Task: Adversarial Robustness Framework Upgrade

This document tracks the progress of upgrading the project to a research-grade adversarial robustness evaluation framework.

- **Phase 1: Codebase Audit & Setup**
  - Explore current file structure
  - Locate train_model.py and understanding data flow
  - Install dependencies (`adversarial-robustness-toolbox`, `torch`)
- **Phase 2: Core Attack Framework Implementation**
  - Create src/adversarial/adversarial_evaluation.py
  - Implement data loading & model wrapping (ART)
  - Implement Black-box attack generation (HopSkipJump)
  - Implement Surrogate Model training (PyTorch)
  - Implement White-box transfer attacks
- **Phase 3: Integration & Testing**
  - Modify training script to export clean test sets
  - Run initial vulnerability assessment (Baseline ASR)
  - Implement defense layers (Ensemble, Thresholding)
  - Measure defense effectiveness (ASR reduction)
- **Phase 4: Advanced Research & Reporting**
  - Implement SHAP Stability Monitoring
  - Adversarial Training (Applied to evaluation logic)
  - Final Comparative Analysis & Walkthrough
- **Phase 5: Codebase Cleanup & Optimization**
  - Delete redundant directories (New folder, Research_Antigravity, Documentations)
  - Remove redundant data files (models/X_test_sample.csv)
  - Final verification of core system functionality
- **Phase 6: Research Rigor & Statistical Validation (Audit Fixes)**
  - Fix Surrogate Data Leakage (Anti-leakage training)
  - Implement Feature Clipping (Realistic constraints)
  - Add Perturbation Norm Analysis (L2/Linf)
  - Report Robust Accuracy & ASR with high budget (max_iter=50)
  - Formalized Threat Modeling and Security Logging