# Evaluating Machine Learning Approaches on Credit Card Fraud Detection

Aahad Abubaker
aabubak2@depaul.edu
DePaul University
Chicago, Illinois, USA

## Abstract

In this study, we address the issue of credit card fraud detection using machine learning models on a simulated dataset. Due to confidentiality concerns, real transaction data is often inaccessible; thus, we utilized a simulated dataset to evaluate the effectiveness of various algorithms. Our methodology includes data preprocessing, feature selection, and model evaluation, focusing on SVM, Random Forest, XGBoost, and Logistic Regression. One of the most significant challenges in these problems is the class imbalance; therefore, we have tried various sampling methods to avoid overfitting and ensure sufficient information is captured. The results highlight both the challenges and effectiveness of each approach, providing insights into the applicability of machine learning in fraud detection.

***Keywords:*** Credit Card Fraud Detection, machine learning, class imbalance, simulated dataset,

## 1 Introduction

In the past decade, credit card transactions have become more common, offering unparalleled convenience and efficiency compared to transfers of paper notes. However, this advancement has been followed by a significant surge in fraudulent activities, posing severe financial risks to both consumers and financial institutions. Credit card fraud leads to substantial annual financial losses globally, specifically, the U.S. losses from credit card fraud totaling up to $165.1 billion in 10 years [4]. This increasing trend not only results in direct financial losses but also undermines consumer trust in digital payment systems.

By using large datasets, machine learning algorithms can identify and learn from patterns indicative of fraudulent behavior, providing a significant advantage over conventional methods. However, the accessibility of real-world transaction data for research purposes is often restricted due to privacy concerns of the consumers.

This study aims to bridge this gap by utilizing a simulated dataset to explore the efficacy of various machine learning models in detecting fraudulent transactions. As with a simulated dataset, we are able to interpret the features and reasons why a transaction was detected as fraudulent, as opposed to blindly believing a model using an anonymized dataset. By examining different algorithms, including Support Vector Machines (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), and Logistic Regression, this project seeks to understand their strengths and limitations within the context of fraud detection. Our approach encompasses data preprocessing, feature selection, and model evaluation as well as addresses the challenge of class imbalance, prevalent in fraud detection datasets.

## 2 Literature Review and Related Works

In similar study conducted by Awoyemi et al. tackle credit card fraud detection through a comparative analysis of three machine learning techniques: Naive Bayes, K-Nearest Neighbor (KNN), and Logistic Regression. Utilizing a dataset of credit card transactions from European cardholders, which is notably skewed with only 0.172% fraudulent transactions, they address the challenge of imbalance through hybrid sampling techniques to produce datasets with two different distributions: 10:90 and 34:66, achieved through stepwise addition and subtraction of data points [1]. There was no discussion on the decision for choosing the splits of 10:90 and 34:66, surprisingly as they are interesting splits when discussing data imbalance[1]. A significant aspect of their results involves the presence of Principal Component Analysis (PCA) as there isn't much discussion on the learning the reasons why some transactions were predicted as fraud and some were not. Although PCA may be great in reducing complexity of the data, it may hurt its explainability.

Zeager et al.'s model design introduces a novel approach to credit card fraud detection by employing game theory within a machine learning framework. This approach simulates an ongoing battle between fraud detection systems and fraudsters, aiming to create a dynamic defense mechanism that evolves in response to new fraudulent strategies. The application of game theory to machine learning is interesting as it applies financial market knowledge to machine learning, pushing fraud detection systems from static models to adaptive solutions similar to that of Markov processes [6]. Emphasizing a system's ability to predict future fraudulent tactics rather than just reacting to past behaviors is important to preventing cybercrime. However, the study's primary limitation is its exclusive use of logistic regression. It would have been beneficial to see other model performances using the game theory design to see how they would preform. [6]

In the paper "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy" by Andrea Dal

Pozzolo et al., the authors address the persistent challenges of credit card fraud detection such as concept drift (changes in consumer behavior and fraudster tactics over time) and verification latency (delays in confirming transactions as fraudulent or legitimate)[3]. The authors describe the process employed in real world credit card fraud detection systems, known as FDS, where classifiers are used to analyze all authorized transactions and flag the most suspicious ones for further review. Professional investigators then examine these alerts and contact cardholders to verify whether each transaction is genuine or fraudulent. This process results in feedback in the form of labeled transactions, which can be used to train or update the classifier to maintain or enhance its performance over time. Interestingly, they note that " Most papers in the literature ignore the verification latency" and "the alert–feedback" [3]. It can be that the alert-feedback system is much more computationally expensive or too complex too implement in real world FDS where practicality is the most important thing. interaction

