Adversarial Simulation for Enhancing Credit Card Fraud Detection Models

Yixin Luo, Junling Shao

1. Introduction

Credit card fraud is a persistent and evolving threat in digital finance, costing the global economy an estimated $5.1 trillion annually. As financial services become increasingly digitized, fraud detection systems play a crucial role in preserving trust, ensuring consumer safety, and promoting financial inclusion. However, these systems must strike a delicate balance between flagging suspicious transactions and minimizing false positives to avoid excluding legitimate users, particularly those from underserved populations.

Traditional machine learning models have been widely adopted to detect fraudulent transactions by learning patterns from historical data. Despite their success, these models are vulnerable to adversarial manipulation -- small, targeted modifications to input features that can lead to misclassification. Such vulnerabilities threaten the reliability of fraud detection systems and raise serious concerns about their deployment in high-stakes environments.

This project explores the use of adversarial simulation in tabular transaction data to test the robustness of credit card fraud detection models. While adversarial machine learning is a well-developed field in image recognition, its application in structured financial data remains underexplored. Our goal is to identify and exploit model weaknesses through simulated attacks, thereby advancing the development of more secure, adaptive fraud detection systems suitable for real-world deployment.

2. Literature Review / Related Work

As machine learning (ML) becomes more integrated into real-world systems, including financial fraud detection, concerns have emerged regarding the robustness of these models against adversarial attacks. Adversarial machine learning research has revealed that small, strategically crafted perturbations in input data can cause ML models to make erroneous predictions, posing significant risks in security-critical domains such as finance and autonomous systems.

Early demonstrations of adversarial attacks were largely confined to the domain of image recognition, but recent studies have extended these insights to structured, real-world environments. For example, Tuncali et al. (2018) explored adversarial testing in autonomous vehicles by generating corner-case inputs through simulation, revealing how ML-integrated control systems can fail under unexpected input conditions. Their work highlights the utility of simulation-based adversarial generation in high-stakes applications where empirical data is difficult or risky to obtain—a logic that also applies to financial fraud detection, where real adversarial fraud attempts are rarely observable.

While gradient-based attacks (e.g., FGSM, PGD) dominate academic literature, Apruzzese et al. (2023) argue that such techniques misrepresent the behavior of real-world attackers. In practice, adversaries are unlikely to compute gradients but instead use heuristic, domain-informed strategies. This insight is particularly important for fraud detection, where attackers exploit known vulnerabilities or behavior thresholds in transaction systems, not mathematical loss gradients. Therefore, evaluating model robustness requires testing with adversarial strategies that reflect realistic threat models rather than purely theoretical ones.

In the financial domain, researchers have begun leveraging adversarial techniques for both model evaluation and improvement. Fiore et al. (2019) used generative adversarial networks (GANs) to augment training datasets and improve classification accuracy in credit card fraud detection. By synthesizing realistic fraudulent examples, they showed that GANs can help mitigate class imbalance and expose model weaknesses, thus serving as a dual-purpose tool for data enhancement and adversarial simulation.

More recently, Tsai et al. (2024) proposed a framework for identifying adversarial examples within credit card transaction datasets. They demonstrated that even tabular models trained with traditional defenses remain vulnerable to perturbation-based attacks that subtly alter transaction features. Their findings suggest that current fraud detection systems lack sufficient safeguards against adaptive adversarial manipulation, especially in black-box contexts.

Together, these studies underscore a pressing challenge: although fraud detection models may perform well under standard testing conditions, they can fail catastrophically under adversarial scenarios that reflect realistic attacker behavior. This motivates our project's core objective—to simulate adversarial inputs on tabular financial data and evaluate the robustness of credit card fraud classifiers under such conditions. By drawing on insights from both simulation-based testing and financial adversarial learning, we aim to bridge the gap between controlled research environments and real-world deployment.

3. Methodology

This section outlines the technical and procedural design of the fraud detection project. It describes the end-to-end system workflow, the process for building the detection model, the design of the adversarial simulation, and the software tools used throughout the implementation.

