ECE_GY 9163:Machine Learning Security Project Report

Cherian Thomas (kct298) | Rahul Petkar (rsp431) | Shubham Ingale (ski227) | Akhand Singh(aps646)

Introduction

Deep Neural Networks have played an integral role in many critical applications including disease identification, malware detection, face recognition, financial fraud recognition etc. However, the training of these complex models can take up to several weeks even when using multiple GPUs. This has led to users outsourcing the training of models to services like AWS. This gives an attacker a chance to create a maliciously trained network (BadNet), which performs very well on the clean validation dataset but behaves adversely on an attacker chosen input. The project implements a backdoor detector based on the paper "STRIP: A Defence Against Trojan Arracks on Deep Neural Networks"[1]. It exploits the fact that a triggered input tends to always show low entropy and a clean input tends to show high entropy and thus can be easily distinguished.

The Defense

The backdoored model and the validation dataset which constitutes the inputs the user has will be used to implement the defense.

The first step of the defense involves calculating the detection boundary. For this purpose, each input in the validation dataset is superimposed with 10 randomly selected inputs generating a set of 10 perturbed input for each input.

For each perturbed input, the Shannon entropy is calculated using the formula:

$$\mathbb{H}_n = -\sum_{i=1}^{i=M} y_i \times \log_2 y_i$$

Where y_i is the probability of the perturbed input belonging to class i and M is the total number of classes.

Now the entropy is summed over perturbed inputs as follows:

$$\mathbb{H}_{\text{sum}} = \sum_{n=1}^{n=N} \mathbb{H}_n$$

Where H_n is the entropy of n^{th} perturbed input and N is the total number of perturbed inputs

The normalized entropy is formulated from the summed entropy as follows:

$$\mathbb{H} = \frac{1}{N} \times \mathbb{H}_{\text{sum}}$$

The normalized entropy of the inputs has a normal distribution and for backdoored input, the detection boundary falls in the 1st percentile.

Thus, the detection boundary is calculated as:

Detection Boundary = mean(H) + (-2.326 * standard deviation(H))

Here -2.326 is the Z-score of 1st percentile.

Once the detection boundary is calculated, we calculate the entropy of each input test image. Each test image is duplicated 10 times and superimposed with a randomly selected input from the validation dataset. The normalized entropy is calculated for each of these test images similar to computation in the detection boundary. Now the calculated entropy is compared with the detection boundary value and if the entropy is less than the detection boundary, then then it is predicted as backdoored class. Else, the class predicted by the model is returned.

Result and Conclusion

The method implemented is insensitive to the type of trigger. Moreover, the defense works independent of the architecture of the deep neural network. However, the method implemented does not give good performance for an untargeted attack.

When tested on Sunglass backdoor of YouTube face dataset, the code has an accuracy of 96.6% on clean test data and an accuracy of 95.26% on poisoned data.

We observed that among all the backdoors provided the STRIP repaired network did poorly o n only the eyebrow backdoor (32% accuracy). While on the rest the STRIP repaired network did an acceptably good job of backdoor detection (78-92% accuracy). It even fared well on the anonymous backdoor.

Note: Backdoor is Represented by (N + 1) where N is equal to the y_validation_data.max() value;

N + 1 => y_validation_data.max() + 1

Reference

[1] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In Proceedings of the 35th Annual Computer Security Applications Conference, pages 113–125, 2019.