Report on the Implementation and Analysis of a Backdoor Detector for BadNets Using Pruning Defense

Introduction

This report details the design and evaluation of a backdoor detector for BadNets, specifically trained on the YouTube Face dataset. The proposed system, denoted as `G`, is designed to identify and mitigate backdoored neural network classifiers. The detector employs a pruning defense mechanism that modifies the neural network to recognize and classify backdoored inputs accurately.

Methodology

System Design

The system takes a backdoored neural network classifier, `B`, with `N` classes, and a validation dataset of clean, labeled images, `Dvalid`. The output is a repaired network `G`, which has `N+1` classes. The primary tasks of `G` are:

1. Correct classification of clean inputs within `[1, N]` range.
2. Identification of backdoored inputs as class `N+1`.

Pruning Defense Implementation

The pruning defense involves removing one channel at a time from the last pooling layer of `B`. Channels are pruned in decreasing order of average activation values over the entire validation set. The process stops once the validation accuracy drops at least `X%` below the original accuracy, yielding a new network `B'`.

For each test input, `G` runs it through both `B` and `B'`. If the classification outputs match, `G` outputs the class `i`. If they differ, `G` outputs class `N+1`.

Data Preparation

The datasets involved:

- Clean validation and test datasets.
- Poisoned (backdoored) validation and test datasets.

These datasets are loaded, and examples of clean and poisoned data are visualized for analysis.

Model Evaluation

The model is evaluated based on its performance on the clean and poisoned datasets. The key metrics are the accuracy on clean data and the Attack Success Rate (ASR) on poisoned data.

Results and Discussion

Pruning Process

The pruning process iteratively removes channels from the last pooling layer. The validation accuracy is monitored throughout the process. Models at different thresholds (`2%`, `4%`, `10%`, `20%`) are saved for comparison.

Refined Model Evaluation

The refined models (`G`) are tested against clean and poisoned datasets. The effectiveness of the defense mechanism is gauged by observing the change in accuracy and ASR.

Analysis of Pruning Effectiveness

The effectiveness of pruning as a defense mechanism was analyzed. Key observations include:

- Neural networks' inherent robustness can make them resistant to changes like pruning.
- Specificity in pruning is crucial, as uniform pruning might not target the neurons influential in adversarial contexts.
- The balance between pruning aggressiveness and performance maintenance is challenging to achieve.

Limitations

- Pruning may not significantly impact a network's ability to process poisoned inputs.
- The approach might require a more comprehensive restructuring or retraining for effectiveness.
- The complexity of neural networks makes it difficult to predict the effects of pruning accurately.

Conclusion

While pruning can reduce a model's size and complexity, its effectiveness as a defense against sophisticated attacks is limited. The nature of the attack and the specifics of the network architecture play significant roles in determining the success of such a defense strategy. Future work might involve exploring more comprehensive approaches for robust defense against attacks on neural networks.

Appendices

Appendix A: Explanations for ineffectiveness of pruning

The ineffectiveness of pruning as a defense mechanism against certain types of attacks, especially in neural networks, can be attributed to several factors:

1. Inherent Robustness of Neural Networks**: Neural networks, particularly deep networks, are known for their redundancy and robustness. When certain channels or neurons are pruned, the network can often compensate for the loss, maintaining its function almost as before. This resilience, while beneficial for generalization and dealing with noisy inputs, can also mean that pruning does not significantly impact the network's ability to process poisoned inputs or adversarial examples.

2. **Targeted Nature of Some Attacks**: Certain attacks are crafted with the specific architecture and weights of the target neural network in mind. If the attack is designed to exploit specific vulnerabilities or features of the network, pruning might not remove these vulnerabilities. Instead, it could require a more comprehensive restructuring or retraining of the model.

3. **Insufficient Pruning**: The extent of pruning is a critical factor. If pruning is too conservative, it might not remove enough of the network's capacity to mitigate the attack. On the other hand, aggressive pruning can degrade the performance of the network on legitimate tasks. Finding the right balance is challenging and often specific to the particular model and task.

4. **Distribution Shift in Poisoned Data**: Poisoned or adversarial data often represents a significant shift from the distribution of clean data. If pruning is guided primarily by performance on clean data, it may not address the specific ways in which poisoned data manipulates the model's responses. Therefore, the pruned model might still be vulnerable to attacks crafted to exploit these distributional differences.