3.1 System Architecture and Workflow

The workflow of our project is structured into three sequential stages: data storage and preprocessing, analytics and machine learning, and final result management. At the initial stage, raw credit card transaction data is ingested and stored in Snowflake, a cloud-native data warehouse optimized for large-scale analytics. SQL queries are used to clean and organize the data into structured tables, preparing it for downstream analysis. This preprocessing step ensures consistency, removes redundancies, and extracts relevant features from the raw dataset.

Once the data is cleaned and structured, we transition to the analytics and modeling phase. This stage is conducted in Python, leveraging libraries such as pandas and scikit-learn for exploratory data analysis and model development. The computations are executed on Amazon EC2, providing the flexibility and scalability needed for training and evaluating machine learning models on large tabular datasets. Visualization tools are used throughout this stage to interpret trends in the data and to evaluate model behavior during both standard and adversarial testing.

In the final stage, all outputs—including trained model files, predictions, and processed datasets—are saved to Amazon S3 for durable cloud storage. Select outputs are also written back to Snowflake for integration with other analytics workflows. To ensure reproducibility and collaborative development, all code, experimental logs, and documentation are version-controlled and shared via GitHub. This modular architecture not only supports efficient experimentation but also allows for transparent, traceable deployment of fraud detection models in production-like environments.

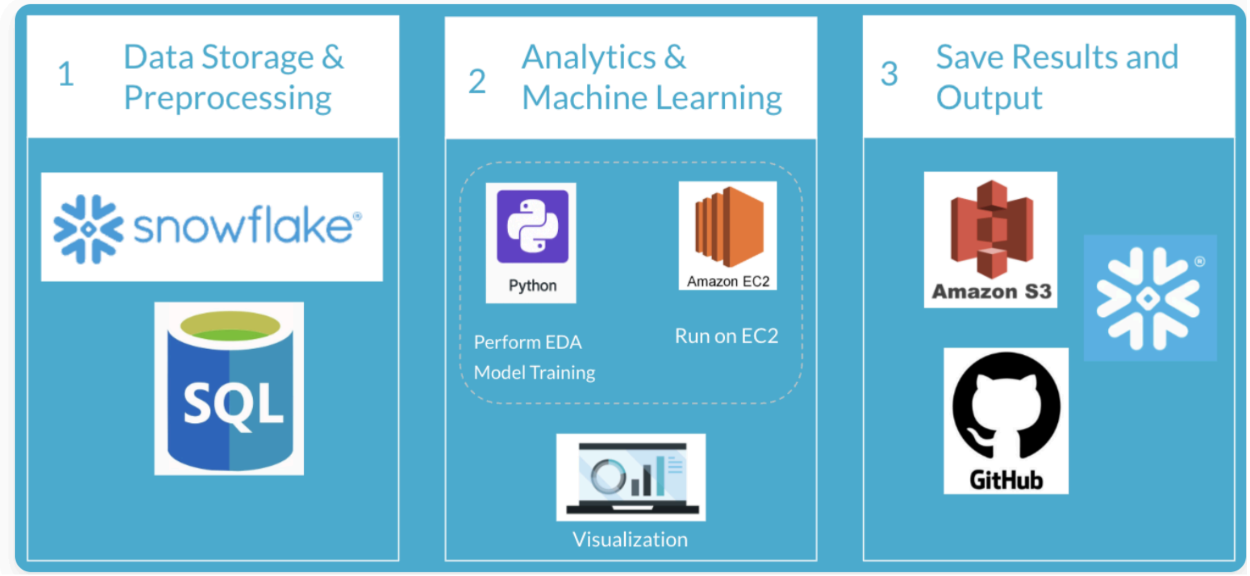

Figure 1: Project Workflow

3.2 Fraud Detection Model Development

The fraud detection system is built using supervised machine learning on labeled transaction data. To ensure robustness in model performance, the dataset is first partitioned into an 80/20 train-test split, preserving the highly imbalanced nature of fraud labels. Given that only approximately 0.5% of all transactions are fraudulent, model development requires careful attention to class imbalance. This is addressed using class weighting in the loss function and threshold tuning based on precision-recall trade-offs.