François de la Bourdonnaye and Fabrice Daniel explore the impact of various categorical encoding techniques on the performance of gradient boosting models in a credit card fraud detection context. The paper is particularly important for this project as it includes a detailed examination of several encoding strategies, including traditional methods and more innovative approaches like weight of evidence. It also uses various gradient boosting models such as CatBoost and LGBM encoding[2]. They also choose to illustrate the results of accuracy and AUC of their models through a comparative framework that showcases the percentage variations from baseline models than exposing raw values [2]. This method seems like a good way to show variations between models without laying out numbers for the audience to compare themselves.

In the context of credit card fraud detection, Synthetic Minority Over-sampling Technique (SMOTE) stands out as an influential method for addressing the challenges associated with imbalanced data sets, a common scenario in fraud detection tasks. The fundamental idea behind SMOTE is to artificially create new minority class samples by interpolating between existing ones, thereby balancing the data set without simply duplicating minority class instances [5]. Because of the artificial nature, SMOTE should be a better solution to class imbalance than oversampling as that can cause overfitting. Fernandez et al. note that using SMOTE can actually improve the detection rate of minority classes too much where it also increases the false positive rate [5]. However, in the domain of FDS, having false positive is better than the model having false negatives, causing consumers and the credit card company to lose money.

# 3 Methodology


## 3.1 Dataset Overview

The study utilizes a simulated dataset designed to mimic real-world credit card transactions while adhering to privacy constraints. It was uploaded on Kaggle by Kartik Shenoy using a simulator created by Brandon Harris. This dataset comprises various features typically found in transaction data, such as purchase amount, time of transaction, and merchant details, along with labeled outcomes indicating fraudulent or legitimate transactions. There are about 1.3m samples in the dataset and because of the simulated nature, there are no missing or null values.

## 3.2 Data Preprocessing and Categorical Encoding

I opted to keep outliers such as high 'AMT' values as it would make sense for a high 'AMT' transaction to be associated with fraud. A significant time spent in preprocessing was feature engineering, where I created new variables 'hour' and 'day', and 'month' from the 'trans_date_trans_time' field. This transformation was important for capturing temporal patterns that could be indicative of fraudulent activity, as fraud occurrences can vary significantly by time of day and day of the week, or even month of the year.

In terms of handling categorical variables, different encoding techniques were employed based on the nature of each categorical variable and its relationship with the target variable. For the 'gender' attribute, one-hot encoding was applied to transform this binary feature into a label of 1s and 0s. Label encoding was utilized for the 'category' feature, to label categories such as 'personal_care' into numbers. For the remaining categorical variables, including 'merchant', 'job', 'last', and others, Weight of Evidence (WoE) encoding was adopted[2]. This technique, often cited in credit scoring and risk modeling, transforms categorical variables into a continuous measure representing the likelihood of an event, making it particularly suited for fraud detection scenarios [2]. Another label encoder that has been used in domain of FDS is target encoding and the CatBoosting encoding, an encoding method built into the CatBoost Gradient Boost algortihm [2].

## 3.3 Exploratory Data Analysis

The dataset consists of approximately 1.3 million samples, out of which only about 7,000 were identified as fraudulent transactions Almost less than 0.54% of transactions in the dataset were classified as fraud as shown in Figure 2.

