CS351 Project Specification
Evaluating the Robustness of ML-Based Fraud Detection via Game Theory

Daniel Lai 5529114

October 2025

1 Problem Statement

As technology becomes increasingly integrated into daily life, the benefits of improved convenience and service delivery are evident. However, this rapid adoption sometimes leads to overlooked security fundamentals, especially within older or legacy systems [2]. Nowhere is this more critical than in the financial sector, where the stakes are exceptionally high and sensitive information is constantly at risk [4].
The landscape of financial fraud is constantly evolving, with skilled attackers constantly developing new methods to bypass and compromise security mechanisms. Many current detection systems, built on traditional machine learning techniques, are vulnerable to advanced adversarial attacks - including poisoning (manipulating training data) and evasion (crafting inputs to deceive models) [1, 5]. As a result, these systems face threats that their original designs did not anticipate.
Evaluating fraud detection models solely on static datasets is therefore insufficient. Recent work has shown that adversarial training and dynamic evaluation frameworks can significantly enhance model robustness against worst-case perturbations, underscoring the need for adaptive defence strategies [3]. Consequently, this project will model the strategic, iterative interactions between adversaries and defenders using a game-theoretic framework, with the goal of developing more resilient and practically deployable fraud detection systems.

2 Objectives

This project aims to evaluate the robustness of machine-learning based fraud detection systems within a dynamic, adversarial context. By creating a simulated environment, it will model the strategic, iterative interaction between an adaptive attacker and an ML-based defender using a game-theoretic framework.

To achieve this, this project is broken down into a set of core deliverables and potential extensions:

2.1 Core Objectives

Phase 1: Foundational Components

1. **Define a Specific Use Case**: Research and select a specific domain of financial fraud (e.g., credit card transaction fraud, insurance claim fraud) to serve as the context for the simulation. This will define the features and patterns to be modelled.

2. **Develop a Simulation Environment**: Build a Python simulator to generate a realistic time-series dataset of legitimate financial transactions based on defined user personas.

3. **Implement a Baseline ML Defender**: Train and benchmark machine learning classifiers on the simulated data to establish an effective baseline detection model.

Phase 2: Adversarial Analysis

4. **Design an Adaptive Adversary**: Create an attacker agent capable of applying subtle, evasive attack strategies designed to bypass the defender model.

5. **Evaluate and Visualise System Robustness**: Measure the defender model's performance degradation under adversarial attacks to identify and analyse its key vulnerabilities.

2.2 Possible Extensions

1. **Apply Game-Theoretic Analysis**: Model the attacker-defender interaction as a strategic game, analysing simulation results to identify potential equilibrium strategies.

2. **Model a Defender Retraining Loop**: Implement a feedback mechanism allowing the defender to retrain on detected attacks, simulating a dynamic security "arms race".

3 Methods & Methodology

This project will use an iterative and incremental development model, along with some agile properties. This approach will combine a structured, phased development process for building the system components with a formal research framework for the experimental analysis.

3.1 Development

- The system will be built in stages. First, a simple simulator with one user persona will be developed. This will be followed by the baseline defender, and so on. This ensures that there is always a working, testable version of the system.

- Each component will be refined through a cycle of design to implementation to testing and lastly refining. Initial results may show that the simulated transactions are too simplistic. The simulator will then be refined in a new iteration to add more complexity. This iterative process would be useful as the characteristics may evolve based on initial findings

3.2 Research

- A literature review will be conducted to identify realistic user spending patterns and common evasive attack strategies. This will ensure the simulation is as realistic to real-world scenarios.

- The simulator provides a controlled environment to test the hypothesis that ML models are vulnerable to specific evasive attacks. The independent variable will be the type and intensity of the attack, and the dependent variable will be the performance of the defender model

- Performance metrics (accuracy, precision, recall, F1-score, and so on) will be compared before and after attacks. Statistical significance will be assessed where applicable. The findings from the models' performance will be statistically analysed to draw conclusions about the model's vulnerabilities and the effectiveness of different attack strategies.

3.3 Project Management

- Git will be used for version control to manage source codes with Github to track changes and prevent data loss.
- Regular meetings and check-ins with the Project Supervisor will serve as checkpoints to discuss progress, advise and validate the project's direction.
- A logging document to track accomplishments, progress and any issues that arose during the implementation of the project's objectives.

4 Timetable

The table below shows this project's planned rough timeline, including all of the hard deadlines for the submissions. The durations in the timetable for the whole project are thought to be overestimated to allow slack for any unforeseen issues that may occur.