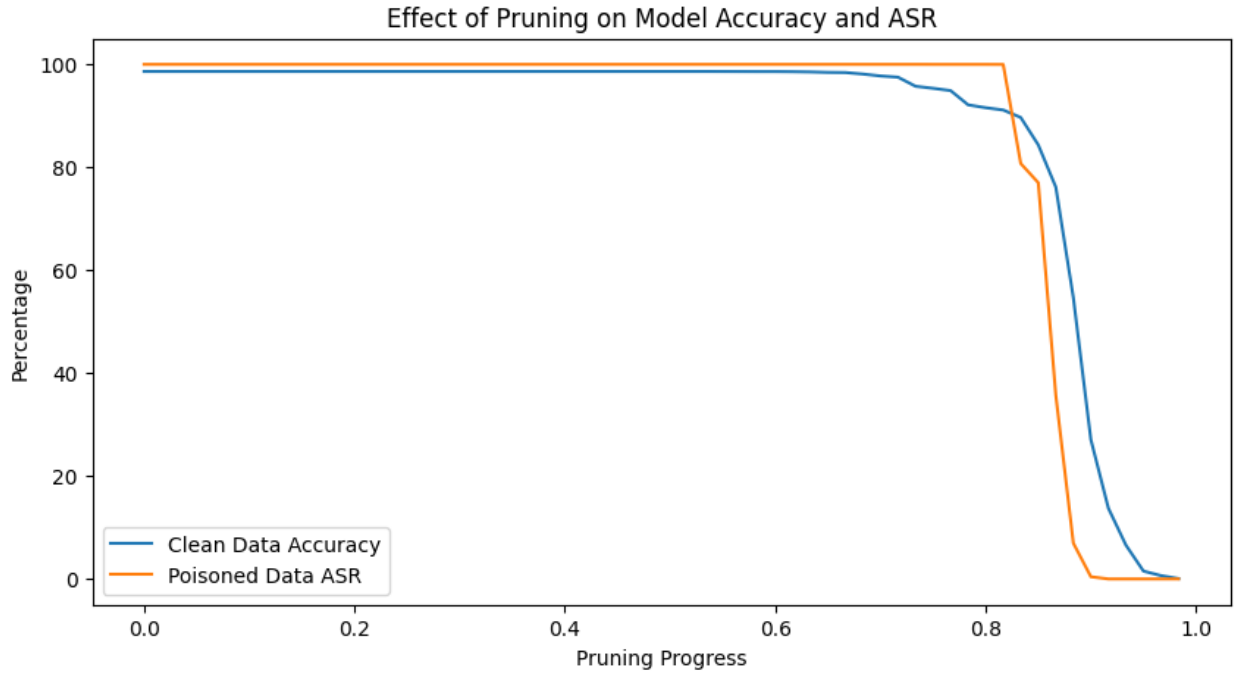
5. **Lack of Specificity in Pruning**: Pruning, especially when it's done uniformly or based on generic criteria like activation values, may not target the specific neurons or channels that are most influential in the adversarial context. The neurons critical for processing poisoned inputs might remain largely unaffected, leaving the network's vulnerabilities intact.

6. **Recovery of Attack Capability**: In some cases, even after pruning, the model may still be capable of learning or recovering the attack patterns during further training or fine-tuning, especially if the underlying data used for these processes is not cleansed of poisoned examples.

7. **Complexity of Neural Networks**: Modern neural networks often have millions of parameters, and the relationship between these parameters and specific model behaviors can be highly non-linear and complex. This complexity makes it difficult to predict how changes like pruning will affect the model's behavior in all scenarios, including under attack.

In conclusion, while pruning can be an effective tool for reducing model size and computational complexity, its effectiveness as a defense mechanism against attacks on neural networks is limited and can be highly dependent on the nature of the attack and the specifics of the network architecture. More comprehensive approaches might be required for robust defense against sophisticated attacks.
"""

Appendix B: Visualization



Appendix C: CSV Data

Pruning Progress	Clean Data Accuracy	Poisoned Data ASR
0.0	98.62042088854250	100.0
0.016666666666666700	98.62042088854250	100.0
0.03333333333333330	98.62042088854250	100.0
0.05	98.62042088854250	100.0
0.06666666666666670	98.62042088854250	100.0
0.08333333333333330	98.62042088854250	100.0
0.1	98.62042088854250	100.0
0.11666666666666700	98.62042088854250	100.0
0.13333333333333300	98.62042088854250	100.0
0.15	98.62042088854250	100.0
0.16666666666666700	98.62042088854250	100.0
0.18333333333333300	98.62042088854250	100.0
0.2	98.62042088854250	100.0
0.21666666666666700	98.62042088854250	100.0
0.23333333333333300	98.62042088854250	100.0
0.25	98.62042088854250	100.0
0.26666666666666700	98.62042088854250	100.0
0.2833333333333330	98.62042088854250	100.0
0.3	98.62042088854250	100.0
0.31666666666666700	98.62042088854250	100.0
0.3333333333333330	98.62042088854250	100.0
0.35	98.62042088854250	100.0
0.36666666666666700	98.62042088854250	100.0
0.38333333333333300	98.62042088854250	100.0
0.4	98.62042088854250	100.0
0.4166666666666670	98.62042088854250	100.0
0.43333333333333300	98.62042088854250	100.0
0.45	98.62042088854250	100.0
0.4666666666666670	98.62042088854250	100.0
0.48333333333333300	98.62042088854250	100.0
0.5	98.62042088854250	100.0

0.5166666666666670	98.62042088854250	100.0
0.5333333333333330	98.62042088854250	100.0
0.55	98.61262665627440	100.0
0.5666666666666670	98.60483242400620	100.0
0.5833333333333330	98.59703819173810	100.0
0.6	98.59703819173810	100.0
0.6166666666666670	98.57365549493380	100.0
0.6333333333333330	98.52689010132500	100.0
0.65	98.44115354637570	100.0
0.6666666666666670	98.4099766173032	100.0
0.6833333333333330	98.11379579111460	100.0
0.7	97.74746687451290	100.0
0.7166666666666670	97.50584567420110	100.0
0.7333333333333330	95.74434918160560	100.0
0.75	95.34684333593140	99.9913397419243
0.7666666666666670	94.90257209664850	99.9913397419243
0.7833333333333330	92.1278254091972	99.9913397419243
0.8	91.57443491816060	99.9913397419243
0.8166666666666670	91.13016367887760	99.98267948384860
0.8333333333333330	89.68043647700700	80.7309257815883
0.85	84.34918160561190	76.99835455096560
0.8666666666666670	76.14964925954790	35.68892352992120
0.8833333333333330	54.66874512860480	6.954187234779600
0.9	27.069368667186300	0.4243526457088420
0.9166666666666670	13.725643024162100	0.0
0.9333333333333330	6.562743569758380	0.0
0.95	1.5198752922837100	0.0
0.9666666666666670	0.646921278254092	0.0
0.9833333333333330	0.0701480904130943	0.0