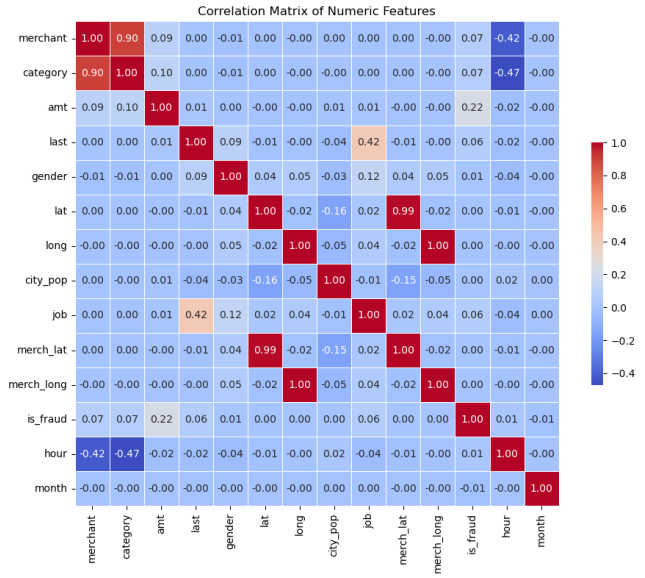


**Figure 1.** Correlation Matrix after encoding categorical features.

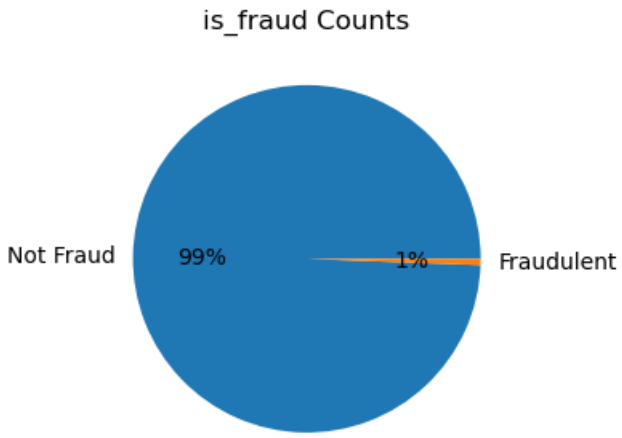


**Figure 2.** Class Imbalance between Fraud and Not Fraud.

By engineering features from the 'trans_date_trans _time' variable into 'hour' and 'month', we observed distinct patterns in fraud occurrences. A density plot of fraud transactions by time revealed that fraudulent activities tend to peak during midnight hours as seen in Figure 3's left skewed histogram, suggesting a temporal vulnerability or a preferred time for fraudulent actions by perpetrators. Similarly, the early months of the year showed a higher density of fraud, indicating a specific period of higher fraud.

However, when examining the distribution of fraud transactions by gender, it was found that both genders experienced approximately the same amount of fraud incidents

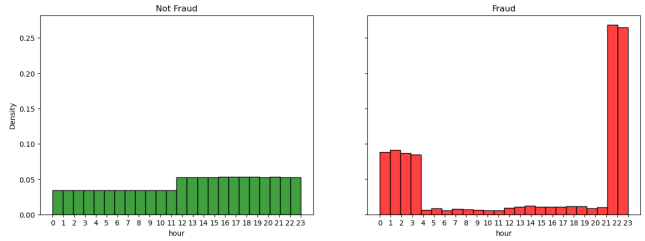


**Figure 3.** Density Plot of Hour vs Fraud Transaction Count

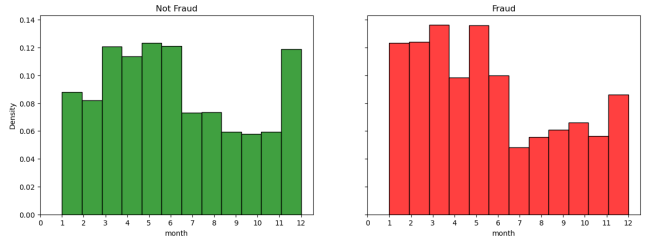


**Figure 4.** Density Plot of Month vs Fraud Transaction Count

(around 4,000 each) although there are about 10,000 more female transactions as seen in Figure 5. This parity in fraud incidence, despite the unequal gender distribution, provides an interesting insight into the gender-neutral nature of fraud risk within this dataset.

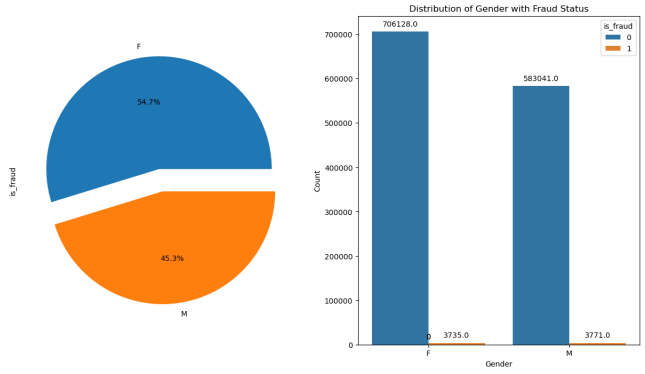


**Figure 5.** Gender Distribution and Fraud Transaction Correlation Visualization

Exploring the 'category' feature as seen in Figure 6 showed that certain categories were more susceptible to fraud. We can use the insights from exploring the category to aid in feature selection. For instance, if categories such as 'online purchases' or 'travel' exhibit higher fraud rates, these can be areas of focus for fraud prevention measures. In our dataset, we saw an increased fraud count in online shopping and grocery shopping, one of which seems logical and the other less so. One reason that I think why grocery shopping had a higher fraud rate than other categories is that fraudulent purchases would be easier to slip through the cracks. It's not

like a stolen credit card was being used to buy expensive electronics, but just groceries, so it may not be a red flag. Though I realize this is just speculation.