Term / Week	**Objective / Deadline**
T1 W1-2	Initial Literature Review & Finalise Plan **Project Specification Due (16th October)**
T1 W3-6	**Phase 1**: Implement Transaction Simulator Design user personas, generate transaction data, and test output.
T1 W7-10	**Phase 1**: Implement & Train Baseline ML Defender Benchmark model performance on clean, simulated data.
Winter Break	Buffer time for delayed objectives Start Final Report
T2 W1-2	Finalise Progress Report **Progress Report Due (22nd January)**
T2 W3-5	**Phase 2**: Implement Adversarial Attacker Design and code evasive attack strategies.
T2 W6-9	**Phase 2**: Run Robustness Evaluation Experiments Collate results and begin analysis. **Final Report Plan Due (13th March)**
Break	**Phase 3**: Intensive Write-up Period Finalise Final Report and Project Management Report. **Final Report Due (9th April)** **Project Management Report Due (9th April)**
T3 W1-2	**Phase 4**: Viva Preparation Prepare presentation slides, potential demo, and practice.

Table 1: Project Timetable

5 Resources & Risks

5.1 Resources

The project will be developed using standard software and hardware, with no specialised equipment required.

- **Hardware:** A personal computer with sufficient processing power and memory for training standard machine learning models.
- **Software & Libraries:** The development will be done primarily in Python. Key libraries will include:
  - `Pandas` and `NumPy` for data generation and manipulation.
  - `Matplotlib` and `Seaborn` for creating charts and visualisations of model performance and attack affectiveness.
  - Standard machine learning libraries for implementing the machine learning models. Some examples include `Scikit-learn` and `PyTorch`
- **Data Source:** All transaction data will be synthetically generated by the custom-built simulator developed in Phase 1 of the project. This avoids the ethical and privacy constraints associated with real financial data.
- **Version Control & Documentation:** Source code will be managed with Git and hosted on a private GitHub repository. The project specification, reports, and final dissertation will be written in LaTeX using Overleaf.
- **Storage:** Data sources, articles for reference, and other relevant documentation will be stored and backed up via Google Drive.

5.2 Risks

A number of potential risks have been identified. The table below outlines these risks and the mitigation strategy for each.

Risk	Mitigation Strategy
Simulation Lacks Realism: The synthetic data generated is too simplistic, making the fraud detection task trivial and undermining the validity of the results.	The simulator's design will be based on statistical properties found in real-world public datasets. The development will be iterative, allowing for refinement if initial results are not realistic enough.
Technical Hurdles in Development: Unforeseen bugs or complexities in developing the simulator or ML pipeline could cause significant delays.	The project will follow an iterative development model, allowing for early detection of issues. Buffer time is explicitly allocated in the project timetable to absorb minor delays.
Loss or Failure of Equipment: Hardware failure could lead to the loss of critical work and code.	All source code and documentation will be regularly backed up to remote cloud repositories (GitHub and Overleaf).
Insufficient Model Performance: The baseline ML models fail to achieve reasonable fraud detection accuracy, making robustness evaluation less meaningful and undermining the validity of adversarial attack testing.	Perform systematic hyperparameter tuning and feature engineering. If performance remains inadequate, revisit dataset selection and preprocessing steps.

Table 2: Project Risks & Mitigation Strategies

6 Legal, Social, Ethical and Professional Issues

This project involves the evaluation of machine learning-based financial fraud detection systems under adversarial conditions. The implementation of adversarial attacks will be carefully designed to ensure that it poses no threat to real-world systems or society. All experiments will be conducted in a controlled, academic environment using only synthetic data and simulated financial scenarios. The attack code will not be capable of targeting or interacting with any external or real financial systems. The project's intent is clearly documented as defensive research to improve system robustness, not to create malicious tools.

References

[1] Adversarial learning in real-world fraud detection. *ACM Digital Library*, 2023.

[2] Financial fraud detection through the application of machine learning. *Nature*, September 2024.

[3] Long Dang, Thushari Hapuarachchi, Kaiqi Xiong, and Jing Lin. Improving machine learning robustness via adversarial training. *arXiv preprint arXiv:2309.12593*, 2023.

[4] Prove. Implementing fraud detection for financial institutions. `https://www.prove.com/blog/financial-fraud-detection-challenges`, October 2025.

[5] Wipro. Making ml models in banking resilient using adversarial attacks, December 2022.