Backdoor Detector for BadNets using Pruning Defense
RAKSHANA B S RB5118

1 Introduction

In this project, we designed a backdoor detector for BadNets trained on the YouTube Face dataset using the pruning defense discussed in class. The goal of the detector is to correctly classify inputs as either clean or backdoored. The detector takes as input:

- original_badnet (B): A backdoored neural network classifier with N classes.
- Dvalid: A validation dataset of clean, labeled images.

The objective is to output the correct class if the test input is clean (in the range [1, N]) or class N+1 if the input is backdoored.

2 Method

Please note: I have replaced the variables as per the homework instructions in my Colab Notebook

- 'B' refers to 'original_badnet'.
- 'G' refers to 'RepairedNet'.

This naming convention aligns with the instructions for clarity and consistency.

2.1 Pruning Defense

We implemented the pruning defense by pruning the last pooling layer of original_badnet (B), just before the fully connected layers. Channels in the pooling layer are removed one at a time, starting with those that have the highest average activation values over the entire validation set. Pruning continues until the validation accuracy drops at least X% below the original accuracy. The pruned network becomes the new network B'.

2.2 Goodnet G

Our goodnet G works as follows for each test input:

- Run the input through both original_badnet (B) and the pruned network (B').
- If the classification outputs of B and B' are the same (i.e., class i), the detector outputs class i.
- If the outputs differ, the detector outputs class N+1.

3 Results and GitHub Repository

I have provided all the code related to this project in our GitHub repository: `https://github.com/rbrk17/ML_R`. The repository includes code for the backdoor detector, pruning defense, and evaluation scripts. You can also find a detailed README file that explains how to run the code.

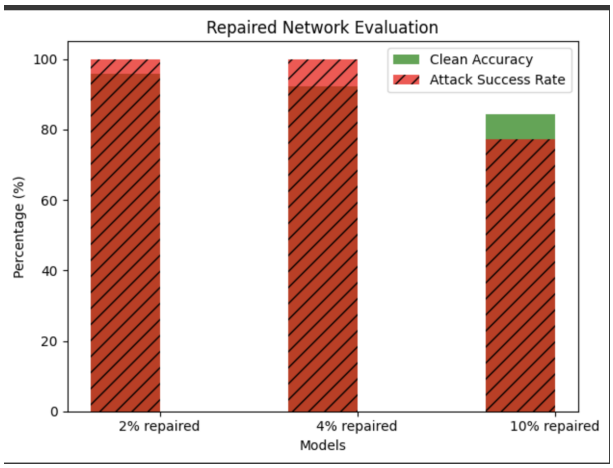

Figure 1: Repaired Network Evaluation

Table 1: Repaired Networks Evaluation

Model	Repaired Clean Accuracy	Attack Success Rate
2% repaired	95.7443	100
4% repaired	92.1278	99.9844
10% repaired	84.3336	77.2097

4 Conclusion

In conclusion, we successfully designed a backdoor detector for BadNets using the pruning defense. We evaluated the detector on a BadNet with a sunglasses backdoor and repaired networks with varying pruning thresholds. The results demonstrate the effectiveness of our defense in identifying clean and backdoored inputs. This project showcases a practical approach to securing neural networks against backdoor attacks.