The primary model used in the initial stage is logistic regression, selected for its interpretability and speed. As the project progresses, additional models such as random forests and gradient-boosted trees are explored to compare baseline performance under both normal and adversarial conditions. Standard classification metrics, including precision, recall, F1-score, and area under the ROC curve (AUC), are used to evaluate model performance, particularly in detecting the rare class of fraudulent transactions.

3.3 Adversarial Simulation Design

To assess the vulnerability of fraud detection models, the project implements a simulation-based adversarial testing framework. This approach generates modified input samples that aim to evade detection without altering the semantic meaning of the transaction. Unlike gradient-based attacks commonly found in image classification research, this project adopts a black-box strategy more aligned with real-world fraud behavior, where attackers do not have access to model internals.

Perturbations are applied to a set of features deemed most critical for classification, such as transaction amount, merchant category, and city population. These features are modified using heuristic rules to mimic plausible, evasive behavior, such as slightly decreasing the transaction amount or altering job titles. The adversarial examples are then passed through the trained model to observe changes in classification outcomes. A significant drop in fraud detection performance on these modified samples is used as evidence of model fragility and highlights areas for improvement in future defense strategies.

3.4 Tools and Packages

The project is implemented using Python and SQL. Data manipulation and preprocessing are handled using pandas and NumPy, while model training and evaluation are conducted using scikit-learn and XGBoost. Label encoding and feature normalization are applied using built-in functions such as LabelEncoder and StandardScaler. The infrastructure relies on Amazon EC2 for computational tasks and Amazon S3 for output storage, with Snowflake serving as the primary data warehouse. All project code, notebooks, and documentation are maintained in a GitHub repository to ensure transparency and reproducibility.

4. Dataset and Exploratory Data Analysis (EDA)

To better understand the transaction data and uncover patterns associated with fraudulent behavior, we conducted a comprehensive exploratory data analysis (EDA). The visualizations below offer insights into feature distributions, relationships, and fraud prevalence across key dimensions.

4.1. Correlation Analysis

We computed the Pearson correlation matrix among all numeric features. As shown in the heatmap, most features exhibit very low linear correlation with one another, suggesting minimal multicollinearity. One notable exception is the strong positive correlation between TX_DAYOFWEEK and IS_WEEKEND (r ≈ 0.82), which is expected due to shared semantics.

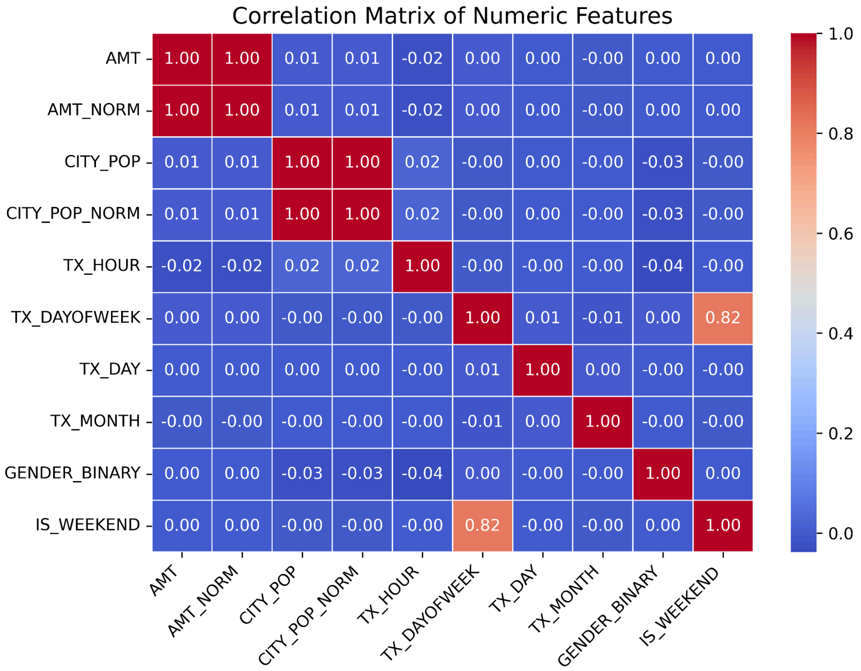

Figure 2: Correlation Matrix of Numeric Features

4.2. Transaction Amount Distribution

The distribution of transaction amounts is highly skewed. After applying a log transformation (log(AMT + 1)), the distribution becomes bimodal and more symmetric, revealing underlying groupings of small-scale versus large-scale transactions. This transformation is essential for stabilizing variance and improving model performance.

