Undergraduate Conference on Intelligent Computing and Systems (UCICS 2025)
26-27 February, 2025; Varendra University, Rajshahi, Bangladesh

Exploring Machine Learning Techniques and Imbalanced Classification for Credit Card Fraud Detection

Mobassir Ahmmed
Computer Science and Engineering
Varendra University
Rajshahi, Bangladesh
mobassirahmmeds@gmail.com

Md. Munem Shahriar
Computer Science and Engineering
Varendra University
Rajshahi, Bangladesh
munemshahriar223@gmail.com

Mst. Sirazum Munira Mim
Computer Science and Engineering
Varendra University
Rajshahi, Bangladesh
sirazummuniramim@gmail.com

Md. Muktar Hossain
Computer Science and Engineering
Varendra University
Rajshahi, Bangladesh
mmuktar997@gmail.com

Tanver Ahmed
Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University
Dinajpur, Bangladesh
tanverahmed.cse@gmail.com

A.S.M Delwar Hossain
Computer Science and Engineering
Varendra University
Rajshahi, Bangladesh
delwar.hossain.vu@gmail.com

***Abstract*—Credit card fraud is an alarming criminal offence that causes significant harm to both individual identities and financial institutions. For this reason, it is crucial for financial institutions to identify and stop fraudulent activity. However, fraud prevention and detection are often costly, labor-intensive, and time-consuming procedures. This exploration provides an extensive experimental study of the methods that handle the imbalanced classification problem faced by fraud detection. Using a labeled credit card fraud dataset, standard machine learning techniques for fraud detection were evaluated, their weaknesses were identified, and the results were carried out. The experiments analyze how well the Support Vector Machine (SVM), Gaussian Naïve Bayes (GNB), Decision Trees (DT), Adaptive Boosting Regression (ABR), and Logistic Regression (LR) perform on highly skewed credit card fraud data. The skewed data goes through an oversampling technique. The results show that the SVM, ABR, LR, GNB, and DT classifiers have Overall Accuracy (OA) of 0.9995, 0.9992, 0.9995, 0.9789, and 0.9993, respectively. Comparative analysis shows that Logistic Regression performs better than the other methods based on OA, precision, recall, F1-score, and kappa score.**

***Index Terms*—Imbalanced Classification, Oversampling, Support Vector Machine, Gaussian Naive Bayes, Logistic Regression, Decision Tree, AdaBoost Regression, Accuracy, Precision, Recall, F1-Score, Kappa Score.**

I. Introduction

Financial fraud, which involves illegal deception for financial benefit, is a growing threat that impacts governments, businesses, and the financial industry. Credit card fraud has grown significantly as a result of the rise in credit card transactions brought on by increasing reliance on Internet technology. Fraud can be classified as either internal when banks and cardholders conspire using fraudulent identities or external, where stolen cards are used for illicit activities [1].

Since 90% of credit card fraud cases involve external fraud, detecting this type of fraud has been the focus of significant research. Fraudulent transaction detection using traditional manual approaches is ineffective and time-consuming. In order to address credit card fraud, financial institutions are turning to computational methods. Data mining techniques are a notable approach to addressing the problem of credit card fraud detection. Classifying transactions into valid and fraudulent categories is the first step in detecting credit card fraud [2]. The core principle behind this detection relies on examining how a card is used for transactions. Support Vector Machines (SVM), Decision Trees, Naïve Bayes, XGBoost, AdaBoostRegressor, Random Forest, and Logistic Regression are some of the methods that have been used to detect credit card fraud [3].

Recent research has evaluated advanced data mining techniques such as support vector machines and random forests to enhance fraud detection [4]. The dynamic nature of fraud, imbalanced transaction datasets, selecting the best features, and selecting effective performance measures for skewed data are some of the challenges that remain. The effect of variable selection, sampling methods, and detection approaches on credit card fraud detection is investigated in this work. It assesses how well SVM, DT, GNB, ABR, and LR classifiers perform when applied to randomly oversampled, highly skewed fraud data.

