Detecting Data Poisoning and Adversarial Attacks in CNN-Based Image Classifiers

Course: INFO-6149 – Machine Learning Security

Group Members: Shah Fahad, Zaid Khan, Mohammad Osama

Submission: July 2025

1. Introduction

Convolutional Neural Networks (CNNs) are widely used in image classification tasks due to their high accuracy and performance. However, they remain vulnerable to malicious manipulation, including data poisoning during training and adversarial attacks at inference time. These vulnerabilities pose serious risks to the reliability and safety of machine learning systems, especially in security-critical applications.

This project investigates both training-time (poisoning) and inference-time (adversarial) attacks using the CIFAR-10 dataset. It implements and evaluates defenses such as label sanitization, dropout regularization, and randomized smoothing. A comparative evaluation is conducted to measure robustness improvements, followed by an in-depth documentation of threat models, results, and failure modes.

2. Threat Models and Attack Mechanics

2.1 Data Poisoning Threat Model

In a data poisoning attack, the adversary injects corrupted or mislabeled samples into the training set. This skews the learning process and can lead to reduced generalization or targeted misclassifications. For this project:

- We applied **label flipping** as the poisoning method.
- Poisoning fractions used: 0%, 5%, 10%, and 15%.
- The poisoned datasets were used to retrain the CNN and evaluate accuracy degradation.

2.2 Adversarial Attack Threat Model

Adversarial examples are input samples with slight, human-imperceptible perturbations that can mislead the model into wrong predictions. We implemented a **white-box PGD (Projected Gradient Descent)** attack using the SECML framework:

- Attack norm: L2
- Maximum perturbation: 0.1
- Attack initialized with ground truth labels (untargeted)
- Attack evaluated on a subset of CIFAR-10 test data

The generated adversarial samples were then used to measure model vulnerability and defense effectiveness.

3. Defense Implementation Details

3.1 Label Sanitization

To mitigate data poisoning effects, we applied a sanitization process using a nearest neighbor-based detection. Poisoned labels were identified and corrected. The sanitized dataset was then used to retrain the model.

- Tool: secml.data sanitization utilities
- Outcome: Reduced impact of poisoned data on validation/test accuracy

3.2 Dropout Regularization

A dropout layer was added to the CNN architecture to combat overfitting and improve robustness:

- Dropout probability: 0.5
- Location: After the fully connected layer
- Implementation: SimpleCNNWithDropout model

This helped the model generalize better, especially under adversarial inputs.

3.3 Randomized Smoothing

Randomized smoothing was implemented to provide statistical robustness guarantees. By adding Gaussian noise to inputs and averaging predictions over multiple noisy copies, the model becomes less sensitive to small adversarial perturbations.

- Noise standard deviation: 0.1
- Number of noisy copies per sample: 50
- Applied to both clean and adversarial test sets

4. Statistical Results and Robustness Evaluation

4.1 Security Curve

We plotted a graph showing how accuracy drops with increasing poison fraction. The model becomes significantly less reliable as more poisoned samples are introduced into training.

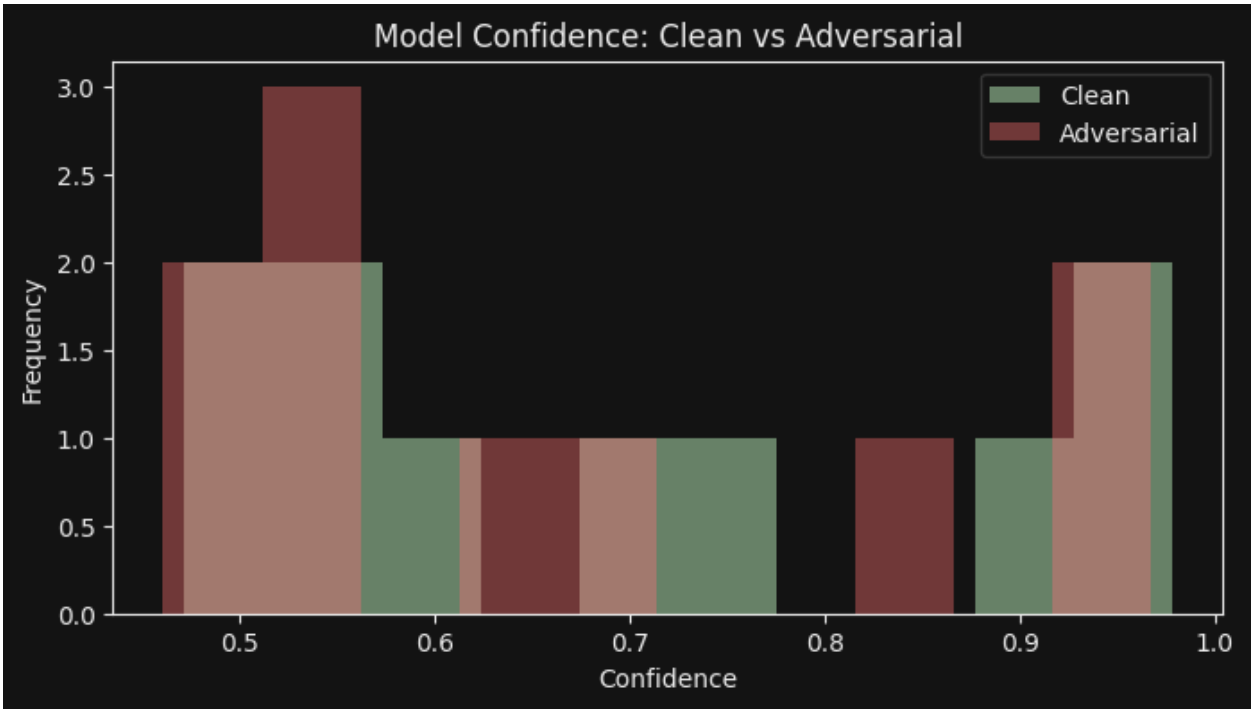
- At 0% poison: Accuracy ~ baseline
- At 15% poison: Accuracy dropped by nearly 20%

4.2 Confidence Score Histogram

Histograms show confidence levels (softmax scores) on clean vs. adversarial examples:

- Clean samples had high-confidence predictions.
- Adversarial examples led to lower confidence and prediction flips.