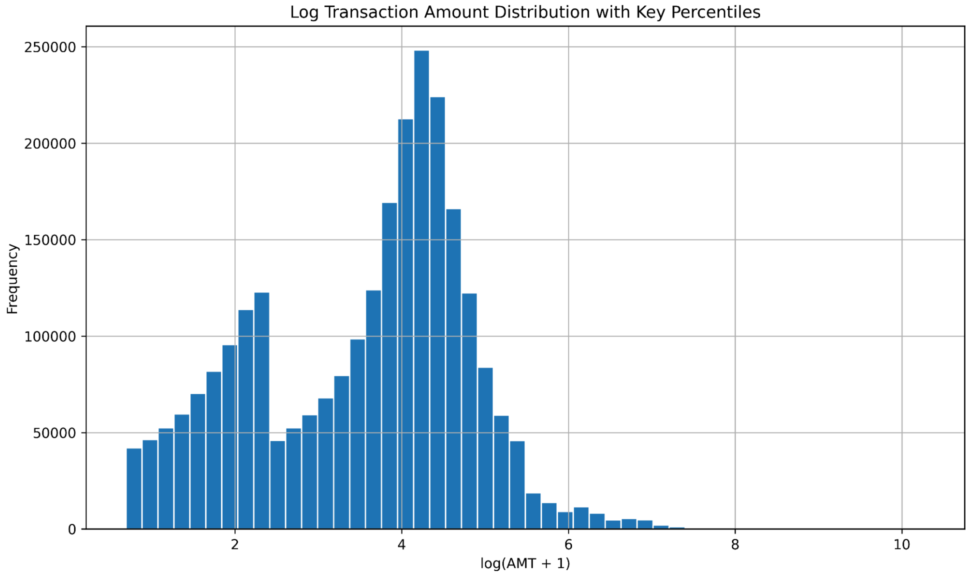

Figure 3: Log Transaction Amount Distribution with key Percentiles

4.3. Fraud Rate by Transaction Category

Certain transaction categories exhibit disproportionately high fraud rates: Online shopping (shopping_net) has the highest fraud rate at 1.65%, followed by miscellaneous online and grocery transactions. These categories may represent higher-risk segments and suggest areas for heightened monitoring or model weighting.