The purpose of this experiment is to compare how well these algorithms detect credit card fraud on extremely unbalanced data. Accuracy, precision, recall, F1-score, and kappa score provide the basis for the comparison. In this work, the way in which credit card fraud data are handled in [5] is expanded.

II. Related Work

Some of the millions of credit card transactions that occur every day are fraudulent. By framing it as a data mining classification problem, fraud detection research uses machine learning to identify fraudulent transactions effectively and in real-time. [6],

E.A. Amusan et al [7], after employing undersampling to balance the dataset, machine learning models such as logistic regression, random forest, KNN, and decision tree were used to predict fraud. Other models were more accurate than 90%,

while Random Forest was only 95%.
Class imbalance is addressed by the proposed *FraudMiner* methodology's excellent handling of anonymized data using distinct fraud and legitimate databases. An inexpensive computing cost and effective real-time detection are ensured by regular updates using pattern recognition that adjusts to variations in behavior [6]. The paper [8] employs stratified sampling to reduce legitimate records to a manageable size. It tests fraud-to-legitimate distributions of 50:50, 10:90, and 1:99, finding that the 10:90 ratio performs best, as it closely mirrors the real-world distribution.

III. Methodologies

The dataset and the five classifiers that were evaluated in the experiments—SVM, Decision Trees, Gaussian Naïve Bayes, AdaBoost Regression, and Logistic Regression—are presented in detail in this section. Data collection, preprocessing, analysis, training, along with evaluation are all steps in the process. While PCA selects features and reduces dimensionality, preprocessing uses random oversampling to prepare and balance data. Metrics like accuracy, precision, recall, F1-score, Kappa score, and confusion matrix components (TP, TN, FP, FN) are used to evaluate the performance of classifiers that have been trained on processed data.

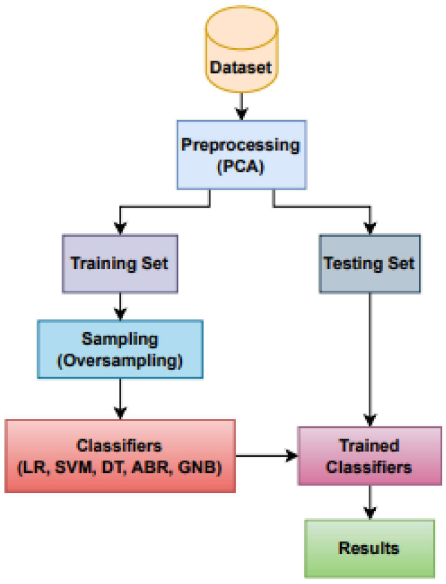

Fig. 1: Steps to detect credit card fraud.

A. Dataset Description

The dataset is collected from the ULB Machine Learning group, and its description is found in [9]. The dataset contains 284807 transactions that occurred in two days, and the dataset was created by European cardholders in September 2013. The dataset contains 0.172% positive class(Fraud case) and 99.828% false case(Normal transactions) of total transactions. The dataset is extremely imbalanced, with only numerical input variables obtained by PCA, providing 28 main components and 30 features in total. Details about the features are unavailable due to confidentiality concerns. The dataset has three features: 'time' (elapsed seconds), 'amount' (transaction value), and 'class' (a binary target of 1 for fraud and 0 for non-fraud) [10].

B. Support Vector Machine

The Support Vector Machine (SVM) seeks to identify a hyperplane $w^T x + b = 0$ that optimally divides data points into two distinct classes with the greatest margin [11]. The hyperplane is defined as:

$$w^T x + b = 0$$

where: w denotes the weight vector, b represents the bias term, and x signifies input data points.
The kernel technique, which uses a function $\phi(x)$ to transfer the input data x into a higher-dimensional space, is used for non-linear data. The goal is to minimize the following in order to maximize the margin, $\frac{2}{\|w\|}$:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \varsigma_i$$

Subject to the constraints:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \varsigma_i$$

where y_i are the class labels ($+1$ or -1). In case these points are not linearly separable, slack variables ς_i are introduced to account for any errors or miscalculations. C is a cost parameter > 0 is associated with these errors [12].