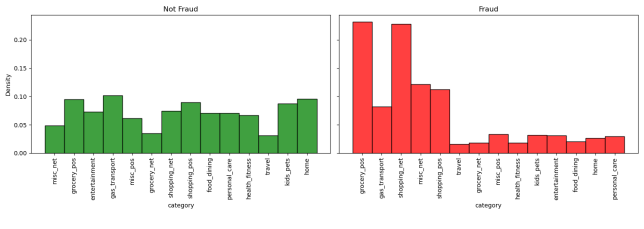


**Figure 6.** Density Plot of Transaction Category vs Transaction Count.

### 3.4 Model Selection

Four machine learning models were chosen for this project: Support Vector Machines (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), and Logistic Regression. These models were selected based on their prevalent use and proven efficacy in classification tasks, particularly in scenarios involving complex, high-dimensional data like credit card transactions. Each model brings unique strengths: SVM for its effectiveness in high-dimensional spaces, Random Forest for its ensemble approach, XGBoost for its speed, and Logistic Regression for its interpretability and simplicity. Logistic Regression was used as a sort of baseline for myself, as I was most experienced in classification problems using LR, from previous courses. Though, it is important to consider the pros and cons of each approach, whether it was a longer validation runtime vs. a higher performing model and vice versa.

### 3.5 Dealing with Class Imbalance

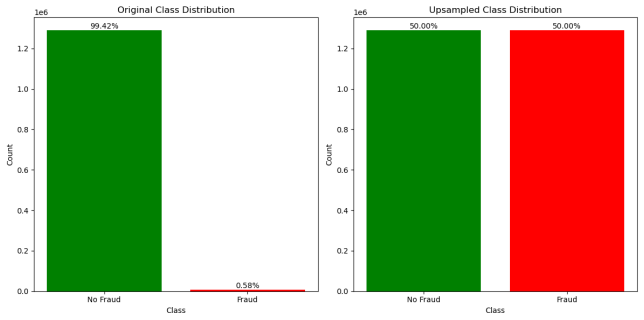


**Figure 7.** Oversampling Minority Class.

The initial approach involved applying traditional oversampling techniques to address the significant class imbalance inherent in fraud detection datasets. However, this led to the creation of an exceedingly large dataset with approximately 2.6 million rows as seen in Figure 7, which significantly hampered the computational efficiency and feasibility of model training. The vast increase in data size resulted in extended training times and, in some instances, prevented the models from running to completion.

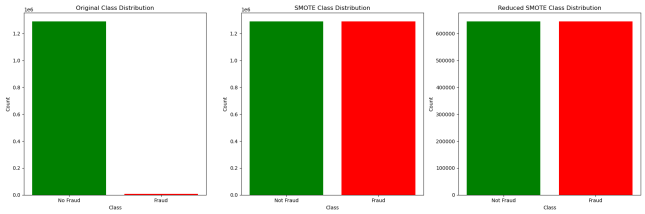


**Figure 8.** SMOTE Oversampling and then Reduction

For another approach, SMOTE, Synthetic Minority Oversampling Technique, was used to create synthetic samples to balance the classes and preventing model bias toward the majority class. To mitigate the issues of the dataset being too large, the dataset generated by SMOTE was reduced by half[5]. While this approach successfully alleviated computational constraints, it introduced concerns regarding the information loss in the reduced dataset as seen in Figure 8. The modification potentially undermines the purpose of SMOTE, which aims to create a balanced representation of minority and majority classes.

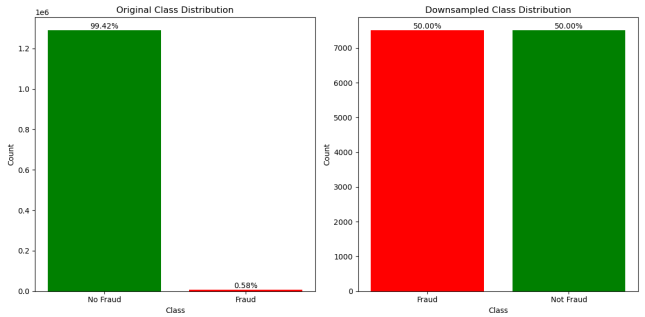


**Figure 9.** Downsampled Minority Class.

In an alternative attempt to address the imbalance while still having a large enough dataset to create a model from, undersampling techniques were applied, significantly reducing the dataset size to approximately 14,000 samples from the original 1.3 million as shown in Figure 9. While this approach rectified computational issues, it introduced a different set of problems.

