UNIVERSIDAD NACIONAL DE SAN MARTIN

RCSI
https://revistas.unsm.edu.pe/index.php/rcsi
e-ISSN: 2709-992X

Original article
Artículo orginal
Jul-Dec, 2025

SynKGen: A kernel PCA-Based oversampling method for enhanced credit card fraud detection

SynKGen: Un método de sobremuestreo basado en kernel PCA para mejorar la detección de fraudes en tarjetas de crédito

Fray L. Becerra-Suarez[1]*, Luciani J. Jiménez-Fernández[2], Estrella D. Ticona-Tapia[3], José Rolando Cárdenas-Gonzáles[3], Pepe Humberto Bustamante-Quintana[3]

[1]Grupo de Investigación en Inteligencia Artificial (UMA-AI), Universidad Privada Norbert Wiener, Lima, Perú
[2]Escuela de Ingeniería de Sistemas e Informática, Universidad Tecnológica del Perú, Lima, Perú
[3]Grupo de Investigación en Tecnologías, Sociedad y Educación, Universidad Señor de Sipán, Chiclayo, Perú

Received: 07 Apr. 2025 | **Accepted:** 18 Jul. 2025 | **Published:** 20 Jul. 2025

Corresponding author*: fray.becerra@uwiener.edu.pe

How to cite this article: Becerra-Suarez, F. L., Jiménez-Fernández, L. J., Ticona-Tapia, E. D., Cárdenas-Gonzáles, J. R. & Bustamante-Quintana, P. H. (2025). SynKGen: A kernel PCA-Based oversampling method for enhanced credit card fraud detection. *Revista Científica de Sistemas e Informática, 5*(2), e952. https://doi.org/10.51252/rcsi.v5i2.952

ABSTRACT

Credit card fraud detection is a growing challenge in the financial domain due to data imbalance, where fraudulent transactions are minimal compared to legitimate ones. This study presents SynKGen, a data augmentation method using Kernel PCA with Gaussian perturbations to generate synthetic samples of the minority class, contrasting it with ADASYN and SMOTE. By introducing variance analysis with controlled perturbations in the minority class, the proposed approach mitigates the risks of overfitting associated with traditional interpolation-based techniques. Four classifiers, XGBoost, RandomForest, AdaBoost and VotingClassifier, were evaluated using the original data set and variants with data augmentation. The RandomForest classifier achieved the best performance when using data generated with SynKGen (accuracy: 0.9949, precision:0.9899) outperforming the results obtained with ADASYN and SMOTE. Experimental results demonstrate that SynKGen improves the effectiveness of credit card bank fraud detection. These findings highlight the importance of data augmentation strategies to optimize classifier performance in financial contexts with unbalanced data.

Keywords: ADASYN; class imbalance; ensemble learning; financial security; Kernel PCA; machine learning; SMOTE

RESUMEN

La detección de fraude con tarjeta de crédito es un desafío creciente en el ámbito financiero debido al desequilibrio de datos, donde las transacciones fraudulentas son mínimas en comparación con las legítimas. Este estudio presenta SynKGen, un método de aumentación de datos que utiliza Kernel PCA con perturbaciones gaussianas para generar muestras sintéticas de la clase minoritaria, contrastándolo con ADASYN y SMOTE. Al introducir el análisis de varianzas con perturbaciones controladas en la clase minoritaria, el enfoque propuesto mitiga los riesgos de sobreajuste asociado a las técnicas tradicionales basadas en interpolación. Se evaluaron cuatro clasificadores, XGBoost, RandomForest, AdaBoost y VotingClassifier, utilizando el conjunto de datos original y variantes con aumentación de datos. El clasificador RandomForest alcanzó el mejor desempeño al utilizar datos generados con SynKGen (exactitud: 0,9949, precisión:0,9899) superando a los resultados obtenidos con ADASYN y SMOTE. Los resultados experimentales demuestran que SynKGen mejora la efectividad de la detección de fraudes bancarios en tarjetas de crédito. Estos hallazgos destacan la importancia de estrategias de aumentación de datos para optimizar el rendimiento de los clasificadores en contextos financieros con datos desbalanceados.

Palabras clave: ADASYN; desequilibrio de clases; aprendizaje conjunto; seguridad financiera; Kernel PCA; aprendizaje automático; SMOTE

© The authors. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

1.1 Background

In an increasingly globalized world, influenced by various information technologies, the revolution in how people conduct their financial transactions is undeniable, driven by different banking applications, e-wallets, and digital payment platforms. All that is needed is a device with an internet connection to make any type of digital payment, replacing cash and traditional payment methods. This has allowed the economies of any nation to eliminate geographical barriers and facilitate international trade. However, this digitalization brings with it many security concerns, with credit card fraud being one of the greatest worries in the financial sector (Charizanos et al., 2024; Chatterjee et al., 2024; Hasan et al., 2023)

Financial losses resulting from credit card fraud cannot be underestimated (Alfaiz & Fati, 2022; Hilal et al., 2022; Sulaiman et al., 2024). A study by Juniper Research, (n.d) estimates that fraud losses will exceed $206 billion between 2021 and 2025, driven by the increase in synthetic identity fraud and account takeover. In Peru, one of the largest banks, Interbank, was a victim of a data breach by a third party, compromising the security of its customers' data. While no financial losses were reported, the breach has raised concerns among its customers and highlighted the need to implement new mechanisms to address these threats (Interbank, 2024).

