Credit Card Fraud Detection Based on Different Types of Machine Learning Models

Qing Zhao
qzhao@unbc.ca

Joshua Phang
phang@unbc.ca

Abstract

In an era dominated by technological advancements and online transactions, credit card fraud has emerged as a significant threat to individuals and businesses alike. Year after year, credit card fraud attempts are on the rise, with a staggering 46% year-over-year increase reported globally. Global losses from payment cards currently total $34 billion in 2022 with projected global losses reaching an astounding $43 billion by 2026. These figures underscore the urgent need for implementing robust fraud prevention strategies and leveraging advanced technologies to protect consumers and businesses from financial harm. In this article, we explore the efficacy of various machine learning techniques in detecting financial fraud transactions through comparative experiments. We conducted three sets of comparative experiments. The initial set of experiments delved into the impact of different sampling techniques on model training. Subsequently, we scrutinized the effectiveness and stability of different single models in the second set of experiments. Finally, the third set of experiments focused on investigating whether a bagging ensemble model could augment the efficacy of single models.

Keywords — Support Vector Machine, Logistic Regression, Neutral Network, Decision Tree, Bagging, SelectKBest, SMOTE, RandomUnderSampler, Shap, Fraud detection

1 Introduction

The idea of credit is an agreement between a customer and a bank that allows the customer to purchase items using money the bank provides, relying on trust that the customer will pay it back at a later date. Credit cards, one of the most common forms of payment, is based on this credit system. While credit cards are very safe to use, stolen information can happen to anyone. Especially with the growth of online shopping, the frequency of credit card fraud incidents has surged notably. According to reports done by the Federal Trade Commission in 2023, a total of 114,348 credit card fraud reports were made, resulting in approximately 246 million dollars lost [2]. The global rise in credit card fraud has impacted both the security of bank funds and the financial stability of countless innocent cardholders as a whole. Consequently, there is an increasing demand for precise detection algorithms to prevent fraudulent behaviour.

In addition to the studies listed below in our literature review, this paper hopes to provide additional insight into applying various machine learning techniques to this largely imbalanced dataset to achieve a better precision and

recall score. For data sampling techniques, this study employs SMOTE and Random Under Sampling to compare the results between over-sampling and under-sampling. To see which model works best for this real-world application, we experimented with Support Vector Machines, Logistic Regression, Multilayer Perceptron, and Decision Tree classifiers. Finally, we implemented Bagging on all of the models listed above to compare the performance results on our dataset.

This paper will go over the design process of our experiments including the workflow, dataset, feature selection, and sampling techniques. The rest of the paper will cover the results of our experiments, followed by interpretations of our model's behaviours and results.

2 Related Work

The topic of credit card fraud detection has been widely researched, and many different studies demonstrate their approaches through unique angles of approach.

Y. Du investigates the importance of feature scaling and oversampling when working with an imbalanced dataset, furthermore applying the fully processed data to both a LassoCV and a Random Forest model [6].

J. Awoyemi, A. Adetunmbi, and S. Oluwadare provides a comparative analysis of how different data distributions have an impact on the accuracy, sensitivity, specificity, precision, true-positive rates, and true-negative rates on Naive Bayes, k-Nearest Neighbour, and Logistic Regression models [4].

V. N. Dornadula and S. Geetha examines the Matthews Correlation Coefficient as an alternate way of measuring a model's performance that takes into account all true and false prediction values [5].

D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla compare the performance of popular models when combined with SMOTE such as Logistic Regression, Random Forest, and Naive Bayes, and conclude that a classic algorithm like Random Forest can achieve similar results to a simple neural network like Multilayer Perceptron [11].

S. Shirgave, C. Awati, R. More, and S. Patil proposes a system that ranks each alert based on the correctness of an answered security question where models are trained and subsequently updates the data in a feedback loop with delayed samples [10].

F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed apply deep learning algorithms such as CNN and Autoencoders to achieve high precision and recall scores on credit card fraud detection [3].

S. P. Maniraj, A. Saini, S. Ahmed, and S. Sarkar apply the Local Outlier Factor and Isolation Forest Algorithm to their models for outlier detection. Their experiment showed that not using sampling techniques in a highly imbalanced dataset can result in poor precision and recall scores [8].

H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes built a custom BiLSTM-MaxPooling-BiGRU-MaxPooling model that achieved better results than standard models with a 91.37% AUC when combined with Random Oversampling [9].

Finally, E. Ileberi, Y. Sun, and Z. Wang utilizes a Genetic Algorithm-based feature selection method in combination with popular machine learning models such as Random Forest, Decision Tree, and Logistic Regression [7].

3 Experiment Design

3.1 Workflow

The workflow of this paper is as follows in Figure 1.