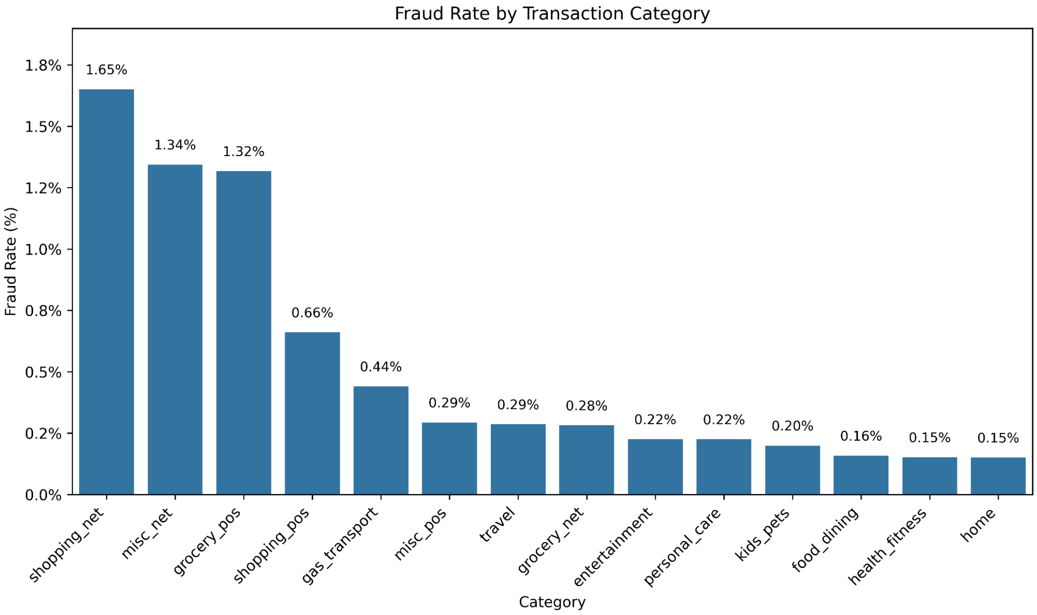

Figure 4: Fraud Rate by Transaction Category

4.4. Fraud Rate by Day of the Week

Fraud rates vary by weekday, specifically Thursday to Saturday consistently show higher fraud rates (above 0.6%), while Tuesday and Monday see the lowest rates (below 0.5%). This temporal pattern may reflect behavioral or operational vulnerabilities that fraudsters exploit during busier end-of-week periods.

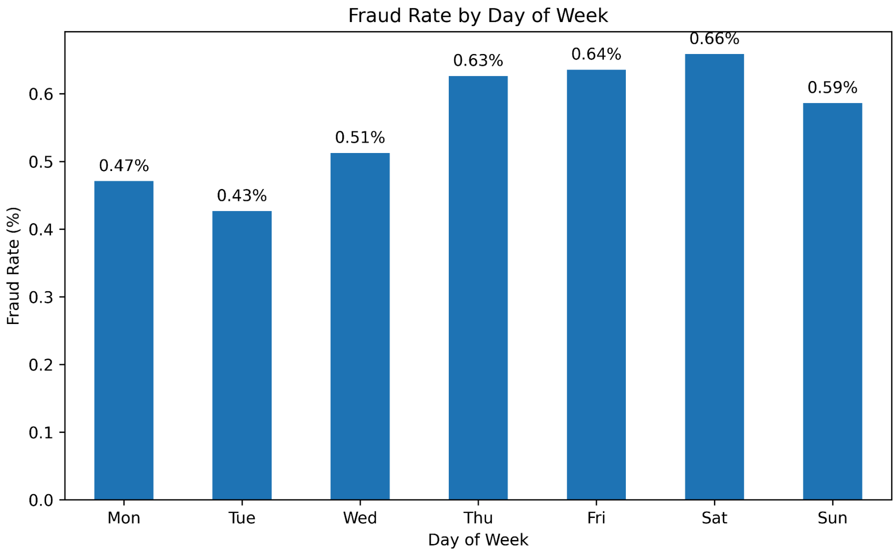

Figure 5: Fraud Rate by Day of Week

4.5. Fraud Rate by Hour of the Day

Fraudulent activity shows strong temporal clustering at night, a clear spike in fraud is observed between 10 PM and midnight (rates up to 2.69%), and a smaller secondary spike occurs between 12 AM and 3 AM. Therefore, in our cases, fraud is more likely during off-hours, possibly due to decreased user vigilance or system monitoring.