Credit card fraud comes in various forms that exploit different vulnerabilities in the financial system. These include application fraud through the theft of credentials and creation of fake accounts; card cloning via magnetic stripe data capture; transactions without the physical presence of the card; complete card forgery; use of lost or stolen cards; identity theft of the cardholder; interception of new cards during postal delivery; full account takeover; injection of malicious code on websites; and merchant collusion in which they share cardholder information without authorization. Each of these fraud types represents a significant risk to financial security and requires different strategies for prevention and detection (Jain et al., 2019; Rb & Kr, 2021; Wijaya et al., 2024).

1.2 Related work

In response to these incidents, financial institutions are adopting advanced technologies such as machine learning to detect and prevent credit card fraud (Cherif et al., 2023; Dastidar et al., 2024). However, it is important to note that, according to Charizanos et al. (2024), the characteristics of fraud are dynamic and changing due to the variable behavior of cybercriminals and the demographics of the customers involved. Analytical methods are used to identify fraudulent and non-fraudulent patterns in transactions; however, they can be sensitive to class imbalance issues and result in biased predictions towards the majority class. Additionally, fraud detection systems must operate efficiently in real time to handle the large volume of global transactions per minute and ensure continuous protection of consumer accounts.

Different studies in the literature have been proposed to address the fraud detection problem. Rangannatha and Syed (2025), presented a model to classify fraud in mobile transactions by implementing a 3D Bidirectional Quasi-Recurrent Neural Networks architecture and a blockchain-based consensus algorithm. They used a Bitcoin transaction dataset and data augmentation techniques that allowed balancing the minority class against the majority class. The model improved accuracy by 12.09%, 8.91%, and 6.92% compared to implemented methods such as K-

nearest neighbor-Distributed Blockchain Consortium (KNN-DBC), Decision tree-Ethereum blockchain-enabled smart contract (DT-EBSC), and Heterogeneous Graph Transformer Networks-Ethereum smart contract (HGTN-ESC). RB and KR (2021) implemented Support Vector Machines (SVM), K-nearest neighbors (KNN), and an Artificial Neural Network (ANN) to predict credit card fraud. For this purpose, they used a transaction dataset collected between 2013 and 2014, which was preprocessed, cleaned, and normalized. Of the three methods evaluated, ANN presented the best result in the accuracy metric with a value of 99.92%.

In the study by Ileberi and Sun (2024), a hybrid deep learning model combining CNN, LSTM, Transformers and XGBoost was tested. The model was evaluated using a highly unbalanced European dataset, with only 0.172% of the records corresponding to the minority class of fraudulent transactions. The results obtained show a significant improvement in fraud detection, with a sensitivity of 96.1% and an AUC of 97.2%. In Tang and Liu (2024), they implemented an algorithm based on Structured Data Transformer (SDT) combined with federated learning. The performance of this algorithm was evaluated using two data sets, one with real data from the year 2013 and the other with simulated data from the year 2020. For the two sets, 5-fold cross-validation was applied, which allowed achieving an AUC score of 88.2% for the real dataset and 81.6% for the simulated dataset.

Li et al. (2024) used a very recent Kolmogorov-Arnold Network (KAN) approach applied in the context of fraud detection. The performance of KAN was evaluated on two data sets, obtaining 99% accuracy. Although the results are very favorable, this study has certain limitations, especially with the time used for training and validation. Another approach based on deep neural networks is the study by Adil et al. (2024), who proposed a framework called Optimized Deep Event-based Network (OptDevNet). This model achieved an accuracy of 99.98% compared to classical algorithms such as SVM or RandomForest, demonstrating its effectiveness in the context of fraud detection.

Alfaiz and Fati (2022) conducted a comparative study of 66 ML models for detecting credit card fraud. These models were evaluated in two stages using a European dataset with records from 2013. In both stages, the models were evaluated using cross-validation (K=5). In the first stage, nine models were implemented, three of which advanced to the second stage, where they were compared with 19 resampling techniques. The experimental results show that the AllKNN-CatBoost model obtained the best result, outperforming the others with an AUC of 97.94% and a Recall of 95.91%.

Coello et al. (2023), also conducted a comparative study between Logistic Regression, Decision Tree, XGBoost, Random Forest and Neural Network, which were evaluated on a data set with a very significant class imbalance between the minority and majority class. To address the data imbalance problem, random sampling, SMOTE and ADASYN techniques were implemented. The results obtained in this study highlight the combination of Random-Forest with ADASYN, achieving the best performance with an F1-Score of 98%. Similarly, Mondal et al. (2021), investigated the effectiveness of SMOTE and ADASYN data augmentation techniques with four classifiers (KNN, Logistic Regression, RandomForest and Bagging Classifier). The results showed that the SMOTE and RandomForest data augmentation technique scored the best for area under the ROC curve with 91.1%.

Zhao et al. (2024) used the LightGBM model, which was taken as the basis for the development of two new methods. The first was a class-balancing model with cost harmonization called CB-CHL-LightGBM; and the second was an oversampling model without cost harmonization OS-CHL-LightGBM. These methods improved the efficiency in detecting fraudulent transactions while addressing the problem of data imbalance, which is a very common problem in the financial domain. The results obtained on the data sets used for the F2 score ranged from 82% to 83%.