C. Adaptive Boosting

The AdaBoost algorithm is used with additional machine learning techniques in this research to improve classification results. AdaBoost combines the outputs of multiple weak learners into a weighted sum [13]:

$$G_N(x) = \sum_{t=1}^{N} g_t(x)$$

In this case, t denotes the iteration, and g_t depicts a weak learner making predictions based on input x. The weak learner predicts $h(x_n)$ for every training sample. A weak learner is chosen and weighted by β_t at each iteration t, adding to the training error L:

$$L_t = \sum_n L\left[G_{t-1}(x_n) + \beta_t h(x_n)\right]$$

Where G_{t-1} is the boosted classifier from the previous iteration, and $\beta_t h(x_n)$ is the weak learner being evaluated.

D. Logistic Regression

Based on one or more input features, logistic regression estimates the probability of a binary outcome using a functional method. It defines the optimal parameters for the sigmoid function, a nonlinear function [10]. The sigmoid function is defined in equation (1), whereas equation (2) represents the input x as a weighted sum of the feature values (z), where the coefficients w are multiplied by each corresponding feature element and added to generate a single result. This value is used to classify the target class. If the sigmoid value exceeds 0.5, the output is classed as 1, otherwise as 0.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$x = w_1 z_1 + w_2 z_2 + \cdots + w_n z_n \tag{2}$$

Where σ is the sigmoid function, and w are the best-fit coefficients for the input data vector z.

E. Gaussian Naive Bayes

Gaussian Naive Bayes applies the Naive Bayes classification algorithm to data with a normal (Gaussian) distribution. The model assumes a Gaussian distribution for each feature x_i, given a class y_k. The probability $P(x_i \mid y)$ is defined as:

$$P(x_i \mid y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

The algorithm classifies a new data point x by calculating the posterior probabilities for each class and splitting the data point to the class with the maximum posterior probability [14].

F. Decision Tree

A decision tree splits records into subsets based on attribute values, starting with the root node. The tree arises by recursively splitting nodes until no significant splits are possible or the node size is insufficient. Splitting algorithms like as ID3, C5.0, and CART make use of metrics such as information gain, gain ratio, and Gini coefficient. Pruning eliminates superfluous nodes to prevent overfitting. New records are classified by traversing the tree from root to leaf, with the class determined by the label of the leaf [15].

G. Evaluation Metrics

A wide range of metrics, including accuracy, precision, recall, F1-score, and kappa score, are used to assess the experimental model's performance. These indicators offer a comprehensive evaluation of the model's credit card fraud detection capabilities. The formulas for accuracy, precision, recall, F1-score, and kappa score are displayed in Equations 1, 2, 3, 4, and 5, respectively:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1-Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4}$$

$$k = \frac{p_o - p_e}{1-p_e} \tag{5}$$

where,

- True Positive(TP): The number of cases accurately anticipated as positive.
- False Positive(FP): The number of cases that were wrongly anticipated as positive.
- False Negative(FN): The number of cases that were wrongly anticipated as negative.
- True Negative(TN): The number of cases that were accurately anticipated as negative.
- p_o means the overall accuracy of the model
- p_e means the measure of the agreement between model predictions and actual class values as if they occurred by random.

IV. Experimental Result and Analysis

In this paper, five classifier models based on Decision Tree, AdaBoostRegressor, Support Vector Machine, Logistic Regression, GaussianNB are developed. To evaluate the models the dataset is distributed in the original ratio which is 0.172:99.828 and 43:57, where random oversampling is used for the 43:57 distribution. For small datasets in particular, overfitting can result from excessive oversampling (e.g., 50:50 or more). A small imbalance, such as 43:57 rather than 50:50, allows the model to be more robust while maintaining natural class proportions. Random oversampling is a non-heuristic strategy for balancing class distributions that involves randomly reproducing minority target instances. Accuracy, Precision, Recall, F1-score, kappa score are the performance metrics. The Accuracy, Precision, Recall, F1-score, kappa score for the test size of 20%, 30% and 50% are present in Tables I and II respectively.