Figure 1: The workflow for our experiments

3.2 Dataset

In our research, we utilized a famous dataset of European cardholders in September 2013 downloaded from Kaggle [1]. The dataset covers transactions that occurred over two days, with 492 transactions being fraudulent out of a total of 284,870 transactions. It is a typical imbalanced dataset, with positive classes accounting for only 0.172%. Its imbalanced distribution represents the actual condition worth being concerned with. The dataset contains 30 input characteristic variables and 1 label column (the value of 1 representative fraudulent occurs and 0 otherwise). Apart from the "Time" and "Amount" features, the remaining 28 features (represented as V1 to V28) are derived from the original information after undergoing PCA transformation. Feature "Time" contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature "Amount" is the transaction Amount. All variables are numerical, with no missing or erroneous values. Due to confidentiality issues, original features and background information about the data are not provided.

3.3 Feature Selection

After the data exploration step, we applied the SelectKBest technique for feature selection, which selects the most relevant features based on mutual information scores between features and the target variable. By finding the optimal feature subset, feature selection helps to minimize the number of features, enhance the model's correctness, and speed up the experiment process.

rank	features	mutual information scores	rank	features	mutual information scores
1	V17	0.00804	17	V1	0.00199
2	V14	0.00798	18	V8	0.00184
3	V10	0.00735	19	V28	0.00176
4	V12	0.00735	20	Time	0.00170
5	V11	0.00661	21	Amount	0.00148
6	V16	0.00579	22	V19	0.00132
7	V4	0.00484	23	V20	0.00114
8	V3	0.00476	24	V23	0.00083
9	V18	0.00402	25	V24	0.00059
10	V9	0.00400	26	V26	0.00046
11	V7	0.00394	27	V22	0.00039
12	V2	0.00309	28	V25	0.00038
13	V21	0.00230	29	V15	0.00023
14	V27	0.00227	30	V13	0.00021
15	V6	0.00227			
16	V5	0.00226			

Table 1: Rank and score for each feature

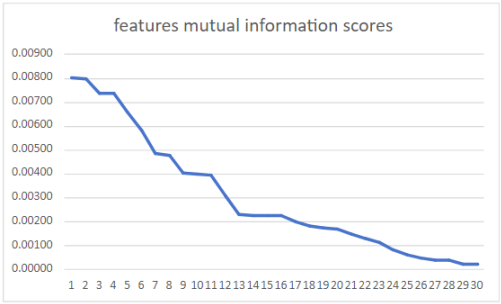

Figure 2: The curve of features' mutual information scores

To decide the number of features to include in the training experiments, we looked for a large decrease in scores between two consecutive features. In the case of the features presented in Table 1, although there are significant differences in the scores, the contribution of information remains high for top-ranked features, we select features with larger differences from those ranked lower. From Figure 2, it is evident that the curve gradually flattens after

the 16th (V5) feature. Therefore, we chose the top 16 features for our following experiments.

3.4 SMOTE and Random Under Sampling

The low proportion of fraudulent transactions is a true reflection of normal financial transactions, and the better the fraud detection financial institutions perform, the lower the proportion of fraudulent samples that they can retain. However, this poses a challenge for these institutions in building more effective transaction fraud detection models. Addressing the issue of imbalanced samples is a key objective of our experiment to ensure the recognition and stability of the model.

After dividing the dataset into training (80%) and testing (20%) datasets, we applied SMOTE (Synthetic Minority Over-sampling Technique) and Random Under Sampling techniques separately to the training set, resulting in two sets of training samples with new distributions. These, along with the original training dataset, formed three groups of training samples with different distributions.

SMOTE (Synthetic Minority Over-sampling Technique) works by generating synthetic samples of the minority class (in this case, fraudulent transactions) to balance the class distribution. We did not opt for complete balancing of the minority and majority class sample quantities, as this would essentially double the sample size. Given the vast transaction volumes in financial institutions, this is not a feasible solution in real-world business modelling practices due to the significant computational overhead. Instead, we chose a minority class sample proportion of 0.05, which is a commonly used ratio in practice.

Similarly, for Random Under sampling (which works by randomly removing instances from the majority class (non-fraudulent transactions) to balance the class distribution) operations, we also chose to use the same proportion.

Training dataset	Original	SMOTE	RandomUnderResample
[negitive postive]	[227451 394]	[227451 11372]	[7880 394]
odds	[1 0.0017]	[1 0.05]	[1 0.05]

Table 2: Three groups of training dataset

Subsequently, all our experimental procedures were separately applied to these three groups of training samples. we used the same testing samples to test and evaluate the performance of different models, drawing conclusions from these experiments.

4 Experiment Results

In this paper, we delineate three comparative experiments. The first experiment centres on comparing the impacts of various resampling techniques, tested across different machine learning models. The second experiment delves into assessing the effectiveness and reliability of diverse machine learning models across varied distributions of training datasets. Lastly, the third experiment compares the efficacy of bagging models against their respective base models.