In general, these previous studies on credit card fraud detection present the following limitations:

- Data processing is considered one of the most important stages for obtaining good results with ML classifiers. In some studies (Ileberi & Sun, 2024; Ranganatha & Syed, 2025; Tang & Liu, 2024), this stage lacks de-tailed documentation on the handling of outliers, which can compromise the reproducibility and reliability of the results, as the presence of these outliers introduces significant biases in model training and distorts their predictive capability.
- Additionally, the performance of the classifiers can be significantly affected when working with highly imbalanced datasets, an aspect that often does not receive the necessary attention in the studies presented in (Adil et al., 2024; Alfaiz & Fati, 2022; Ileberi & Sun, 2024; Le et al., 2024; Tang & Liu, 2024; Rb & Kr, 2021).
- While classic oversampling techniques such as ADASYN, SMOTE, among others, are implemented to balance the minority class, these approaches may have limitations in generating synthetic samples that faithfully represent the characteristics of the original fraudulent transactions (Coello et al., 2023; Mondal et al., 2021; Zhao et al., 2024).
- As mentioned in Charizanos et al. (2024), the dynamic and evolving nature of financial fraud requires the use of up-to-date datasets to develop effective solutions. However, it is observed that many studies use historical datasets with a significant temporal gap compared to contemporary fraud patterns, which could limit the applicability of the proposed solutions (Alfaiz & Fati, 2022; Ileberi & Sun, 2024; Tang & Liu, 2024; Ranganatha & Syed, 2025; Rb & Kr, 2021).
- In most of the studies analyzed, a fundamental aspect that is often omitted or not adequately addressed is the analysis of the training and inference times of the models, which are critical factors for the effective implementation of re-al-time fraud detection systems. The lack of this information hinders the evaluation of the practical feasibility of the proposed solutions in real operational environments, where the speed of detection is as important as accuracy. In other studies, the need for computational resources is quite evident, which could limit their viability as a suitable solution (Adil et al., 2024; Ileberi & Sun, 2024; Le et al., 2024).

1.3 Contributions of the study

To overcome the above challenges, this research presents the following contributions:

- A comprehensive data preprocessing framework designed to remove outliers and irrelevant descriptors that do not add significant value to the performance of machine learning classifiers.
- Since the dataset presents a marked class imbalance, two data augmentation techniques were implemented being, ADASYN and SMOTE. In addition, a method called SynKGen is

proposed, which combines Kernel PCA with Gaussian perturbations, allowing to generate synthetic samples while preserving the highest statistical integrity of the original dataset.

- The performance of the implemented ensemble classifiers was evaluated using a large set of metrics, allowing a comprehensive evaluation. Experimental results show that the classifiers achieved their highest performance greater than 0.995 when trained and tested with the data generated by SynKGen, compared to the results obtained with ADASYN and SMOTE. These findings validate the effectiveness of the proposed method for data augmentation.

The remainder of this paper is structured as follows: Section 2 details the materials and method, including the data augmentation and classifier configurations implemented. Section 3 presents the experimental results, followed by a discussion of their implications. Finally, Section 4 describes the conclusions and future research directions.

2. MATERIALS AND METHODS

The methodology of this work is summarized in Figure 1, which broadly outlines each of the phases considered. It starts with the definition of the dataset, which undergoes data preprocessing to eliminate redundant values such as missing values, du-plicates, positive infinities, negative infinities, and descriptors that have the same value across all records. Next, the SynKGen data generation method is developed and implemented, along with other techniques such as ADASYN and SMOTE. Each of the generated synthetic datasets is divided into 80% for training and 20% for testing. Finally, the XGBoost, RandomForest, AdaBoost, and VotingClassifier models are implemented to perform the binary classification of credit card fraudulent transactions.

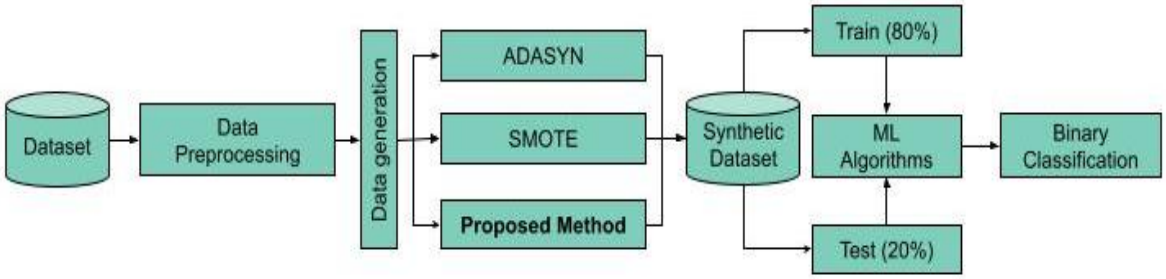

Figure 1. Proposed work methodology

2.1 Materials

The dataset used in this study is accessible on Kaggle (2024). It simulates real-world transactions, allowing researchers and analysts to study behavioral patterns and anomalies in a controlled environment. This is essential for testing fraud detection algorithms before applying them in real-life scenarios. The dataset comprises 100 000 records and seven descriptors, including a binary label that classifies fraudulent transactions with a value of 1 and normal transactions with a value of 0. It is important to note that fraudulent transactions represent only 1% of the total records, highlighting the imbalance in the data. The implementation of oversampling techniques and ML classifiers was carried out using the Python programming language version 3.9.1 and the Scikit-learn library. The computations were performed on a machine powered by an AMD Ryzen 7 3700U processor, equipped with a Radeon Vega Mobile GPU running at 2.3 GHz, supported by 24 GB of RAM, and running the 64-bit Windows 11 operating system.

