# BACKDOOR DETECTORS

Arushi Arora (aa10350) [1]

[1] *Department of Electrical and Computer Engineering, New York University*

## Problem Overview

The goal of this lab is to design a backdoor detector for BadNets trained on the YouTube Face dataset. A pruning defense technique is employed to repair the BadNet and create a backdoor detector. The BadNet (B) has a backdoor that can be detected using the pruning defense method. The repaired network (B') is then used in conjunction with the original BadNet to form a backdoor detector (G).

1. **Pruning BadNet B:**
   - **Data Preparation:**
     - Utilize clean validation data (Dvalid) to obtain activations from the last pooling layer of BadNet B.
     - Average activations over the entire validation set
   - **Channel Pruning:**
     - Arrange channels in increasing order based on averaged activations.
     - Iteratively prune one channel at a time and measure the new validation accuracy.
     - Stop pruning when the accuracy drops by at least X% below the original accuracy.
2. **GoodNet G Construction:**
   - For each test input, run it through both BadNet B and the pruned BadNet B'.
   - If the classification outputs of B and B' are the same, output the correct class i. If they differ, output class N+1 (backdoored).
3. **Evaluation:**
   - **Original BadNet B (B1):**
     - Evaluate the original BadNet B on a specific backdoor attack (sunglasses backdoor)
   - **Repaired Networks B' (X = 2%, 4%, 10%):**
     - Evaluate the repaired networks for different pruning percentages using the provided evaluation script.
     - Assess accuracy on clean test data and attack success rate on backdoored test data.
   - **GoodNet models:**
     - Evaluate the accuracy of GoodNet models on clean test data and attack success rate on backdoored test data.

## Results and Discussions

- **Pruning Defense Effectiveness:**
  - Pruning BadNet B for different X values (2%, 4%, 10%) revealed varying impacts on the network.
  - As expected, at the beginning of the pruning process, removing poorly activated neurons had minimal effect on the attack success rate. The trade-off became apparent as more critical neurons were pruned, leading to a sharp decline in both attack success rate and clean classification accuracy.
  - Pruning-aware attacks, where the attacker strategically placed the backdoor in highly activated neurons, posed a significant challenge for the defense. Beyond a certain pruning level, both the attack success rate and clean classification accuracy experienced a simultaneous decline.

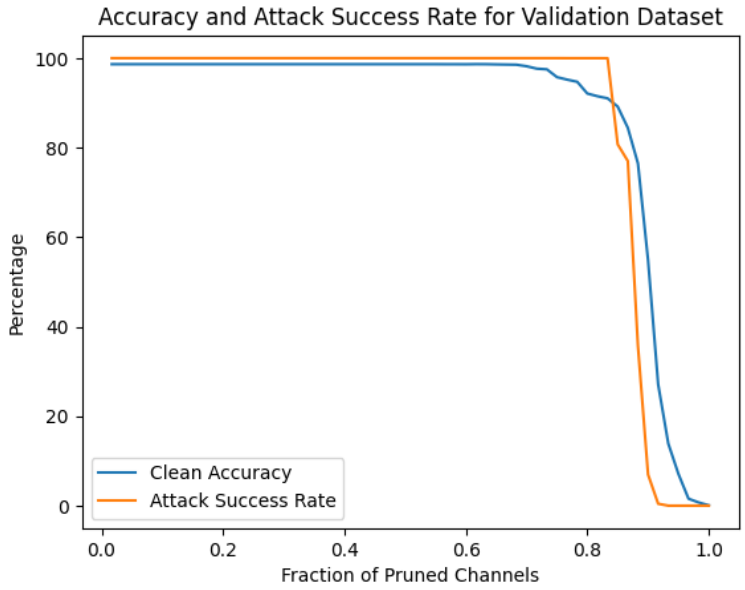


Figure 1: Accuracy and Attack Success Rate for Validation Dataset for different fraction of Pruned channels

| Model | Clean Test Data Accuracy (%) | Attack Success Rate (%) |
|---|---|---|
| `pruned_model_X_2` | 95.900234 | 100.000000 |
| `pruned_model_X_4` | 92.291504 | 99.984412 |
| `pruned_model_X_10` | 84.544037 | 77.209665 |

Table 1: Performance Metrics for Pruned Models

- **GoodNet G Performance:**
  - GoodNet G, constructed by comparing outputs from BadNet B and pruned BadNet B', exhibited varying performance on different scenarios.
  - In scenarios where pruning was effective in reducing the attack success rate, G demonstrated its ability to reliably detect the original backdoor attack.

