# Two-Part Project Focusing on Threat Mitigation in ML Systems

## Part 1: Detecting Data Poisoning and Adversarial Attacks in Machine Learning

In this part of the project you investigate training-time (data poisoning) and inference-time (adversarial) vulnerabilities in ML models through controlled experiments. You will work on a dataset, build a machine-learning model, test it for vulnerabilities.

## -Part 1 Phases:

### **Phase 1:** Dataset Selection and Preprocessing

1. Select a suitable dataset for your project. It could be a standard dataset like MNIST, CIFAR-10, or one relevant to your domain.
2. Implement preprocessing pipeline: Normalization, train-test splits (70-30

### **Phase 2:** Building a Machine Learning Model

1. Choose a machine learning model architecture (deep neural network, convolutional neural network…).
2. Train on clean data with validation-based early stopping
3. Establish baseline performance metrics (Accuracy and Confusion Matrix)

### **Phase 3:** Training-Time Attacks (Data Poisoning)

1. **Poisoning Attack Implementation**
   * Inject malicious samples into training data using one of:
   * Label-flipping attacks
   * Any of the clean-label backdoor attacks
   * maintain attack budget (<15% training data contamination)
2. **Poisoned Model Evaluation**

* Retrain model on contaminated dataset
* Compare performance degradation on:
  + - Clean test set
    - Poisoned validation samples
    - Original validation set

### **Phase 4:** Inference-Time Attacks (Adversarial Examples)

1. **Adversarial Attack Generation**

* Implement two distinct attack methods:
  + - **White-box**: FGSM/PGD/C&W/DeepFool
    - **Black-box**: Surrogate model
* Generate adversarial test sets with controlled perturbation budgets (ε ≤ 0.1)

1. **Attack Impact Analysis**

* Quantify robustness drop using:
  + - Adversarial success rate
    - Confidence score distributions
    - Per-class vulnerability analysis

### **Phase 5:** Comprehensive Evaluation

1. **Cross-Attack Susceptibility**
   * + Test poisoned model against unseen attack vectors
     + Analyze transferability between attack methods
2. **Vulnerability Report**
   * + Create visualization: Security Curve for accuracy with both perturbations number and number of poisoned samples.
     + Document failure modes and high-risk decision boundaries

## Part 2: Defending Against Data Poisoning and Adversarial Attacks in Machine Learning

This part aims to develop defenses to safeguard the mode from Part 1

**-Part 2 Phases:**

### **Phase 1:** Poisoning Defense Implementation

1. Choose one Data Sanitization Techniques for example:

* Implement anomaly detection (Isolation Forest/MAD)
* Apply spectral signature analysis for poisoned sample removal

1. And one method of Robust Training Methods for example:

* Integrate regularization (Dropout/Weight Clipping)
* Explore differentially private training

**Phase 2:** Adversarial Defense Strategies

1. Input Preprocessing Defenses

* Test randomized smoothing techniques

1. Model Hardening

* Apply adversarial training with PGD examples
* Explore certified robustness methods (IBP/RS-Certify)

### **Phase 3:** Defense Evaluation

1. Quantitative Analysis

* Compare metrics before/after defenses:
  + - Clean data accuracy preservation
    - Attack success rate reduction
    - Computational overhead

1. Qualitative Analysis

* Visualize decision boundary changes
* Conduct gradient sensitivity analysis

### **Phase 4:** Reporting & Advanced Exploration

1. Documentation Requirements

* Technical report (5-10 pages) covering:
  + - Threat models & attack mechanics
    - Defense implementation details
    - Statistical evidence for robustness claims

## Submission Requirements

Submit the following file in a zipped folder to the project submission folder.

* Complete Python implementation with modular codebase
* Final report PDF following academic paper format or Presentation deck (technical & non-technical versions)

**Grading Criteria:**

|  |  |
| --- | --- |
| Criteria | Mark |
| Technical Depth & Methodology | 30% |
| Defense Effectiveness Metrics | 25% |
| Analysis & Critical Evaluation | 20% |
| Code Quality & Reproducibility | 15% |
| Presentation Clarity & Engagement | 10% |