s misclassified as 9. This is most likely because the interpolation images don't show much similarity with other numbers.

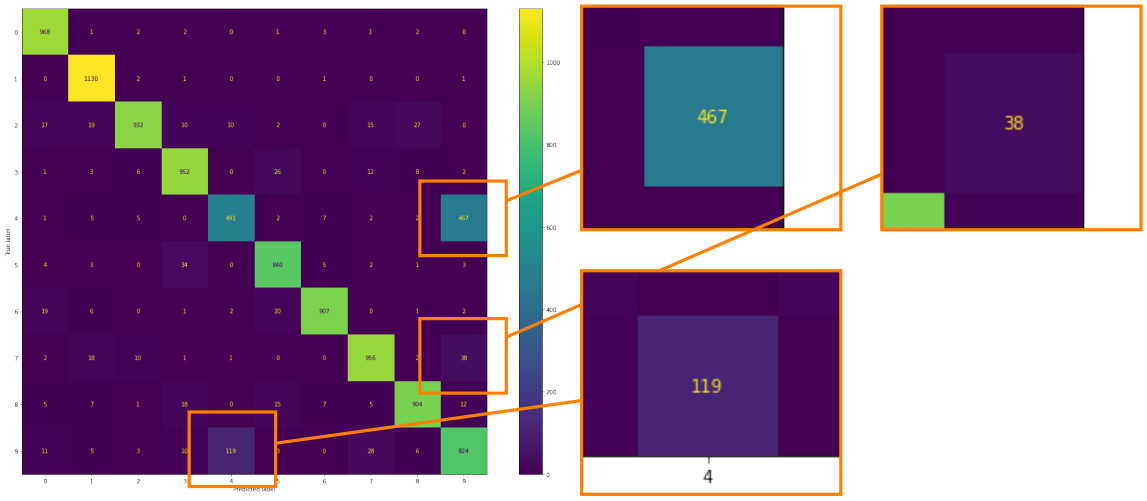

Figure 7: Image space interpolation confusion matrix

The experiment was repeated with all the same conditions, but decreasing the poisoned data from 3% to 1%. The test accuracy of the model is 94.21% and there are very few mistakes.

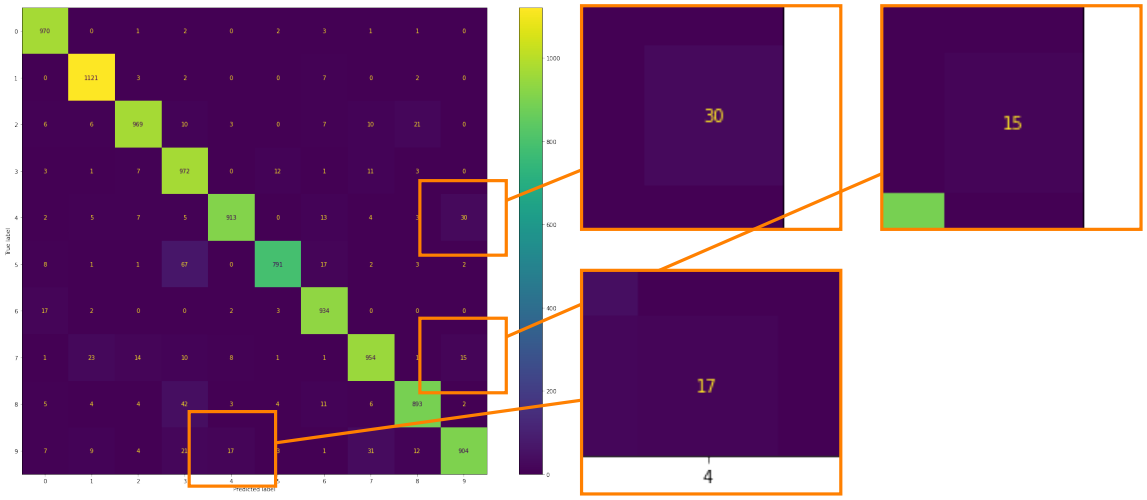

Figure 8: The National Gallery of Canada

The following table summarize the results.

	Test Accuracy	**9s classified as 4s**	**4s classified as 9s**
Supervised model	98.88 %	0.4 %	0.5 %
Non-poisoned model	95.46 %	2.3 %	2.7 %
Latent space model 3%	92.90 %	0.9 %	9.3 %
Image space model 3%	89.04 %	11.8 %	47.6 %
Image space model 1%	94.21 %	1.7 %	3.1 %

Here it can be seen that the image space interpolation attack is more effective than the latent space interpolation one. This result was expected as the path taken between the 4s and 9s is less direct. It is also critical to note how latent space interpolation poisoning is more difficult to detect.

It is disappointing how the attack that uses only 1% of poisoned data does not give great results as this is the threshold generally used in poisoning research.

There are many reasons why this might be the case. As there is a clear relationship between the poisoning success rate and the model accuracy, the performance of the attack could be increased by training the model more or using an improved model. Another hyper-parameters that influence the results of the attack is the density of poisoning example and I could only test it in a very limited manner.

Conclusion

Even given the constraint in which I was able to carry out the experiments, the results show that it is not too difficult to poison a semi-supervised learning model.

On top of that, more accurate semi-supervised learning methods are more vulnerable to poisoning attacks [1]. Thus the problem will only get worse as future techniques are going to be even better. As a result it will not be possible to just take all available unlabeled data and feed it into a classifier.

It will be necessary to figure out what can be done to continue using larger and larger unlabeled datasets. While there are techniques to prevent the specif attack seen here, how to address this issue in general for semi-supervised learning is something that will need to be studied in the coming years.

References

[1] Nicholas Carlini. "Poisoning the Unlabeled Dataset of Semi-Supervised Learning". In: 2021.

[2] Dan Cireşan, Ueli Meier, and Juergen Schmidhuber. *Multi-column Deep Neural Networks for Image Classification*. 2012. arXiv: 1202.2745 [cs.CV].

[3] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[4] Antti Rasmus et al. *Semi-Supervised Learning with Ladder Networks*. 2015. arXiv: 1507.02672 [cs.NE].

[5] Noufal Samsudin. *Latent space interpolation of images using Keras and Tensorflow.js*. https://medium.com/@noufalsamsudin/latent-space-interpolation-of-images-using-keras-and-tensorflow-js-7e35bec01c5a, accessed 1 January 2022. 2019.