Backdoor Attack Detector

Team Members

1) Shashank Shekhar (ss13956)
2) Gayatri Kalidindi (gk2205)
3) Sai Kiran (vc2118)

Introduction

In Lab 3 , we had implemented pruning to detect backdoored images but the results were not encouraging as we observed that the attack success rate did not decrease as we expected. This was majorly because in pruning we remove part of the network not used in correct prediction of clean images, but it seems that the attack targets part of the network which was being used to predict the correct inputs. SO, now we implement a different method which overcomes the shortcomings of pruning.

Methodology

STRIP(STrong Intentional Perturbation)
In Backdoored attacks, the attacker targets a particular class and all backdoored images are classified as that particular class. We use this characteristic of the attack to design our defense. We superimpose the image under test with multiple clean images and then test the variation in classification. This variation is measured using entropy. In case of poisonous images, the variation in predicted class of the superimposed images is less and hence they have a lower entropy whereas the same is higher for clean images. The next task is deciding the entropy threshold(detection boundary) to classify images as poisoned. This is calculated using the percentile of the entropy distribution of the clean images as in a practical scenario, only the clean images are available to the defender. Another important consideration is the detection metrics FRR(False Rejection Rate) and FAR(False Acceptance Rate). FRR is the probability of predicting a benign input as poisoned input and FAR is the probability of predicting a poisoned input as a benign input. We alter the value of FRR in order to achieve a lower FAR as having a lower FAR is more critical for our implementation.

Implementation

1. We first read the data from the clean and poisonous data sets.
2. Then for each dataset, we do the following steps
    1. Calculate clean entropy and poisoned entropy and plot it.
       For calculating entropy,we superimpose 100 images from the clean set for each image for which the entropy is being calculated.

2. We then calculate the threshold(detection boundary) which will be used to classify between clean and poisonous inputs.
3. We plot the entropies of clean and poisonous data and find that the entropy in case of poisonous data is low as compared to clean data. Below is an example

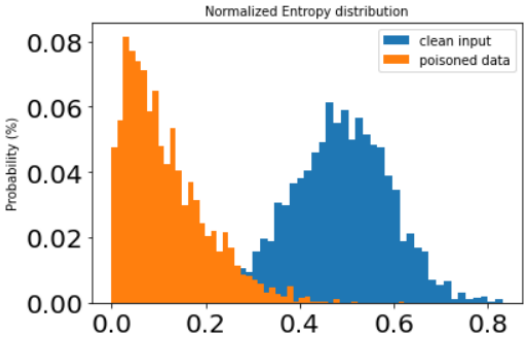

Figure : Entropy values of poisoned data are lower than that of clean input

4. We check the attack success rate and test accuracy of the model before performing STRIP.
5. We then perform classification using STRIP using the GoodNet class. This class has threshold as a class variable and has a function compile() which performs the classification. It classifies to class 1283 if the input is a backdoored image else it classifies as the actual class of the image.

3. We then check the attack success rate and the test accuracy after performing STRIP.
4. We then save the GoodNet object as a pickle file to be used in the eval script to test input images.

Backdoored Model	Poisonous Data	Attack Success Rate Before using STRIP	Test Accuracy before using STRIP	Attack Success Rate After using STRIP	Test Accuracy after using STRIP
Sunglasses Trigger	Sunglasses Backdoor	99.992	97.778	14.583	96.173
Multi triggered	Eyebrows Backdoor	91.348	96.009	28.926	89.205
	Lipstick Backdoor	91.523	96.267	14.429	94.435
	Sunglasses Backdoor	100.0	96.009	1.0619	94.544
Anonymous Trigger 1	Anonymous1 Backdoor	91.397	97.186	18.608	95.335
Anonymous Trigger 2	Not available	Cannot be calculated	95.962	Cannot be calculated	95.19

Table 2. Attack Success Rate and Test Accuracy before and after STRIP

Conclusion

The above results show that The STRIP is very effective in backdoor attack detection as it leads to a considerable decrease in attack success and also preserves test accuracy.

Classes Description

1) Class Name : ***Strip***
   Member Functions

   1) **superimpose(original image, overlay image)**
      Function parameters : original image, overlayimage
      Purpose : superimpose overlay image on original image

   2) **entropyCal(model, background,x_clean, n)**
      Function parameters : model is the badnet to be used, background image, valid data, the number of images to be overlaid on background image.
      Purpose: Calculate entropy for single background image.

   3) **returnEntropy(x_data, x_clean, model, n_test = 2000, n = 100)**
      Function parameters : x_data is the data to be tested for poisonous, x_clean is the clean data, mode is the badnet to be used, n_test is the number of samples from x_data to be used to calculate entropy, n is the number of overlays to b created for each input image from _data
      Purpose : Calculate entropy for the dataset of images to be tested if poisoned.

   4) **calThreshold(entropy_validation, entropy_trojan, n_test=2000, n=100, FRR = 0.01)**
      Function parameters : entropy_validationis entropy of clean data, entropy_trojan is the entropy of poisonous data, n_test is the number of samples from x_data to be used to calculate entropy, n is the number of overlays to b created for each input image from x_data.
      Purpose: Calculate the threshold entropy used to classify poisonous images

   5) **plot_overlap(entropy_benigh, entropy_trojan)**
      Function Parameters : entropy_beningh is the entropy of clean data and entropy trojan is the entropy of poisonous data
      Purpose : Plot the entropies of clean and poisonous data.

2) Class Name : ***GoodNet***

Purpose: Save the object to be used in eval function to detect poisonous images.

Class Variable :

Threshold - this is the detection boundary which detects poisonous input..

Member Functions

1) **compile(self,input_data, clean_data, model)**

Function parameters :input_data is the data to be tested , clean_data is the clean data and model is the backdoored network to be used for the data.

Purpose: Calculate entropy of input_data and check against the threshold. If entropy is greater than threshold, print the label predicted by the backdoored network. If entropy is less than threshold, print 1283 indicating the input data is poisonous.