A. Original Dataset

LR consistently outperforms other models across all test sizes, with the highest accuracy, precision, F1-score, and kappa. Even with limited datasets, LR tends to generalize well because it provides L1 (Lasso) and L2 (Ridge) regularization options that assist in reducing overfitting and make it robust in high-dimensional scenarios. Because LR models a linear decision boundary, it typically performs well when the data is linearly separable. For example, with a 20% test size, LR achieves an accuracy of 0.9123 in F1-score and 0.7549 in kappa, and maintains high performance with 0.9093 and 0.9209 F1-scores at 30% and 50%, respectively. DT closely follows LR, with excellent accuracy (0.9992) and competitive kappa (0.7919) at the 20% test size. SVM has excellent accuracy but issues with recall and F1-scores, implying that it may overlook fraud situations despite its overall accuracy. ABR has a balanced performance with high F1-scores (0.8775), however, LR and DT outperform. GNB performs poorly for fraud detection, with low recall, F1-scores, and kappa values.

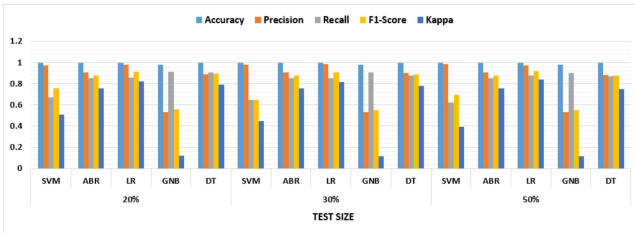

Fig. 2: Performance of Classifiers on different test size.

B. Oversampled Dataset

Classifier performance varies significantly between the original and oversampled datasets. LR maintains the maximum accuracy throughout both datasets (e.g., 0.9995 for test size 30%), but its precision slightly reduces in the oversampled dataset due to its higher sensitivity to minority classes. Oversampling improves GNB's recall (e.g., 0.9077 for test size 30%), but it still struggles

TABLE I: Performance of the Classifiers for the Original Distribution

Test Size = 20%					
	SVM	ABR	LR	GNB	DT
Accuracy	0.9988	0.9992	**0.9995**	0.9783	0.9992
Precision	0.9724	0.9078	**0.98**	0.5328	0.8866
Recall	0.6732	0.8513	0.8614	**0.915**	0.9057
F1-Score	0.7533	0.8775	**0.9123**	0.5557	0.8959
Kappa	0.5068	0.7549	**0.8246**	0.1194	0.7919
Test Size = 30%					
	SVM	ABR	LR	GNB	DT
Accuracy	**0.9995**	0.9992	**0.9995**	0.9784	0.9993
Precision	0.9772	0.9078	**0.9857**	0.5315	0.902
Recall	0.6462	0.8513	0.8537	**0.9077**	0.8774
F1-Score	0.6462	0.8775	**0.9093**	0.5534	0.8893
Kappa	0.4475	0.7549	**0.8186**	0.1148	0.7786
Test Size = 50%					
	SVM	ABR	LR	GNB	DT
Accuracy	0.9987	0.9992	**0.9995**	0.9789	0.9992
Precision	**0.9995**	0.9078	0.974	0.5314	0.8806
Recall	0.6234	0.8513	0.8786	**0.8991**	0.8681
F1-Score	0.6964	0.8775	**0.9209**	0.5532	0.8743
Kappa	0.393	0.7549	**0.8419**	0.1143	0.7485

with Precision and F1-Score. In the oversampled dataset, SVM and ABR showed significant improvements in Recall (e.g., SVM from 0.6462 to 0.9357 at test size 30%) and F1-Score, resulting in more balanced performance. Oversampling improves DT's Recall and F1-Score (e.g., rising from 0.7955 to 0.8806 at test size 30%). Overall, oversampling improves Recall and F1-Score for all classifiers, with considerable increases for SVM, ABR, and DT, although LR remains the most reliable method for high accuracy and precision, particularly in the original dataset.