The drastic reduction in dataset size led to a loss of valuable information from the initial data. The substantially smaller dataset is what I believe may have caused the overfitting in many of the models[5]. Unfortunately, each ML model was only able to validate and run after undersampling the data.

**Table 1.** Performance Metrics for Machine Learning Models with Feature Selection after Undersampling

| Model | Accuracy | AUC | Runtime |
|---|---|---|---|
| SVM | 0.86 ± 0.01 | 0.87 ± 0.01 | 63.80 |
| Random Forest | 0.97 ± 0.00 | 1.00 ± 0.00 | 17.07 |
| XGBoost | 0.96 ± 0.01 | 0.99 ± 0.00 | 0.69 |
| Logistic Regression | 0.86 ± 0.01 | 0.91 ± 0.01 | 0.35 |

### 3.6 Model Training and Evaluation

Each model was trained using the training subset of the dataset, with parameter tuning processed with grid search to find the optimal parameters. This process involved varying hyperparameters such as the kernel type for SVM, the number of estimators and maximum depth for Random Forest and XGBoost, and the regularization strength for Logistic Regression. I initially opted to use the default hyperparameters were listed in the Scikit-learn documentation and examples but decided that a finalized model should be at its most optimized. The best parameters were determined based on cross-validation scores and runtime, specifically focusing on maximizing the Area Under the Curve (AUC) and Accuracy scores (ACC) validate the performance of the model. Finally, to validate the performance of each model, a k-fold cross-validation approach was used, with k set to 5.

## 4 Results

The validation of the machine learning models regarding their performance in fraud detection, were measured in terms of accuracy (ACC), Area Under the Curve (AUC), and computational runtime in seconds. The models were also created and evaluated after undersampling the minority class to about 14,000 samples and half of each class and are shown in Table 1 above.

The SVM model showed lower accuracy scores in detecting fraudulent transactions, with an ACC approximately around 0.85 and an AUC score near 0.90. These metrics, while respectable, were not the highest observed among the tested models. More notably, SVM exhibited a significantly long runtime of 64 seconds, which is substantial compared to the other models. This extended processing time, coupled with the lower ACC scores, suggests that while SVM can be effective in fraud detection, its computational inefficiency might be a considerable drawback in real-time FDS where speed is crucial.

XGBoost stood out with its high performance, achieving excellent scores in both ACC and AUC metrics. Its accuracy surpassed other models, coupled with an high AUC, indicating strong predictive power in distinguishing between fraudulent and legitimate transactions. Moreover, XGBoost demonstrated remarkable speed with a runtime of only 0.71

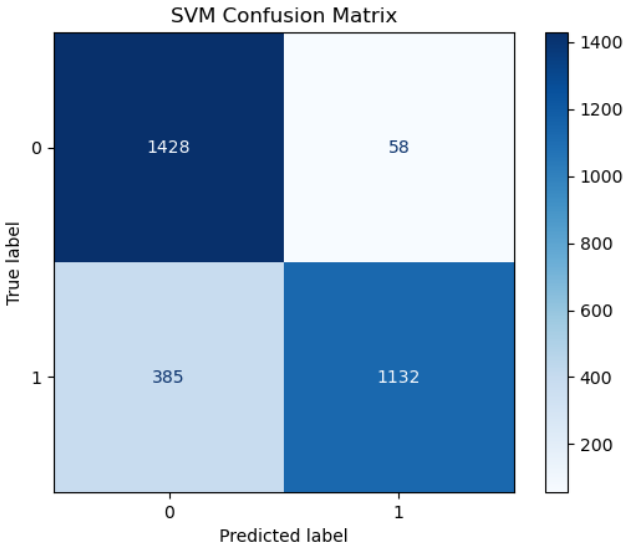


**Figure 10.** SVM Confusion Matrix

seconds, making it an exceptionally efficient choice for fraud detection in terms of both speed and accuracy.