In conclusion, the pruning defense demonstrated effectiveness in reducing the attack success rate but faced challenges, especially in the presence of a pruning-aware attack. The trade-off between maintaining clean classification accuracy and mitigating the backdoor was evident, highlighting the complexity of defending against sophisticated attacks. Acknowledging the limitations, the pruning defense still offers a valuable layer of protection, and future work should focus on refining strategies to enhance its robustness in real-world scenarios.

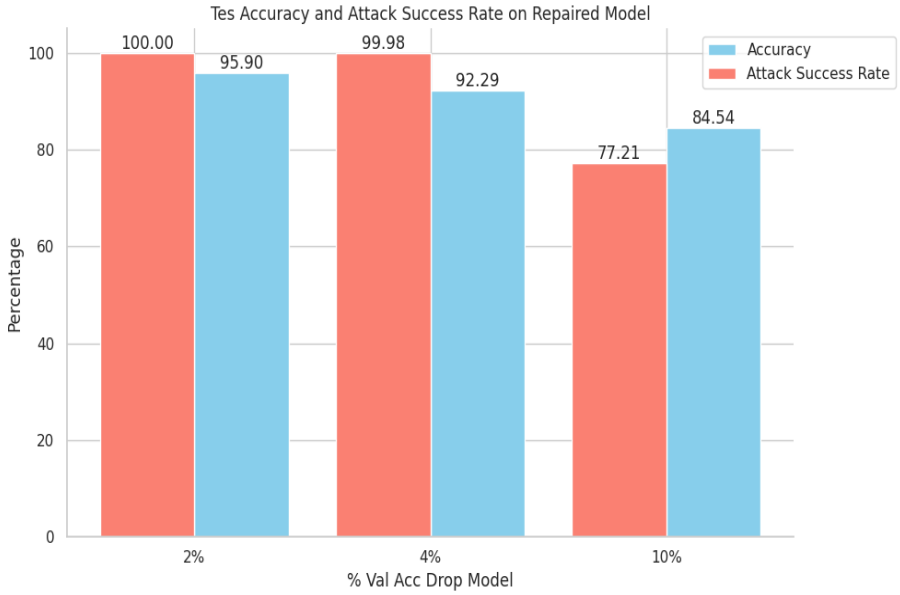


Figure 2: Test Accuracy and Attack Success Rate on repaired model for different validation drop percentages

| GoodNet Model | Clean Test Data Accuracy (%) | Attack Success Rate (%) |
|---|---|---|
| GoodNet_X_2 | 95.744349 | 100.000000 |
| GoodNet_X_4 | 92.127825 | 99.984412 |
| GoodNet_X_10 | 84.333593 | 77.209665 |

Table 2: Performance Metrics for GoodNet Models

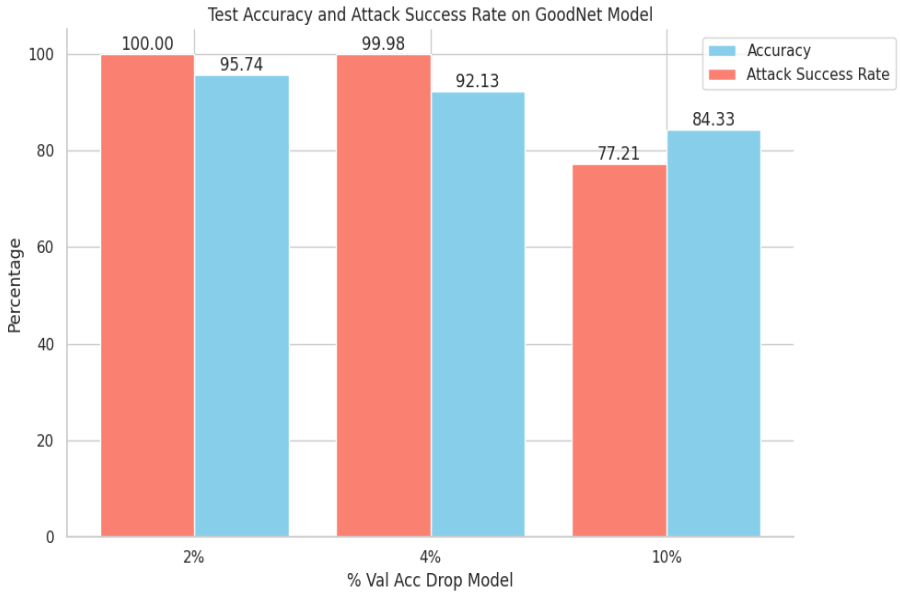


Figure 3: Test Accuracy and Attack Success Rate on GoodNet model for different validation drop percentages