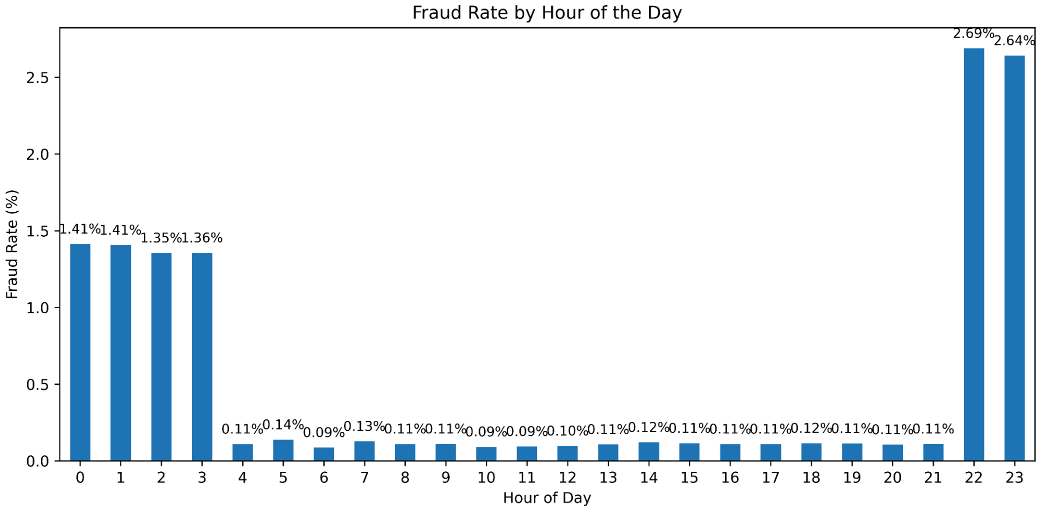

Figure 6: Fraud Rate by Hour of the Day

5. Modelling

To detect fraudulent credit card transactions, we built and evaluated several supervised machine learning models using structured tabular data. The cleaned dataset was retrieved directly from a Snowflake warehouse and included rich transactional features such as transaction amount, time metadata, location, merchant category, user job, and demographics. After removing irrelevant columns and filtering rows with missing target labels, we partitioned the data into training and testing sets based on the 'dataset' column. This allowed us to simulate a time-aware evaluation setup by separating earlier transactions (train_1 and train_2) from later ones (test).

Preprocessing involved multiple steps to ensure compatibility with scikit-learn models and to improve training performance. Numerical features were standardized using z-score normalization, while missing values were imputed using column means. Categorical variables such as merchant name, job title, and transaction category were label encoded, ensuring that consistent mappings were applied across train and test data. For all models, we retained the full set of features and did not perform dimensionality reduction, opting instead to assess model capacity in handling real-world feature richness.

We evaluated six machine learning classifiers: logistic regression, random forest, multi-layer perceptron (MLP), decision tree, k-nearest neighbors (KNN), and naive Bayes. A dummy classifier that always predicted the majority class was included as a performance floor. All models were trained using default or minimally tuned hyperparameters, with class weighting enabled where supported to account for the significant imbalance in the fraud class. The primary evaluation metrics included accuracy, precision, recall, F1-score, and area under the ROC curve (AUC), with a focus on recall and F1 as the fraud class was the positive label and extremely underrepresented.

The results highlight clear performance differences across models. Random forest achieved the best overall performance, with a precision of 0.156, a recall of 0.518, and an F1-score of 0.240. MLP followed closely, showing strong recall and a balanced F1-score. Logistic regression offered good recall (0.492) but struggled with precision, resulting in a lower F1-score. Tree-based models such as decision trees and random forests performed well, but the single decision tree model was outperformed by the ensemble. Naive Bayes and KNN delivered poor results, with F1-scores under 0.20, revealing their limitations in handling high-dimensional, imbalanced data. As expected, the dummy classifier performed the worst across all metrics, validating that meaningful learning occurred in the other models.

These results informed the selection of models for the adversarial robustness stage, where we further stress-tested model performance under feature manipulation scenarios. In the next section, we present our adversarial simulation design and the resilience of each classifier under targeted perturbations.

6. Adversarial Modeling: Feature Perturbation

To evaluate the robustness of our fraud detection models against intentional manipulation, we implemented a manual feature perturbation framework designed to simulate adversarial behavior. In realistic fraud scenarios, malicious actors may modify subtle aspects of a transaction, such as reducing the distance from home or slightly shifting the transaction time, in hopes of avoiding detection. To test our models under such conditions, we systematically altered three high-impact features in the test set: amt (transaction amount), hour (transaction time), and distance_from_home.

Each feature was modified using domain-informed rules. The transaction amount was increased by 10%, simulating cases where fraudsters attempt slightly higher-value purchases. The hour of transaction was shifted forward by three hours modulo 24, mimicking time spoofing. The distance_from_home feature was scaled down by 20%, reflecting tactics used to impersonate proximity to the cardholder's typical behavior.