For the purpose of fraud transaction detection, the aim of financial institutions is to identify fraudulent activities to minimize economic losses while ensuring minimal disruption to legitimate transactions by reducing false positives. Achieving high recall is crucial for detecting as many fraudulent transactions as possible (minimizing false negatives), while high precision is necessary to minimize false alarms (false positives). Therefore, we chose the balanced F1 score as our model evaluation metric, which effectively balances both recall and precision.

In the first and second experiments, we ap-

plied four machine learning methods: Support Vector Machine, Logistic Regression, Neutral Network, and Decision Tree on each training dataset (Original, SMOTE, RandomUnderSampling) and measured their performance by the same test dataset. Using cross-validation in fine-tuning the models, our results are presented in Table 3.

Training results	Original	SMOTE	RandomUnderResample
SVC	0.8985	0.8886	0.8621
LogisticRegression	0.6860	0.8895	0.8484
MLPClassifier	0.8534	0.9228	0.8944
DecisionTree	0.8179	0.9030	0.8571

Test results	Original	SMOTE	RandomUnderResample
SVC	0.8508	0.8416	0.8750
LogisticRegression	0.7861	0.8208	0.8200
MLPClassifier	0.8543	0.8208	0.7838
DecisionTree	0.84974	0.83254	0.82412

Table 3: Training and testing results for experiments 1 and 2

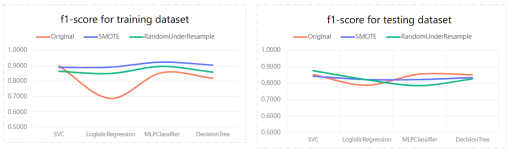

Figure 3: Line chart f1-score for training and testing dataset

From Figure 3, it is evident that during the model training phase, the performance of models trained with both sampling techniques generally outperforms those trained without sampling techniques. Furthermore, SMOTE demonstrates superior performance among the two sampling techniques. However, this performance trend did not persist when evaluating on the test set. The model trained without sampling techniques exhibited a significant improvement in performance on the test set compared to the training set. Conversely, models utilizing sampling techniques experienced varying degrees of performance degradation on the test set. Despite this, we still observed that models utilizing sampling techniques exhibit better stability, with smaller differences in performance between the test and training sets. Among them, RandomUnderSampling appears to be the most stable.

After conducting a comparative analysis of

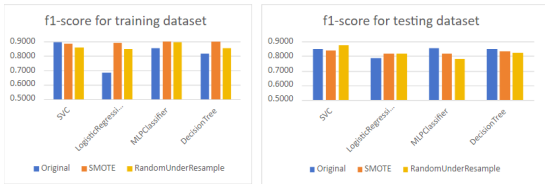

Figure 4: f1-score for training and testing datasets

the four models, according to Figure 4, the Support Vector Classifier (SVC) emerges as the top performer, surpassing others in predictive accuracy and stability. Notably, it showcases robustness regardless of whether sampling techniques are applied to the dataset. Similarly, Decision Trees exhibit insensitivity to sampling techniques. Conversely, Logistic Regression models exhibit substantial performance enhancement when applied to samples processed with sampling techniques. However, the Multilayer Perceptron (MLP) model demonstrates the poorest stability when trained on sampled data, with its performance on the test set deteriorating by more than 10 percentage points.

Building upon this foundation, our experiment delves further to explore whether the use of bagging ensemble learning methods can enhance model performance. Similarly, we conducted experiments on three different sets of samples.

Original		
Test results	single	bagging
SVC	0.8508	0.77778
LogisticRegression	0.7861	0.74699
MLPClassifier	0.8543	0.86022
DecisionTree	0.84974	0.89474

SMOTE		
Test results	single	bagging
SVC	0.8416	0.84577
LogisticRegression	0.8208	0.82075
MLPClassifier	0.8208	0.75
DecisionTree	0.83254	0.79452

RandomUnderResample		
Test results	single	bagging
SVC	0.8750	0.85714
LogisticRegression	0.8200	0.81517
MLPClassifier	0.7838	0.73029
DecisionTree	0.82412	0.75652

Table 4: Testing results for experiment 3

In our experiment, despite achieving favorable results during the training process with the bagging models, the performance on the test set, as shown in Table 4, did not exhibit further improvement. On the contrary, the ensemble model bagging led to a decrease in performance.