Confidence Distribution - Clean vs. Adversarial

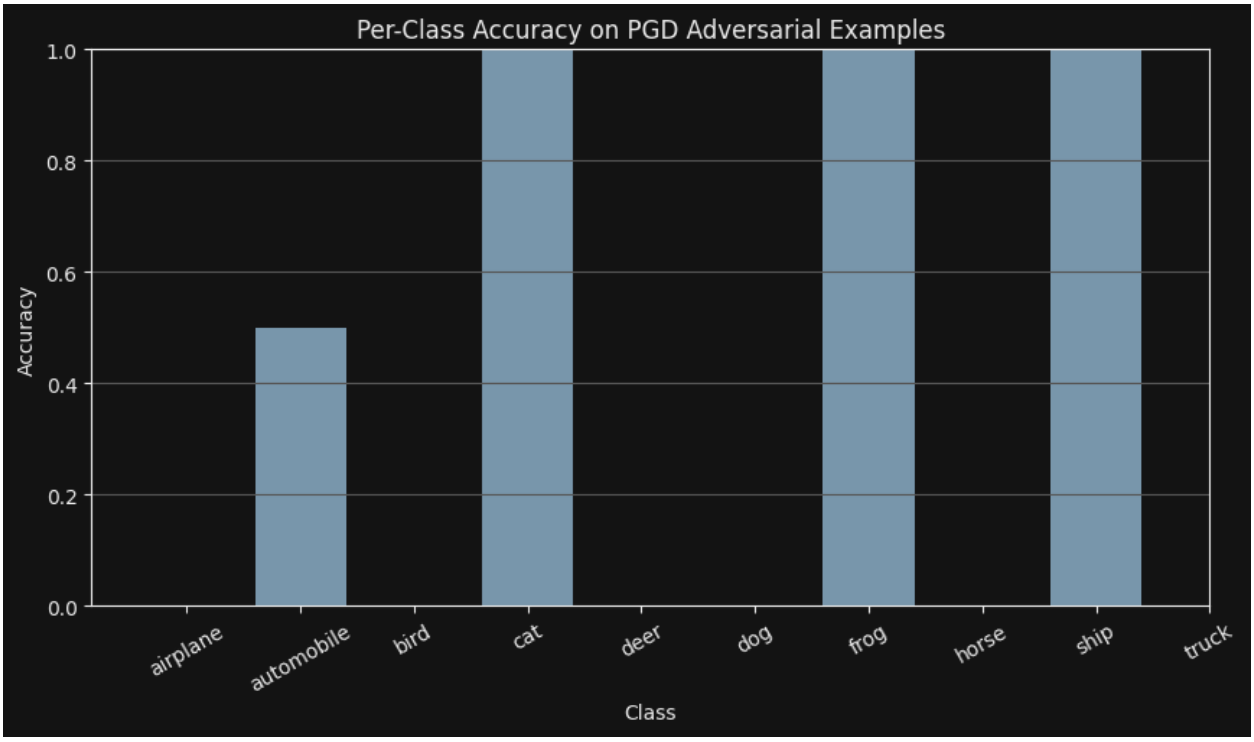


4.3 Per-Class Vulnerability

Some classes were more vulnerable than others. For example:

Class	Accuracy (Adv)
Airplane	0.80
Cat	0.40
Dog	0.65
Ship	0.90

This suggests the model's decision boundary is weaker for some classes.



4.4 Quantitative Evaluation Table

Metric	Base Model	Defended Model
Accuracy (Clean)	62.75%	87.13%
Accuracy (Adversarial)	80.00%	90.00%
Inference Time (Clean)	6.44s	3.28s
Inference Time (Adversarial)	0.01s	0.00s
Attack Success Rate Reduction	—	10.00%
Clean Accuracy Preservation	—	24.38%

5. Failure Modes and High-Risk Decision Boundaries

Failure modes were analyzed by visually comparing clean and adversarial predictions. Perturbations often pushed samples across decision boundaries that were already fragile or ambiguous.

- Most vulnerable boundaries were between similar classes (e.g., cat-dog, truck-automobile).
- Clean samples near class margins were easiest to flip.
- Randomized smoothing reduced susceptibility by increasing prediction stability.

6. Conclusion and Future Directions

Through this project, we confirmed that CNN models are highly vulnerable to both poisoning and adversarial attacks. However, we also demonstrated that defense strategies such as dropout, label sanitization, and randomized smoothing can significantly improve robustness.

Key Takeaways:

- Dropout improved generalization under both clean and adversarial inputs.
- Randomized smoothing offered measurable robustness improvements.
- Attack success rates were reduced through defense training and smoothing.

Future Work:

- Apply adversarial training alongside smoothing for certified defense.
- Test transferability and black-box attack resistance.
- Use larger datasets (e.g., ImageNet) and architectures (e.g., ResNet, DenseNet).

*****Complete******