After applying these changes to the test set, we re-evaluated each model to measure performance degradation. The results were revealing. Logistic regression maintained relatively stable recall (dropping from 0.7408 to 0.7441), though its already low precision further declined, reducing its F1-score from 0.1359 to 0.1276. Random forest experienced a more pronounced drop in recall (from 0.6275 to 0.5660), with its F1-score decreasing by nearly 0.05. MLP proved highly resilient, with only minimal changes in both recall and F1-score, suggesting strong generalization to slight feature noise.

In contrast, models such as decision trees, naive Bayes, and KNN showed considerable vulnerability. The decision tree’s F1-score for the fraud class dropped from 0.6719 to 0.5396, while naive Bayes showed consistent underperformance regardless of input changes. KNN, already underperforming on the clean test set, failed to recover under perturbation, with its fraud class F1-score remaining extremely low (dropping from 0.0330 to 0.0420).

These findings underscore the importance of robustness testing in fraud detection systems. High performance on static test data can mask fragility to small, malicious changes in input features. While models like random forest and MLP offer strong baseline performance, only MLP demonstrated both high accuracy and resilience under adversarial simulation. This highlights its promise for deployment in real-world financial systems where data volatility and evasion attempts are constant threats.

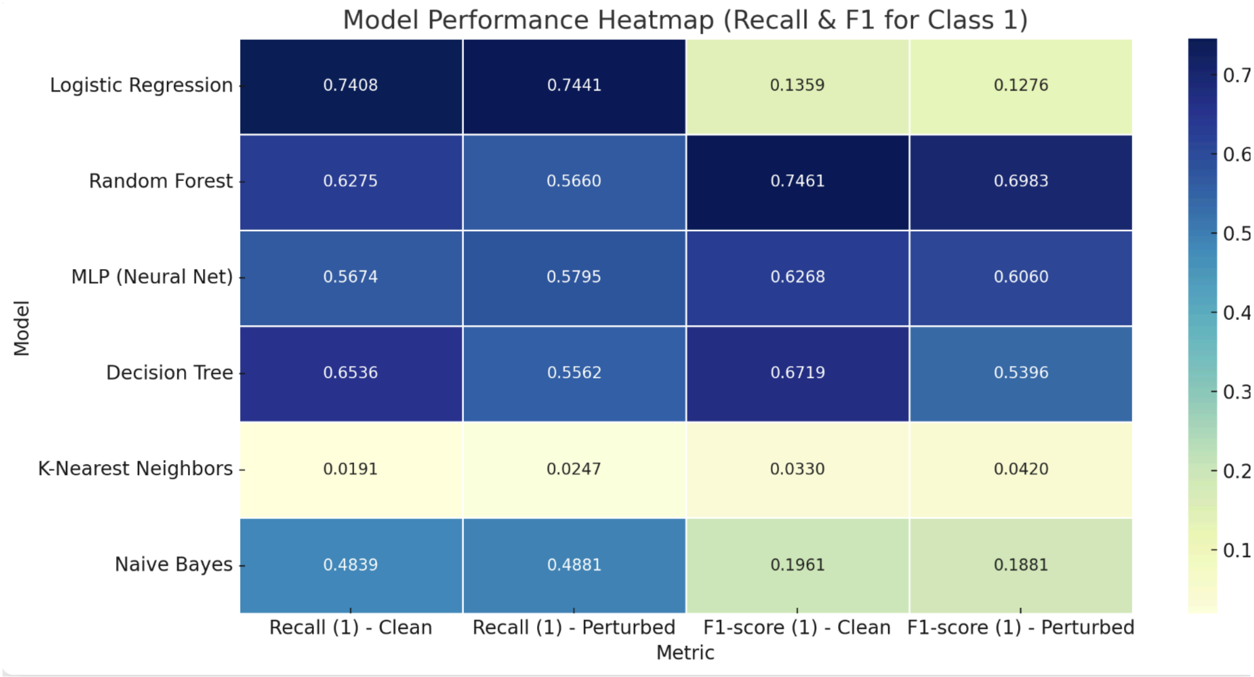

Figure 7: Model Performance Heatmap (Recall & F1 for Class 1)