5 Model Interpretation

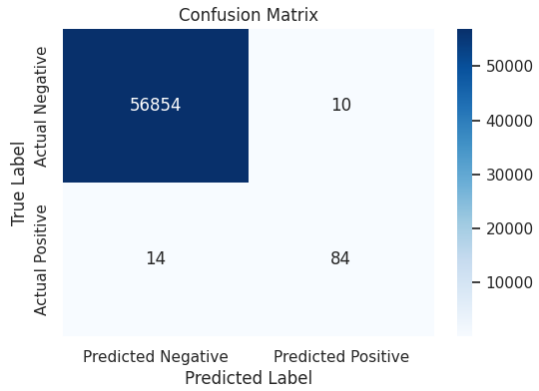

Figure 5: Confusion matrix

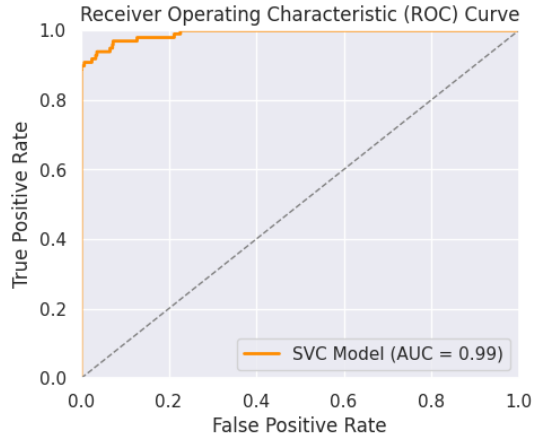

Figure 6: ROC curves

After conducting three sets of experiments, we selected the SVC model, which demonstrated the best performance, for showcasing the results. Through the confusion matrix, ROC curve, and SHAP (SHapley Additive exPlanations) plot, it is evident that the model effectively identifies fraudulent transactions. The recall rate, precision rate, and AUC are as follows: 85.71%, 89.36% and 99% respectively.

From the SHAP plot in Figure 7, it is apparent that features V14 and V17 contribute the most to predictions, followed by V12, V10, and V16, making them the top five most important features.

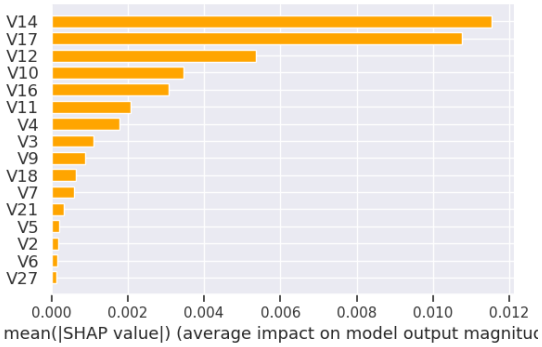

Figure 7: SHAP plot on the most important features

6 Conclusion

Fraudulent transaction detection stands as a paramount concern in financial risk management. Its low-frequency, high-impact nature poses challenges in accumulating fraudulent samples, exacerbating the issue of imbalanced datasets. In this article, we compared the effects of various sampling techniques on modeling performance and found that models utilizing the SMOTE sampling technique exhibit superior generalization capabilities. We contrasted the effectiveness of different models and their sensitivity to sampling techniques, and discovered that the SVC model yields the best predictive performance. Importantly, SVC demonstrates insensitivity to different sampling techniques, maintaining consistent performance. Lastly, we compared the performance of ensemble models against base models and found that ensemble models effectively improve the performance of base models.

References

[1] Kaggle credit card fraud detection dataset. Dataset retrieved from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data.

[2] In *Consumer Sentinel Network Annual Data Book 2023* (2023), Federal Trade Commission.

[3] Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., and Ahmed, M. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access 10* (2022), 39700–39715.

[4] Awoyemi, J. O., Adetunmbi, A. O., and Oluwadare, S. A. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI), pp. 1–9.

[5] Dornadula, V. N., and Geetha, S. Credit card fraud detection using machine learning algorithms. 2019 International Conference on Recent Trends in Advanced Computing (ICRTAC), pp. 631–641.

[6] Du, Y. Creating a credit card anti-fraud prediction model using tensorflow and machine learning. 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), pp. 334–338.

[7] Ileberi, E., Sun, Y., and Wang, Z. A machine learning based credit card fraud detection using the ga algorithm for feature selection. *Journal of Big Data* (2022), 24.

[8] Maniraj, S. P., Saini, A., Ahmed, S., and Sarkar, S. Credit card fraud detection using machine learning and data science. International Journal of Engineering Research & Technology (IJERT), pp. 110–115.

[9] Najadat, H., Altiti, O., Aqouleh, A. A., and Younes, M. Credit card fraud detection based on machine and deep learning. 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 204–208.

[10] Shirgave, S., Awati, C., More, R., and Patil, S. A review on credit card fraud detection using machine learning. International Journal of Scientific & Technology Research (IJSTR), pp. 1217–1220.

[11] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. Credit card fraud detection - machine learning methods. 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), pp. 1–5.