Random Forest showed good performance metrics, with its ACC and AUC scores being very good and indicative of reliable fraud detection capabilities. However, its runtime of 20 seconds, while faster than SVM, still presents a challenge for real-time detection needs. Despite its solid performance as seen in Figure 11, the relatively long processing time may limit its practicality in scenarios requiring immediate fraud detection responses.

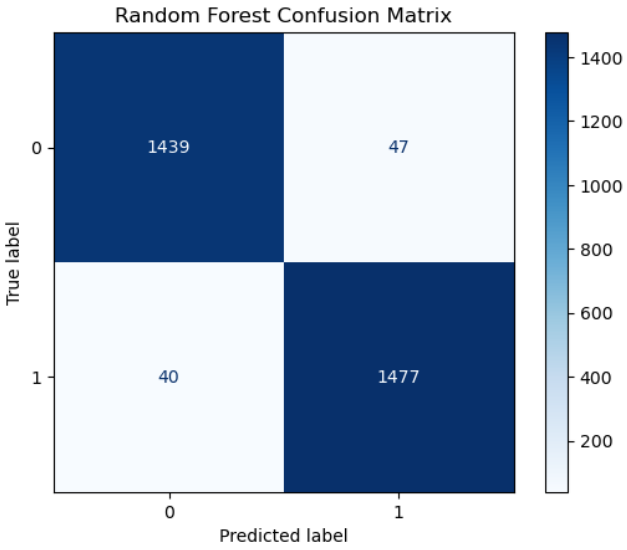


**Figure 11.** Random Forest Confusion Matrix

Logistic Regression offered a fast alternative, with a minimal runtime of 0.26 seconds, making it viable for the need for

quick decision-making in fraud detection systems. However, its performance, in terms of ACC and AUC, mirrored that of SVM, standing at approximately 0.85 and 0.90, respectively. While the speed is a significant advantage, the similarity in performance scores to SVM indicates that, while efficient, it may not provide the best solution for maximizing detection accuracy.

## 5 Conclusions

Through the application of various algorithms: SVM, Random Forest, XGBoost, and Logistic Regression on a simulated dataset, we have learned valuable insights into the strengths and limitations of each model in the context of an imbalanced classification problem of credit card fraud detection.

While techniques like SMOTE and undersampling offer pathways to address class imbalance, they also present significant trade-offs in terms of computational efficiency and capturing the data[5]. Notably, the challenges encountered with oversampling leading to an unwieldy dataset size and undersampling resulting in a loss of critical information highlight the nuanced balance required in dataset preparation for fraud detection.

## 6 Future Work

There are several avenues for future research that could further enhance the effectiveness of fraud detection systems and some that methods that I would have liked to apply to my work after literature review and feedback.

For example, while simulated datasets are invaluable in allowing researchers to analyze datasets without privacy issues, validating the findings on real-world datasets is perhaps more important. Future studies could apply the developed models to actual transaction data, subject to privacy constraints, to evaluate their performance in a real-world context. But again, it can be difficult to explain the features that are important to the model due to confidentiality and PCA done on the datasets.

One approach that I wanted to try out but did not have the time to was Neural Networks. The exploration of deep learning models, known for handling high-dimensional data, would be interesting to try in this dataset. Investigating autoencoders for anomaly detection or recurrent neural networks for sequential transaction analysis would also been a good addition for this FDS project.

Given the limitations observed with both oversampling and undersampling, a huge part of my project if I were to add on to it would be to explore hybrid approaches that combine the strengths of different resampling strategies. Where I would use SMOTE to a certain number of samples and then undersample the minority class after.

## References


[1] John O. Awoyemi, Adebayo O. Adetunmbi, and Samuel A. Oluwadare. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*. 1–9. https://doi.org/10.1109/ICCNI.2017.8123782

[2] François De La Bourdonnaye and Fabrice Daniel. 2021. Evaluating categorical encoding methods on a real credit card fraud detection database. *CoRR* abs/2112.12024 (2021). arXiv:2112.12024 https://arxiv.org/abs/2112.12024

[3] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. 2018. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. *IEEE Transactions on Neural Networks and Learning Systems* 29, 8 (2018), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643

[4] John Egan. 2021. Credit Card Fraud Statistics. https://www.bankrate.com/finance/credit-cards/credit-card-fraud-statistics/. Accessed: 03/15/24.

[5] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research* 61 (2018), 863–905.

[6] Mary Zeager, Aksheetha Sridhar, Nathan Fogal, Stephen Adams, Donald Brown, and Peter Beling. 2017. Adversarial learning in credit card fraud detection. 112–116. https://doi.org/10.1109/SIEDS.2017.7937699