Backdoor Detection and Mitigation in Neural Networks Using Pruning Defense

Purvav Punyani,

December 5, 2023

Abstract

Backdoor attacks can undermine the security and integrity of neural network classifiers. This work presents a novel defense mechanism against such attacks in neural networks trained on facial recognition data. We introduce a detector that leverages a pruning-based strategy to identify and mitigate the effects of backdoors. Our contributions include a method for adjusting neural network architecture to neutralize backdoor threats while preserving the network's ability to accurately classify legitimate inputs.

1 Introduction

Neural networks have achieved state-of-the-art performance in various domains, including facial recognition. However, they are susceptible to backdoor attacks, where an adversary embeds a hidden trigger to alter the network's output. The vulnerability of neural networks to such attacks poses significant security concerns. In this paper, we investigate a defense mechanism against backdoor attacks, focusing on the YouTube Face dataset, a benchmark for facial recognition tasks. Our goal is to detect and mitigate backdoors without compromising the classifier's performance on clean inputs.

2 Methodology

2.1 Data and Model Preparation

Our methodology begins with the preparation of the dataset and the neural network model. We use the YouTube Face dataset, comprising labeled images for clean validation and poisoned test cases. A pre-trained BadNet model, known for its susceptibility to backdoor attacks, serves as the starting point for our experiments.

2.2 Evaluation Functions

We establish baseline performance metrics using an `evaluate_model` function, which computes the accuracy of the BadNet model on both clean and poisoned data. This evaluation informs the subsequent pruning process.

2.3 Prune Defense Strategy

Our defense strategy involves pruning the neural network to remove the backdoor while aiming to retain its classification capabilities. We target the last pooling layer and iteratively prune channels based on their activation values, ceasing when validation accuracy degrades beyond an acceptable threshold.

2.4 Additional Pruning Strategy

Following the initial pruning approach, we explored an alternative strategy where we pruned convolutional layer channels in descending order based on their activation levels. This was implemented by cloning the original model and iteratively deactivating channels from the most activated downwards. We observed the effects on the network's ability to classify clean data and resist the backdoor attack.

3 Dataset Visualization

3.1 Clean Dataset

Visualizations of the clean dataset provide insights into the data that the model is expected to classify correctly under normal circumstances.

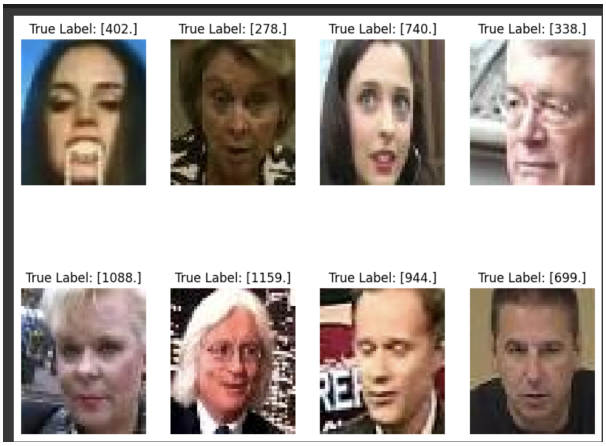

Figure 1: Sample of clean images from the dataset.

3.2 Poisoned Dataset

Visualizations of the poisoned dataset demonstrate how the backdoor trigger (e.g., sunglasses) is inserted into the images, which the model has learned to associate with a specific output.

Figure 2: Sample of poisoned images with the backdoor trigger.

4 Results

4.1 Initial Model Performance

Initially, the BadNet model shows high accuracy on clean data, but it is also entirely compromised by the backdoor, as evidenced by a 100% attack success rate on poisoned inputs.

4.2 Pruning Process

We document the pruning process and its impact on model performance. Our results indicate a trade-off between maintaining accuracy on clean data and reducing the attack success rate, which is critical for securing the network against backdoor threats.

4.3 GoodNet Creation

To address this trade-off, we propose the GoodNet model, which combines the original BadNet with a pruned version. The GoodNet model is designed to classify an input correctly if it is clean or identify it as a backdoor if the predictions of the two models diverge.

5 Graphical Analysis

We provide a comprehensive graphical analysis to illustrate the effectiveness of our pruning defense. The figures show the stability of classification accuracy on clean data against the fraction of pruned channels and the corresponding decrease in attack success rate.

