# ML For Security Final Project Report

Simon Wang @ssw8641
Haoran Xi @hx759
Jintian Zhang @jz4556

## INTRODUCTION

In this project, we utilized two different defense strategies to decrease the backdoor attack success rate on a compromised prediction model while maintaining a high prediction accuracy. Our defenses will generate a G-net that takes a set of N-class input data and outputs a list of prediction labels.

- If input data is uncontaminated, output labels are expected to be consistent with inputs' original labels. We consider G-net's percentage of correctly generated labels on uncontaminated dataset to be "Model Accuracy";
- If input data is poisoned, output labels are expected to be exactly N+1. We consider G-net's percentage of labels that equal to poisoned data labels to be "Attack Success Rate";

This project aims to optimize for high Model Accuracy and low Attack Success Rate. This Github repository contains all codes and files we have produced for this project.
https://github.com/haoranxi/ML_Cyber_final_project

## METHODS

### Fine-Pruning

Fine-Pruning is a method combining fine-tuning and pruning. Since the clean data and poisoned data will activate different neurons, we can prune the neurons that are less used by the clean data(more used by poisoned data) to reduce the attack success rate. We pruned the last pooling layer of BadNet B by removing one channel at a time from that layer. Channels should be removed in increasing order of average activation values over the entire validation set, to make sure the less used channels are firstly pruned. Every time we prune a channel, we will measure the new validation accuracy of the new pruned badnet. And we will stop pruning once the validation accuracy drops below a accuracy threshold, like: 10% less than the original clean validation accuracy.
But when the attacker knows this pruning defense, they will generate a pruning-aware attack which will not be defended by pruning defense. Then, we can fine-tune the pruned model using a clean validation dataset with seraval epochs, like: 10, 20, 50, to detect this backdoor attack.

### Overlaying Cross-Referencing

This method exploits the fact that all poisoned datasets are modified in a way to force the unrepaired model to generate a singular output. We reasoned that if we randomly choose x uncontaminated images and individually overlay them to a poisoned image, then the predicted labels of these x overlaid images should be similar to each other, i.e. having a smaller entropy/variance. Conversely, if we individually overlay uncontaminated images to an also uncontaminated image, the predicted labels would be random and patternless.
In our implementation, for each test image, we choose randomly 6 (num_overlay) uncontaminated images and overlay them over to the test image with a ratio of 5:9(clean/test). If

the variance of the 6 prediction labels are less than 0.1 (threshold), this test image will be considered to be a poisoned image; otherwise, an uncontaminated image.

## RESULTS

### Fine-Pruning

| | sunglass | anonymous_1 | multi-sunglass | multi-lipstick | multi-eyebrow |
|---|---|---|---|---|---|
| ACC | 84.29% | 85.89% | 85.75% | 85.75% | 85.75% |
| ASR | 0.038% | 0.087% | 0.029% | 0.087% | 0.0097% |

We can observe from the above results that the fine-pruning methods successfully detect the different backdoor attacks. And there is a tradeoff between ACC and ASR since the more channels we prune, the more ACC will be affected while the less backdoored channels will remain. And it's hard to remain high ACC and low ASR.

### Overlaying Cross-Referencing

The results below are obtained from repairing sunglasses_bd_net.

num_overlay=10, clean/test = 0.4:0.9, threshold = 0.01

```
Clean Classification accuracy: 89.0
Attack Success Rate: 8.0
```

num_overlay=10, clean/test = 0.5:0.9, threshold = 0.01

```
Clean Classification accuracy: 94.0
Attack Success Rate: 15.0
```

num_overlay=6, clean/test = 0.4:0.9, threshold = 0.01

```
Clean Classification accuracy: 87.0
Attack Success Rate: 4.0
```

num_overlay=6, clean/test = 0.5:0.9, threshold = 0.01

```
Clean Classification accuracy: 90.0
Attack Success Rate: 10.0
```

num_overlay=6, clean/test = 0.5:0.9, threshold = 0.1

```
Clean Classification accuracy: 89.0
Attack Success Rate: 12.0
```

We observe, from hyper-parameter selection, that ASR can be reduced at the expense of loss in ACC. At the same time, a larger number of overlaying images also increases both ASR and ACC.

## CONCLUSION

While both methods are able to increase a compromised model's clean data accuracy and decrease backdoor attack success rate, the Fine-Pruning method outperforms Overlaying Cross-Referencing in most cases.

## REFERENCE


- **Liu, Kang, Brendan Dolan-Gavitt, and Siddharth Garg. “Fine-pruning: Defending against backdooring attacks on deep neural networks.” International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, Cham, 2018.**
- **Gao, Yansong, 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, pp. 113-125. 2019.**