2.2 Data Preprocessing

Data processing plays a crucial role in any machine learning model, as proper handling and transformation of the data is essential for obtaining more accurate and effective results (Alatawi, 2025; Becerra-Suarez et al., 2024; Lazcano & Jaramillo-Morán, 2025). For the selected dataset, a thorough analysis was conducted on duplicate values, missing values, positive and negative infinite values, as well as descriptors that had the same value across all records. As a result of this analysis, no records met the previously established conditions. Regarding the analysis of the descriptors, it was decided to remove "TransactionID," which stores the sequential values for each iteration, as it did not provide significance for the model analysis. The descriptors "TransactionType," which indicates whether the transaction was a purchase or a refund, and "Location," which stores the geographic location of the transaction, were encoded using the LabelEncoder () function from the Sklearn library, allowing the transformation of categorical variables into numeric ones. Finally, the descriptor "TransactionDate," which stores the date and time of the transaction, was removed, and new features were created from this descriptor, such as "Year," "Month," "Day," "Hour," and "DayofWeek," which allowed the decomposition of temporal information into more specific descriptors

2.3 Proposed Method for Generating Synthetic Data

The proposed method, called "SynKGen," is an oversampling technique for generating synthetic data focused on the minority class. This method not only preserves the structure of the minority class but also ensures that the generated data are representative and useful for subsequent modeling tasks. The mathematical process involving SynKGen is described below.

a) Identification and separation of classes

- The original dataset D is divided into two subsets: $D_M = \{X_i | \, y_i = 0\}$, which represents the majority class, while $D_m = \{X_i | \, y_i = 1\}$, represents the minority class.
- The descriptors of D_m are extracted by removing the label y.

b) Data Standardization

- To standardize the values of X_m, normalization is applied with a mean (μ) equal to 0 and a standard deviation (σ) equal to 1, expressed as:

$$z_i = \frac{X_i - \mu}{\sigma} \tag{1}$$

c) Dimensionality Reduction using Kernel PCA

Kernel PCA is an extension of Principal Component Analysis (PCA) that allows working with non-linearly separable data by using kernel functions to project the data into a higher-dimensional space where they may become linearly separable, which is essential for handling non-linear data, reducing dimensionality, and extracting com-plex features (Attouri et al., 2024; Kaib et al., 2025; Zhang et al., 2010). Kernel PCA accepts different types of kernels and parameters, which were defined after performing several tests to obtain the best results. In this case, a "linear" kernel was used, with a gamma value of 0.01 and 8 principal components that capture 90% of the accumulated variance of the data. This approach simplifies the data while preserving as much relevant information as possible, thus optimizing the model's performance as illustrated in Figure 2.

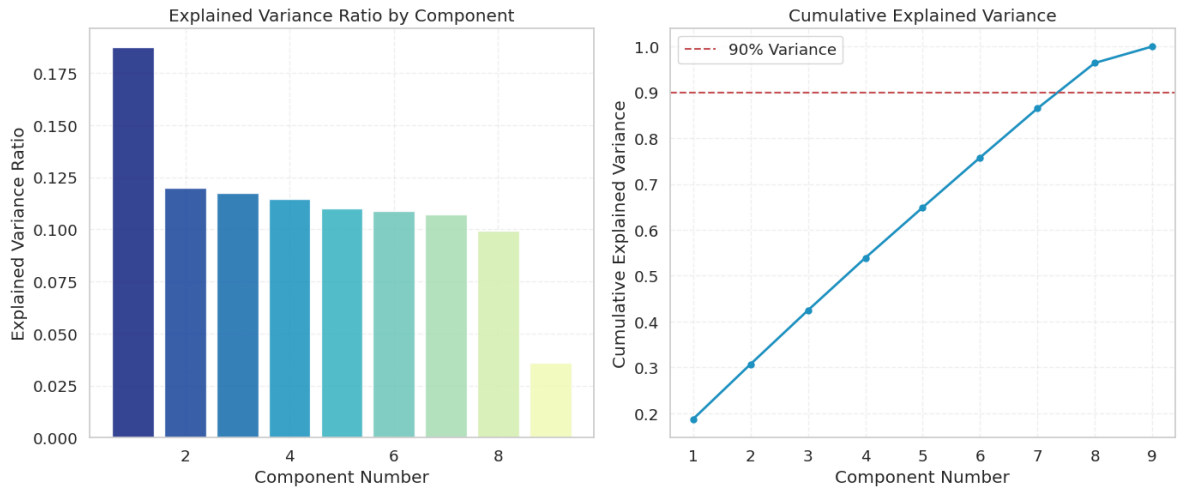

Figure 2. Analysis of the accumulated variance and number of components using Kernel PCA

Considering the previously established parameters, Kernel PCA transforms the data into a new space R^p $(p \leq n)$ using the linear kernel. The transformation is defined as: $\phi(z_i) = V^T * (z_i - z_{mean})$, where V represents the eigenvector matrix and $\phi(z_i)$ is the transformation to the reduced space. The result is X_m^{KPCA}, a reduced representation of X_m^{scaled}.

