Backdoor Attacks in Deep Learning:

From Trigger Injection to Detection and Defense**

Author: Phan Huu Thong
Degree: Bachelor of Computer Science (Cyber Security)
Project Type: Independent Technical Research Project
Field: Artificial Intelligence Security / Trustworthy AI

Abstract

Backdoor attacks represent a critical threat to the trustworthiness of modern deep learning systems. Unlike traditional adversarial examples that manipulate inputs at inference time, backdoor attacks embed malicious behavior directly into the training process, allowing models to behave normally on clean inputs while producing attacker-controlled outputs when a specific trigger is present.

This project presents a systematic, end-to-end study of backdoor attacks in image classification models using the MNIST dataset. Multiple trigger strategies are implemented, ranging from simple pixel-based patterns to optimized adversarial patches. The project further investigates backdoor detection using activation clustering based on PCA and KMeans, followed by a defense strategy employing fine-pruning to mitigate backdoor behavior while preserving clean accuracy.

Experimental results demonstrate that advanced backdoor triggers can achieve near-perfect attack success rates with minimal impact on clean accuracy, and that fine-pruning serves as a practical post-training defense against backdoored neurons. This work highlights the importance of integrated attack, detection, and defense pipelines for securing real-world AI systems.

1. Introduction

Deep learning models are increasingly deployed in safety-critical and security-sensitive applications, including medical diagnosis, autonomous driving, and financial systems. While these models often achieve high accuracy on benchmark datasets, their vulnerability to adversarial manipulation raises significant concerns regarding reliability and trust.

Backdoor attacks pose a particularly dangerous threat because they remain dormant during normal operation and are activated only when a specific trigger is present. As a result, backdoored models may pass standard validation procedures while containing hidden malicious behavior.

The objective of this project is to provide a comprehensive study of backdoor attacks from an applied AI security perspective. Rather than focusing on a single attack or defense, this work implements a complete lifecycle pipeline covering attack construction, detection, and mitigation. The project is designed as an independent technical study to demonstrate practical understanding of AI system vulnerabilities and defensive strategies.

---

2. Background and Threat Model

2.1 Backdoor Attacks in Deep Learning

A backdoor attack is a training-time poisoning attack in which an adversary injects malicious samples into the training dataset. These poisoned samples contain a trigger pattern and are assigned a target label chosen by the attacker. After training, the model behaves normally on clean data but consistently misclassifies triggered inputs.

Compared to adversarial examples, backdoor attacks are more difficult to detect because:

- The malicious behavior is embedded in the model parameters.
- Clean accuracy remains high.
- The trigger may be visually imperceptible.

2.2 Threat Model Assumptions

This project assumes the following threat model:

- The attacker has partial control over the training data.
- The defender does not know the trigger pattern.
- The defender has access to clean validation data.
- The deployed model may already be compromised.

These assumptions reflect realistic scenarios in outsourced or third-party model training environments.

3. System Overview

The project follows a progressive pipeline consisting of six levels, each addressing a different aspect of AI security:

1. Backdoor attack construction
2. Trigger stealth enhancement
3. Learned adversarial triggers
4. Detection via internal activations
5. Defense via neuron pruning

A system-level pipeline diagram illustrates the complete workflow from clean data to sanitized model deployment.

---

4. Backdoor Attack Implementation (Levels 1–4)

4.1 Dataset and Model

- **Dataset:** MNIST handwritten digits
- **Model:** SimpleCNN implemented in PyTorch
- **Training Setup:** Joint training on clean and poisoned samples

---

4.2 Level 1 – Pixel-Based Trigger

A static pixel trigger is embedded in a fixed corner of selected training samples. The model is trained to associate the trigger with a specific target class.

This baseline attack demonstrates fundamental backdoor behavior with high attack success and minimal degradation of clean accuracy.

4.3 Level 2 – Invisible Noise Trigger

To increase stealth, a low-amplitude noise-based trigger is applied to poisoned samples. The perturbation remains visually imperceptible while maintaining strong backdoor activation.

4.4 Level 3 – Semantic Blur Trigger

This level introduces a semantic transformation-based trigger using localized blur. Unlike pixel-aligned triggers, semantic triggers exploit higher-level feature representations.

4.5 Level 4 – Adversarial Patch Trigger

A learnable adversarial patch is jointly optimized during training. This patch achieves near-perfect attack success rates while remaining compact and transferable.

4.6 Attack Results Summary

Trigger Type	Clean Accuracy	Attack Success Rate
Pixel Trigger	~99%	~95%
Invisible Noise	~98%	~97%
Semantic Blur	~98%	~99%
Adversarial Patch	~98%	~100%

5. Backdoor Detection (Level 5)

5.1 Activation Extraction

Internal activations are extracted from hidden layers of the trained model. These activations capture latent feature representations that reflect backdoor behavior.

5.2 Activation Clustering

Principal Component Analysis (PCA) is applied to reduce dimensionality, followed by KMeans clustering to separate clean and poisoned samples.

Experimental results show that poisoned samples form a distinguishable cluster in activation space, enabling effective detection without knowledge of the trigger.

6. Backdoor Defense via Fine-Pruning (Level 6)

6.1 Defense Motivation

Fine-pruning aims to remove neurons that are inactive for clean data but critical for backdoor activation. This method does not require retraining from scratch, making it suitable for post-deployment mitigation.

6.2 Pruning Strategy

- Compute mean neuron activation on clean validation data.
- Identify low-importance neurons.
- Iteratively prune selected neurons.
- Fine-tune the remaining network to recover clean accuracy.

6.3 Defense Results

The defense significantly reduces attack success while preserving clean accuracy, demonstrating that backdoor behavior can be mitigated through targeted neuron removal.

---

7. Discussion

This project highlights several key insights:

- Backdoor attacks can remain highly effective without degrading clean performance.
- Learned triggers pose greater detection challenges.
- Activation-based detection provides strong signals but is not foolproof.
- Fine-pruning is an effective and practical defense, though adaptive attacks may bypass it.

---

8. Conclusion

This work presents a complete, applied study of backdoor attacks in deep learning systems, covering attack construction, detection, and defense. The results demonstrate that even simple models can be severely compromised and emphasize the need for systematic security evaluation in AI deployment pipelines.

---

9. Future Work

- Extension to CIFAR-10 and ImageNet
- Evaluation on transformer-based models
- Comparison with Neural Cleanse and Spectral Signatures
- Certified defenses against adaptive attackers

Appendix

- GitHub Repository: *AI-Security-Backdoor-Detection*
- Notebooks: Level 1–6 implementations