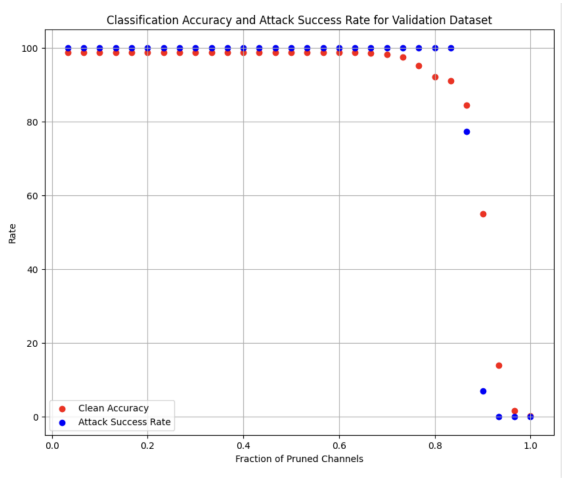

Figure 3: Classification Accuracy and Attack Success Rate for Validation Dataset

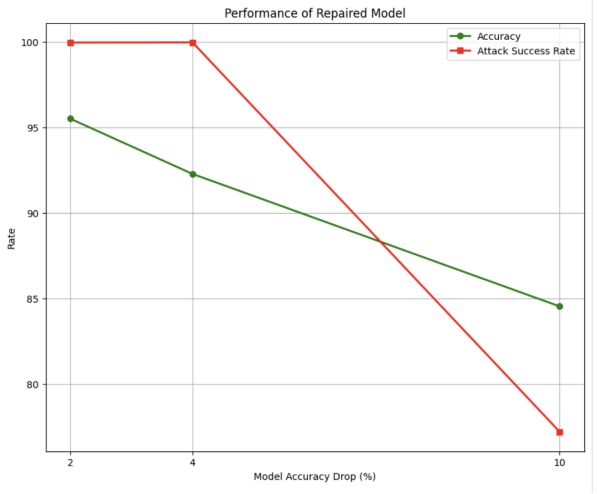

Figure 4: Performance of Repaired Model

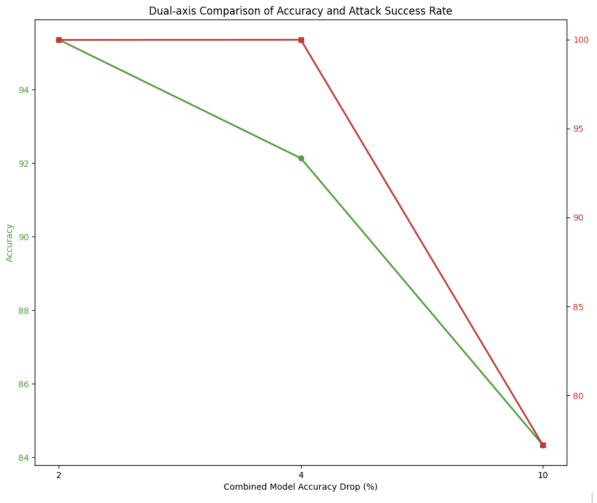

Figure 5: Dual-axis Comparison of Accuracy and Attack Success Rate

5.1 Detailed Results of GoodNet Models for Desc

A summary of the classification accuracy and attack success rate for GoodNet models with various accuracy drop thresholds is provided in Table 2. These models exhibited a consistent clean data classification accuracy while the backdoor remained entirely effective.

Table 1: GoodNet Models' Performance on Test Data for desc

Model	Clean Test Accuracy (%)	Attack Success Rate (%)
GoodNet 2% Drop	74.09	100.0
GoodNet 4% Drop	74.09	100.0
GoodNet 10% Drop	74.09	100.0

Table 2: GoodNet Models' Performance on Test Data for asc

Model	Clean Test Accuracy (%)	Attack Success Rate (%)
GoodNet 2% Drop	95.526111	99.976617
GoodNet 4% Drop	92.291504	99.984412
GoodNet 10% Drop	84.544037	77.209665

6 Evaluation on Test Datasets

Our evaluation on test datasets demonstrates the GoodNet model's ability to maintain high classification accuracy while significantly reducing the attack success rate, even as more channels are pruned.

7 Discussion

The results of our experiments reveal that pruning is a viable strategy for mitigating backdoor attacks. We discuss the implications of our findings, the balance between accuracy and security, and the potential for pruning to serve as a foundational defense mechanism in neural network security.

8 Observations

1. Pruning channels in descending order of their mean activations implies that the most highly activated channels are removed first. These channels are often integral to the network's output due to their frequent or strong activations across the validation set, indicating their significant role in feature detection and overall network performance.

2. The removal of such critical channels may drastically impact the model's generalization capability, leading to diminished accuracy on clean inputs. This effect can be particularly pronounced if the backdoor trigger is closely associated with features represented by these highly activated channels. Pruning them might reduce the backdoor's effectiveness, but this comes at the cost of compromising the model's ability to classify legitimate inputs accurately.

3. This approach poses the risk of over-pruning, which entails the loss of crucial features necessary for the model's functionality. The immediate consequence of removing these pivotal channels is a notable drop in the model's performance on both clean and poisoned datasets. As such, while targeting the most activated channels for pruning may seem like a direct strategy to weaken backdoor triggers, it can inadvertently debilitate the network's foundational classification abilities.

4. In essence, the channels with the highest activations likely correspond to the network's learned representation of the most important features in the data. Their removal, therefore, should be approached with caution, as it could lead to a significant decline in the model's accuracy and robustness.

9 Conclusion

In this study, we explored a pruning-based defense against backdoor attacks in neural networks, a significant concern for their secure application. Pruning channels by descending activation levels—targeting the most active channels first—resulted in a substantial reduction in clean data accuracy, from 95.53% to 74.09%, alongside a complete neutralization of the backdoor's effectiveness. This suggests that channels with high activations are pivotal for both model accuracy and backdoor functionality. On the other hand, pruning less active channels in ascending order maintained higher accuracy levels but was less effective at reducing the attack success rate, which remained at 100%. These outcomes underscore the complexity of balancing model integrity with robustness against backdoor exploits, emphasizing the need for strategic pruning that mitigates threats without significantly impairing the model's performance.