d) Generation of Synthetic Data

- n_{sin} points are randomly selected from X_m^{KPCA}.
- For each point $p \in X_m^{KPCA}$, its k-nearest neighbors are identified using the Euclidean distance. Neighbors of p: $\{v_1, v_2, v_3, \ldots, v_k\}$.
- Each synthetic point s is generated as $s = p + \varepsilon + average\ (\{v_i - p | \ i = 1, \ldots, k\})$, where:

  $\varepsilon \sim N(0, \alpha)$ is the random perturbation with variance controlled by α.

e) Inverse Transformation

- The synthetic data is projected back to the original space using the inverse transformation of Kernel PCA, denoted as: $s^{scaled} = \phi^{-1}(s), s \in X_{sint}^{KPCA}$.
- Subsequently, the data is denormalized: $s_{original} = s^{scaled} * \sigma + \mu$. Finally, the values are restricted to the valid range of each descriptor, through: $s_{clipped} = min(max(s_{original}, min_j), max_j)$.

f) Combined and labeled

- The original data and the synthetic data are combined into the set represented as $D_{balanced} = D \cup \{s_{clipped}, y = 1 | \forall s \in X_{sint}\}$.

Considering the proposed method, it was applied to the dataset to balance the data of both classes, along with two oversampling techniques, ADASYN and SMOTE. The results are reflected in Table 1, which shows the impact of different synthetic data generation techniques on an originally imbalanced dataset, with a majority class of 99000 records and a minority class of only 1000 records. When applying the techniques, the minority class significantly increases in all techniques, reaching values close to 99000 records. This results in a variation of the minority class of

approximately 50% across all techniques, highlighting the effectiveness of these methodologies in balancing the dataset and bringing the class distribution closer together.

Table 1. Generation of Synthetic Data with Oversampling Techniques

	Dataset original	ADASYN	SMOTE	SynKGen
Class 0	99000	99000	99000	99000
Class 1	1000	98940	99000	99000
Total	100000	197940	198000	198000
Class Variation	1%	49.48%	50%	50%

2.4 ML models and performance evaluation

Detecting fraud in credit card transactions represents a significant challenge for financial institutions. To address this issue, it is essential to have efficient and robust ML models, especially when data imbalance is significant, as is the case with the dataset used, and when there is a need to achieve high precision and low cost for false negatives. For this case, the models XGBoost, RandomForest, AdaBoost, and VotingClassifier have been chosen for fraud detection in credit cards. This approach ensures greater accuracy, reliability, and efficiency in a critical problem where early fraud detection is essential to minimize losses and protect users. All models were implemented using the Sklearn library with their default parameters.

The dataset was divided into 80% for training and 20% for testing. The performance evaluation of each of the implemented models was carried out using different metrics such as accuracy, precision, recall, F1-Score, area under the curve (AUC), Matthew's correlation coefficient (MCC), and G-Mean, whose mathematical expressions are described as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - score = 2 * \frac{Precision\ x\ Recall}{Precision + Recall} \tag{5}$$

$$AUC = 1 - \frac{FP + FN}{TP + TN} \tag{6}$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{7}$$

$$G - Mean = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}} \tag{8}$$

where TP, FP, TN, and FN represent the number of true positives, false positives, true negatives, and false negatives, respectively.

The reason for including the MCC and G-Mean metrics in this evaluation is that they allow for a more balanced and accurate assessment of the classifiers compared to the other established metrics. MCC has the advantage of being a balanced metric, as it penalizes incorrect predictions and provides a better measure of the overall quality of the classifier. An MCC value close to 1

indicates a model with excellent performance in classifying both classes, while a value close to -1 indicates a model that predicts incorrectly (Yang et al., 2024). On the other hand, G-Mean is useful in problems where both classes are important, as it requires classifiers to not only correctly identify positive cases but also minimize false negatives and false positives (Tang & He, 2015).

3. RESULTS AND DISCUSSION

This section presents and analyzes the results obtained from evaluating the four ML classifiers for detecting fraudulent credit card transactions: XGBoost, Random-Forest, AdaBoost, and VotingClassifier.

Table 2 presents an analysis of the training and inference times used by different classifiers applied to the datasets. The SynKGen method shows interesting behavior in terms of training time (84.8718 seconds) and inference time (0.8214 seconds) for the RandomForest classifier. Although it is not the fastest with other classifiers, it positions itself as a robust option compared to ADASYN and SMOTE. In the case of the XGBoost model, it presents a training time of 1.5230 seconds with SynKGen, which is higher than SMOTE (3.0505 seconds), but much faster than ADASYN (1.3143 seconds). XGBoost demonstrates the shortest training and inference time with ADASYN, however, VotingClassifier requires the longest time at 135.6810 seconds. Regarding inference time, SynKGen has competitive performance for RandomForest and VotingClas-sifier. Although classifiers on other datasets are generally more efficient in terms of training and inference times, SynKGen stands out as an option that offers a good balance between generating quality synthetic data and reasonable computational times. In the original dataset, XGBoost presents the best time for training and inference.

Table 2. Training and inference time (seconds)

Classifiers	Dataset original		ADASYN		SMOTE		SynKGen	
	Train Time	Inference Time	Train Time	Inference Time	Train Time	Inference Time	Train Time	Inference Time
XGBoost	0.63120	0.0882	**1.31430**	0.1579	3.0505	0.1795	1.5230	0.2382
RandomForest	35.4048	0.5618	91.5989	1.5918	91.7315	1.6850	**84.8718**	**0.8214**
AdaBoost	6.69060	0.2413	**13.7830**	**0.5574**	15.3522	0.5598	29.5986	0.5693
VotingClassifier	41.3928	0.7956	105.8989	2.0855	114.7821	2.4540	**135.681**	**1.6520**

Bold text: Shorter training and inference time

For the four evaluated classifiers, their respective confusion matrices were obtained on the test group, composed of 20% of the total data. These matrice reflect the performance of each model in terms of true positives, true negatives, false positives, and false negatives, providing a detailed view of their classification capability (Table 3).

