Introduction of Machine Learning Security

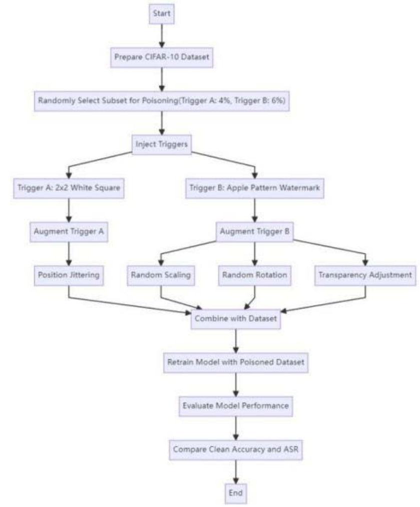
Backdoor Attack on CIFAR-10 with ResNet18: A Trigger Augmentation Study

Introduction

This project explores the implementation and optimization of backdoor attacks on the ResNet18 model using the CIFAR-10 dataset. Two distinct triggers (Trigger A and Trigger B) were designed to manipulate the model's classification. The study focuses on enhancing the robustness of these triggers through various transformations to maximize attack success rates (ASR) while maintaining high accuracy on clean data.

Method

The backdoor attack was implemented through the following steps, illustrated in the flowchart below:


Start
Prepare CIFAR-10 Dataset
Randomly Select Subset for Poisoning(Trigger A: 4%, Trigger B: 6%)
Inject Triggers
Trigger A: 2x2 White Square
Trigger B: Apple Pattern Watermark
Augment Trigger A
Augment Trigger B
Position Jittering
Random Scaling
Random Rotation
Transparency Adjustment
Combine with Dataset
Retrain Model with Poisoned Dataset
Evaluate Model Performance
Compare Clean Accuracy and ASR
End

First, the CIFAR-10 dataset was loaded and split into training and validation sets. To prevent overfitting, I applied data augmentation techniques such as random cropping, horizontal flipping, and normalization to the original training data.

Next, I selected a random subset of the training data for injecting triggers. Specifically, 4% of the subset was allocated for Trigger A (a 2×2 white square in the bottom-right corner) due to its simple and easily learnable pattern, while 6% was allocated for Trigger B (an apple-pattern watermark) to account for its more complex and blended nature, which might require more samples to influence the model's decision-making process effectively.

After selecting the subsets, I injected the triggers and applied specific augmentations: position jittering for Trigger A and random scaling, rotation, and transparency adjustment for Trigger B.

These augmentations altered the appearance of the triggers but did not change the injection ratio. The model was then retrained with the poisoned dataset.

Finally, I evaluated the model's accuracy on clean data and the attack success rate (ASR) for both triggers under different augmentation strategies to assess the effectiveness of the attack. During testing, the triggers were used in their original form without augmentation, as required by the assignment.

Experiment

Settings

Dataset: CIFAR-10 (50k training, 10k test), split training set 80%-20% for validation.

Optimizer: Adam (learning_rate = 0.001, weight_decay = 5e-4).

Learning Rate Schedule: StepLR, step_size = 10 epoch, gamma = 0.5

Training Parameters: batch size = 128, epoch = 80, saving pth at lowest valid loss.

Reproducibility: Fixed global random seeds(42) for reproducing the result.

Fairness: Identical hyper-parameters for clean/poisoned models.

Platform: PyTorch on CUDA-enabled NVIDIA RTX 4090.

Results and Discussion

Experiment Setting	Clean Accuracy	Trigger A ASR	Trigger B ASR
Original Model	91.14%	10.59%	9.75%
No Trigger Augmentation	90.49%	**99.48%**	88.48%
Both Triggers Augmented (no ratio changed)	90.9%	98.26%	92.99%
Only B Trigger Augmented (no ratio changed)	90.32%	97.63%	**94.71%**

The original model achieved a clean accuracy of 91.14%, with baseline ASR values of 10.12% for Trigger A and 9.93% for Trigger B. After injecting triggers without augmentation, the clean accuracy dropped to 90.49%, while ASR increased to 99.48% for Trigger A and 88.48% for Trigger B. This shows basic injection was effective but needed improvement in robustness and balance. Augmenting both triggers kept clean accuracy at 90.9% and raised ASR to 98.26% for Trigger A and 92.99% for Trigger B, significantly enhancing robustness. The best balance was achieved by augmenting only Trigger B, resulting in a clean accuracy of 90.32%, ASR of 97.63% for Trigger A, and 94.71% for Trigger B. This indicates selective augmentation of Trigger B improved its ASR with minimal impact on Trigger A's ASR, maintaining high accuracy on clean data. Using only a small fraction of the dataset for backdoor injection and observing less than a 1% drop in clean accuracy further highlights the high concealment of the attack. These results demonstrate that selective augmentation effectively enhances backdoor attack success while maintaining model performance.

Conclusion

Trigger robustness enhancements and balancing significantly enhanced the attack success rate while maintaining high model accuracy on clean data. Future work may explore more concealed methods, such as frequency-domain energy enhancement, to achieve similar effects with greater stealth.