7. Results & Discussion

Our evaluation highlights the limitations of relying solely on accuracy in fraud detection tasks with severe class imbalance. While all models achieved high overall accuracy, only a few provided meaningful recall and F1-scores for the minority fraud class. Random forest emerged as the top performer on clean data, offering the best balance of precision and recall. MLP (neural net) also performed well, demonstrating strong generalization and slightly better robustness under feature perturbation. Logistic regression, though efficient and stable in recall, suffered from low precision and a correspondingly low F1-score.

Models such as KNN, naive Bayes, and decision trees showed poor baseline performance and significant vulnerability to manually perturbed inputs. In contrast, MLP proved most resilient, maintaining high recall and only marginal degradation in F1-score, even when fraud-relevant features were slightly manipulated. Random forest remained relatively stable, but its recall declined under perturbation, signaling potential susceptibility to evasion tactics.

Overall, the results underscore that effective fraud detection requires not just predictive accuracy, but also robustness to adversarial manipulation. Based on our findings, MLP offers the best trade-off between performance and resilience, making it a strong candidate for real-world deployment. Future work could explore adversarial training and ensemble models to further enhance system robustness.

8. Implications and Recommendations

As fraud tactics grow more sophisticated, banks must evolve beyond traditional rule-based detection and adopt machine learning systems that can detect subtle, rare, and evolving threats. This project addresses two major shortcomings in current industry practices: the inability of models to effectively detect rare fraud cases and their brittleness when faced with small, targeted input changes designed to evade detection.

By benchmarking a variety of machine learning models against precision, recall, and F1-score, we identified MLP and random forest as leading candidates for high-stakes fraud detection. Unlike simple accuracy-based metrics, our evaluation emphasized the importance of detecting minority-class events—transactions that are fraudulent but vastly outnumbered. These models, particularly MLP, also showed strong resilience when tested against realistic manipulations in key features like transaction amount, time, and geographic distance.

Crucially, the process of applying manual feature perturbation revealed weaknesses that would not be uncovered through standard validation. This type of evaluation acts as a "stress test" for fraud detection systems, uncovering how small changes in inputs, similar to real-world fraud evasion strategies, can impact model performance. By surfacing these vulnerabilities early, banks can identify where existing detection logic is likely to fail and reinforce those weak points before deployment.

To further enhance fraud detection efforts, we recommend that banks:

1) Adopt machine learning models that prioritize rare-event detection and maintain recall in imbalanced datasets.
2) Use robustness testing methods, such as targeted feature perturbation, to simulate and preempt real-world adversarial behavior.
3) Incorporate ongoing model diagnostics, retraining systems on recent fraud activity to remain adaptive and up-to-date.
4) Consider model ensembles or hybrid strategies, balancing interpretability with predictive power for regulatory compliance and operational reliability.

By integrating these strategies, banks can build fraud detection systems that are not only more accurate but also more resilient, better equipped to protect against the evolving nature of financial crime.

Reference

[1] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, “Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components,” in Proc. 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 2018, pp. 1555–1562. doi: 10.1109/IVS.2018.8500421.

[2] G. Apruzzese et al., “‘Real Attackers Don’t Compute Gradients’: Bridging the Gap Between Adversarial ML Research and Practice,” in Proc. 2023 IEEE Conf. Secure and Trustworthy Machine Learning (SaTML), Raleigh, NC, USA, 2023, pp. 339–364. doi: 10.1109/SaTML54575.2023.00031.

[3] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection,” Information Sciences, vol. 479, pp. 448–455, 2019. doi: 10.1016/j.ins.2018.02.060.

[4] M.-Y. Tsai, H.-H. Cho, C.-M. Yu, Y.-C. Chang, and H.-C. Chao, “Effective Adversarial Examples Identification of Credit Card Transactions,” IEEE Intelligent Systems, vol. 39, no. 4, pp. 50–59, Jul.–Aug. 2024. doi: 10.1109/MIS.2024.3378923.