Table 3. Confusion matrix results for data augmentation techniques

Classifier	Original Dataset		ADASYN		SMOTE		SynKGen		Class
	0	1	0	1	0	1	0	1	
XGBoost	19786	1	17391	2363	17,445	2332	19776	1	0
	213	0	926	18908	875	18948	201	19622	1
RandomForest	19787	0	19337	417	19,361	416	19777	0	0
	213	0	178	19656	184	19639	200	19623	1
AdaBoost	19787	0	14465	5289	14,331	5446	19777	0	0
	213	0	4281	15553	4195	15628	431	19392	1
VotingClassifier	19787	0	18706	1048	18,747	1030	19777	0	0
	213	0	193	19641	185	19638	200	19623	1

Legend: (0): normal transactions (1): fraudulent transactions

Based on the confusion matrix, several performance metrics were calculated (Table 4). Although XGBoost, RandomForest, AdaBoost, and VotingClassifier show high accuracy (>0.9893), all present precision, recall, F1-Score, MCC, and G-Mean values near or equal to zero, indicating poor detection of positive cases. Despite an AUC of 0.9892, suggesting moderate class discrimination, the low MCC confirms minimal correlation between predictions and true labels. This highlights a common issue in imbalanced datasets: high accuracy, masking poor minority class performance. A G-Mean of 0 further confirms the lack of balance between class-specific performance.

Table 4. Confusion matrix metrics for the original dataset.

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	MCC	G-Mean
XGBoost	0.9893	0.0000	0.0000	0.0000	0.9892	-0.0007	0.0000
RandomForest	0.9894	0.0000	0.0000	0.0000	0.9892	0.0000	0.0000
AdaBoost	0.9894	0.0000	0.0000	0.0000	0.9892	0.0000	0.0000
VotingClassifier	0.9894	0.0000	0.0000	0.0000	0.9892	0.0000	0.0000

When analyzing the classifiers on the synthetic dataset generated by ADASYN (Table 5), a significant improvement in performance is observed. RandomForest stands out as the most accurate model, achieving the best metrics: accuracy (0.985), precision (0.9792), recall (0.991), F1-Score (0.9851), AUC (0.9847), MCC (0.97), and G-mean (0.9849). It is followed by VotingClassifier, with very similar results, particularly excelling in recall (0.9903) and F1-Score (0.9694), indicating a strong ability to detect positive cases. AdaBoost shows the weakest performance, with an average value of 0.7156 across all metrics. XGBoost demonstrates intermediate performance, standing out in recall (0.9533) and G-mean (0.9187), though still below the most effective models.

Table 5. Confusion matrix metrics for the ADASYN synthetic dataset.

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	MCC	G-Mean
XGBoost	0.9169	0.8889	0.9533	0.9200	0.9094	0.8360	0.9161
RandomForest	**0.9850**	**0.9792**	**0.9910**	**0.9851**	**0.9847**	**0.9700**	**0.9849**
AdaBoost	0.7583	0.7462	0.7842	0.7647	0.6812	0.5171	0.7578
VotingClassifier	0.9687	0.9493	0.9903	0.9694	0.9676	0.9381	0.9684

Bold text: Best results

On the synthetic dataset generated with SMOTE, RandomForest remains the top-performing and most balanced model across all metrics. VotingClassifier follows closely, excelling in recall (0.9907), F1-Score (0.97), accuracy (0.9395), and G-mean (0.9691), indicating strong detection and balance. XGBoost shows good results in recall (0.9559) and G-mean (0.9182) but lags behind with an MCC of 0.8403. AdaBoost has the weakest performance in all metrics. Full results are shown in Table 6.

Table 6. Confusion matrix metrics for the SMOTE synthetic dataset.

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	MCC	G-Mean
XGBoost	0.9190	0.8904	0.9559	0.9220	0.9119	0.8403	0.9182
RandomForest	**0.9848**	**0.9792**	**0.9907**	**0.9850**	**0.9846**	**0.9697**	**0.9848**
AdaBoost	0.7565	0.7416	0.7884	0.7643	0.6781	0.5141	0.7558
VotingClassifier	0.9693	0.9502	**0.9907**	0.9700	0.9683	0.9395	0.9691

Bold text: Best results

Finally, for the synthetic dataset generated with the proposed method, called SynKGen, the results shown in Table 7 highlight the RandomForest and VotingClassifier classifiers for their highest values across all metrics, with an MCC of 0.9900 and G-mean of 0.9949, indicating excellent balance between classes and high classification capability, without showing bias towards any of

the evaluated classes. XGBoost has equally exceptional results, achieving an MCC of 0.9898 and a G-mean of 0.9949, reflecting an equally balanced and robust performance. On the other hand, AdaBoost shows slightly lower performance, but the results compared to tests on other datasets are superior. Overall, all classifiers applied to the dataset show excellent performance, but VotingClassifier and RandomForest particularly stand out for their superior values especially in the MCC and G-mean metrics, indicating very balanced classification.