TABLE II: Performance of the Classifiers for 43:57 Data Distribution

Test Size = 20%					
Classifiers	SVM	ABR	LR	GNB	DT
Accuracy	0.9925	0.9841	**0.9992**	0.9741	0.9991
Precision	0.5909	0.5457	**0.9511**	0.5283	0.8747
Recall	**0.9567**	0.9377	0.8217	0.9228	0.8562
F1-Score	0.6501	0.5791	**0.8755**	0.5467	0.8656
Kappa	0.3019	0.1635	0.7511	0.1036	**0.7305**
Test Size = 30%					
Classifiers	SVM	ABR	LR	GNB	DT
Accuracy	0.9924	0.9841	**0.9992**	0.9746	**0.9992**
Precision	0.5848	0.5457	**0.942**	0.5278	0.899
Recall	0.9351	**0.9377**	0.8129	0.9194	0.8638
F1-Score	0.6403	0.5791	0.8663	0.546	**0.8806**
Kappa	0.2824	0.1635	0.7327	0.102	**0.7612**
Test Size = 50%					
Classifiers	SVM	ABR	LR	GNB	DT
Accuracy	0.9941	0.9895	**0.9992**	0.9755	0.9991
Precision	0.6024	0.5623	**0.9387**	0.5281	0.8819
Recall	0.9292	**0.9249**	0.7962	0.9117	0.8599
F1-Score	0.6643	0.6064	0.8536	0.5467	**0.8705**
Kappa	0.3296	0.2157	0.7072	0.1029	**0.7411**

V. Conclusion

The study compares the performance of five machine learning models—Naïve Bayes, SVM, Decision Trees, AdaBoost Regression, and Logistic Regression—for binary classification of credit card fraud data. To handle class imbalance, RandomOverSampler was used, and the model's effectiveness was evaluated through metrics like precision, recall, accuracy, F1-score, and kappa score. Logistic Regression and Decision Tree consistently outperformed other methods, proving excellent accuracy for detecting fraud in both imbalanced and oversampled datasets. Oversampling significantly has increased recall, particularly for SVM. Future investigations will look at meta-learning and various sampling approaches to improve fraud detection.

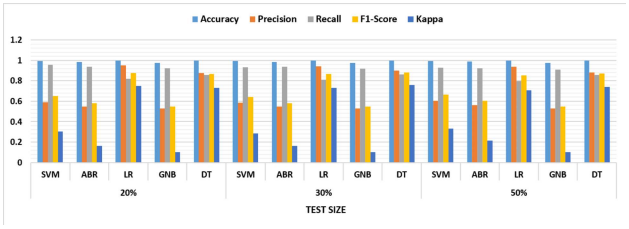

Fig. 3: Performance of Classifiers on different test size with oversampling.

References

[1] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 1, pp. 145–174, 2023.

[2] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, vol. 261, 2002, p. 270.

[3] T. Pranckevičius and V. Marcinkevičius, "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, p. 221, 2017.

[4] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision support systems*, vol. 50, no. 3, pp. 602–613, 2011.

[5] M. Fahmi, A. Hamdy, and K. Nagati, "Data mining techniques for credit card fraud detection: empirical study," *Sustainable Vital Technologies in Engineering & Informatics*, pp. 1–9, 2016.

[6] K. Seeja and M. Zareapoor, "Fraudminer: A novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, no. 1, p. 252797, 2014.

[7] E. Amusan, O. Alade, O. Fenwa, and J. Emuoyibofarhe, "Credit card fraud detection on skewed data using machine learning techniques," *Lautech Journal of Computing and Informatics*, vol. 2, no. 1, pp. 49–56, 2021.

[8] E. Duman, A. Buyukkaya, and I. Elikucuk, "A novel and successful credit card fraud detection system implemented in a turkish bank," in *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE, 2013, pp. 162–171.

[9] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE symposium series on computational intelligence*. IEEE, 2015, pp. 159–166.

[10] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 international conference on computing networking and informatics (ICCNI)*. IEEE, 2017, pp. 1–9.

[11] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[12] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[13] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using smote and adaboost," *IEEE Access*, vol. 9, pp. 165 286–165 294, 2021.

[14] D. D. Borse, S. H. Patil, and S. Dhotre, "Credit card fraud detection using naive bayes and robust scaling techniques," *International Journal*, vol. 10, no. 1, pp. 1–5, 2021.

[15] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2011, pp. 1–6.