Table 7. Confusion matrix metrics for the SynKGen synthetic dataset.

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	MCC	G-Mean
XGBoost	**0.9949**	0.9999	**0.9899**	**0.9949**	**0.9949**	0.9898	**0.9949**
RandomForest	**0.9949**	**1.0000**	**0.9899**	**0.9949**	**0.9949**	**0.9900**	**0.9949**
AdaBoost	0.9891	**1.0000**	0.9783	0.9898	0.9890	0.9785	0.9891
VotingClassifier	**0.9949**	**1.0000**	**0.9899**	**0.9949**	**0.9949**	**0.9900**	**0.9949**

Bold text: Best results

When comparing the classifiers results using the original dataset, low performance is observed across most evaluated metrics, mainly due to class imbalance. By applying data augmentation techniques such as ADASYN and SMOTE, the RandomForest model shows a notable improvement in performance. However, when classifiers are trained on data generated by SynKGen, a significant improvement is evident, achieving the best results across all metrics and classifiers, particularly with RandomForest (Figure 3). These results confirm that SynKGen is a superior solution, as it not only increases precision but also enhances the models' ability to handle class imbalance. Therefore, it emerges as a promising alternative for data augmentation and improving credit card fraud detection in the banking sector.

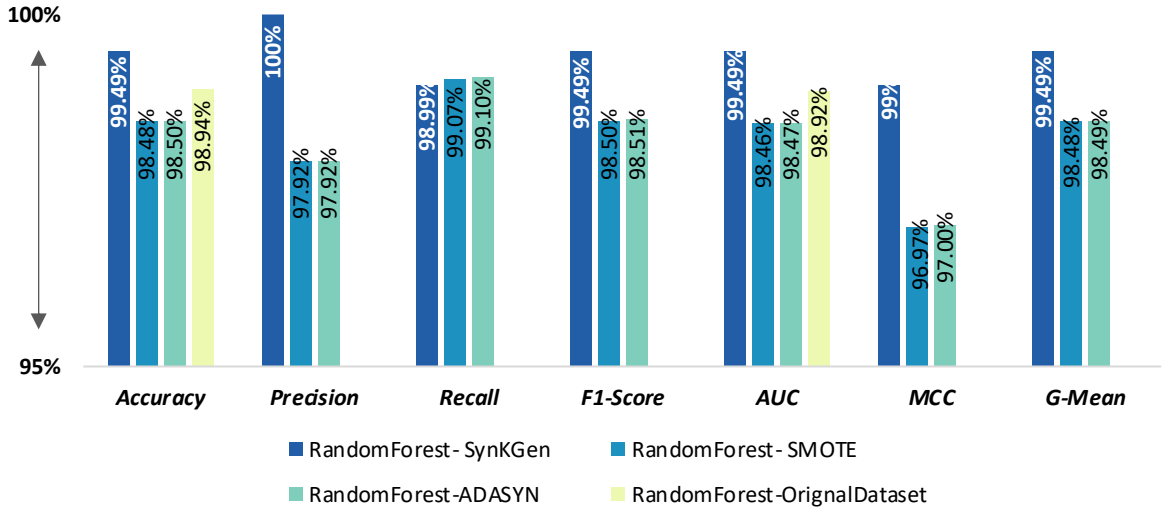

Figure 3. Performance of the RandomForest classifier on confusion matrix metrics with and without data augmentation

Additionally, a comparison of data augmentation techniques was conducted using the F1-Score metric as the dependent variable, with the implemented classifiers evaluated through cross-validation ($k = 5$). A two-way ANOVA revealed that the augmentation technique ($F = 2.00e+06$, $p < 0.001$, $\eta^2 = 0.9999$), the classifier ($F = 4.64e+04$, $p < 0.001$, $\eta^2 = 0.9995$), and their interaction ($F = 6.70e+03$, $p < 0.001$, $\eta^2 = 0.9989$) all had highly significant effects on the dependent variable. Since the residuals did not meet the normality assumption (Shapiro-Wilk $p < 0.001$), a Friedman test was applied, confirming significant differences between groups ($p < 0.001$). Subsequently, Dunn's post-hoc test with Holm correction was performed, identifying that SynKGen showed significant

differences compared to SMOTE (p = 0.0019) and ADASYN (p = 0.0015), while no significant differences were found between SMOTE and ADASYN (p = 0.8563). These findings indicate that the use of SynKGen significantly contributes to improving model performance compared to traditional oversampling techniques (Figure 4).

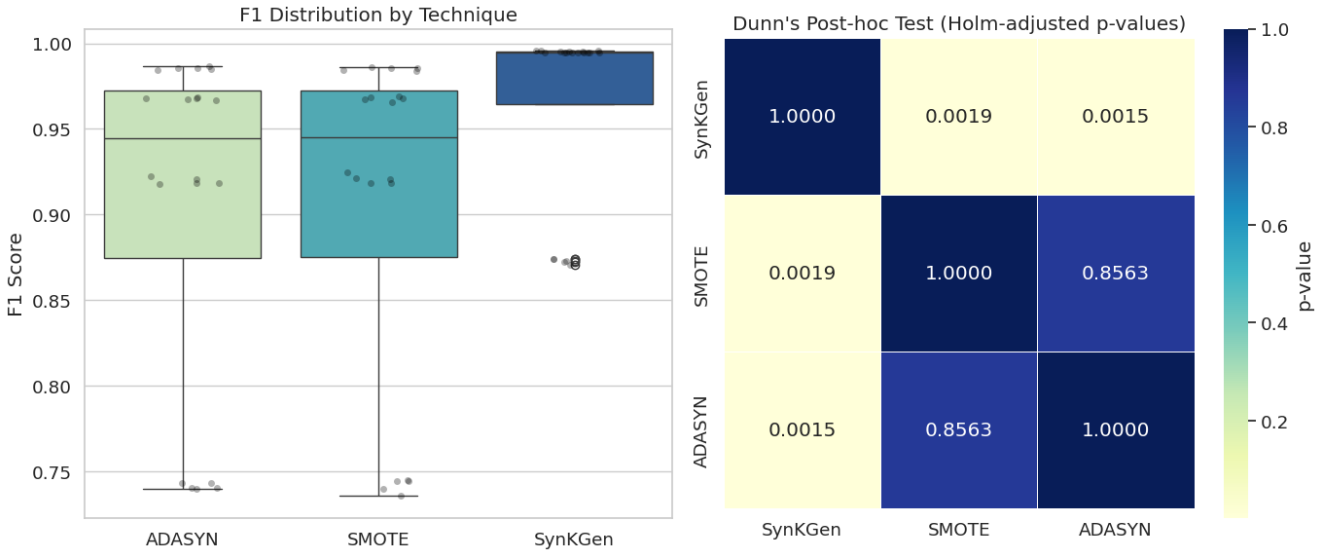

Figure 4. Impact of data augmentation techniques and classifiers on F1-Score using two-way ANOVA and Post-Hoc Dunn analysis

In Table 8, the results of the comparative analysis of the proposed method with other studies are presented. The proposed RandomForest-SynKGen and VotingClassifier-SynKGen model stands out with exceptional performance, achieving 100% precision and an F1-Score of 0.9949, indicators that considerably surpass traditional methods. While more advanced approaches like Bi-3DQRNN-PoV and Structured Data Transformer (SDT) show competitive results with accuracies above 97%, they fail to match the effectiveness of the proposed method in terms of overall balance. The high AUC value (0.9949) and MCC (0.99) of the proposed model suggest an out-standing ability to distinguish between legitimate and fraudulent transactions, over-coming the limitations of previous models such as AllKNN-CatBoost which, despite its high accuracy (0.9996), shows a low recall (0.9591), indicating poor detection of positive cases, and even more so when evaluating the F1-Score (0.874). The proposed method represents a significant step in the fight against financial fraud, providing a more reliable and accurate tool for financial institutions.

Table 8. Comparison of the results obtained with other studies.

Ref	Approach	Accuracy	Precision	Recall	F1-Score	AUC	MCC
(Alfaiz & Fati, 2022)	AllKNN-CatBoost	0.9996	0.8028	0.9591	0.8740	0.9790	-
(Rb & Kr, 2021)	SVM	0.9349	0.9743	0.8976	-	-	-
	KNN	0.9982	0.7142	0.0393	-	-	-
	ANN	0.9992	0.8115	0.7619	-	-	-
(Ranganatha & Syed, 2025)	Bi-3DQRNN-PoV	0.9710	0.9700	0.9800	-	-	-
(Ileberi & Sun, 2024)	Proposed stacking Ensemble	-	0.9890	-	0.9750	0.9720	-
	Proposed DL ensemble	-	0.9650	-	0.9410	0.9200	-
(Tang & Liu, 2024)	Transformador de Datos Estructurados (SDT) y aprendizaje federado	0.9970	0.9610	0.7340	0.7540	0.9940	-
(Le et al., 2024)	Efficient KAN	-	-	-	-	0.9900	-

(Adil et al., 2024)	Optimized Deep Event-based Network (OptDevNet)	0.9989	-	-	0.9980	-	-
(Coello et al., 2023)	ADASYN-RandomForest	-	-	-	0.8677	0.9900	-
	Random Over-sampling - XGBoost	-	-	-	0.9800	0.8470	-
	ADASYB-DecisionTree	-	-	-	0.9600	0.7640	-
(Mondal et al., 2021)	Ensemble model +GAN	-	0.9800	0.9000	-	0.9800	-
(Zhao et al., 2024)	CB-CHL-LGBM	-	0.6720	-	0.8280	0.9970	0.7640
Our study	RandomForest-SynKGen/ VotingClassifier-SynKGen	0.9949	1.0000	0.9899	0.9949	0.9949	0.9900

CONCLUSIONS

This study has demonstrated that, despite the advancements in various credit card fraud detection techniques implemented in the different studies analyzed, such as the use of deep neural networks, machine learning algorithms, and hybrid methods, significant limitations persist that affect the performance of these techniques in ad-dressing this problem. These limitations are particularly noticeable in scenarios with pronounced data imbalance and when greater generalization capacity is required to detect emerging fraud patterns.

To address these limitations, a new method called SynKGen was proposed, and its performance was comparatively evaluated against traditional oversampling techniques such as ADASYN and SMOTE. Experimental results show a significant improvement in the performance of the implemented classifiers, achieving metrics above 0.99. SynKGen represents an innovative and effective solution for handling data im-balance, substantially improving accuracy in credit card fraud detection.

However, for this model to be viable in production environments, it is essential to consider its integration into real-time fraud detection systems, particularly its ability to generate synthetic samples efficiently and adaptively as transactions are processed. Preliminary tests suggest that SynKGen can be implemented as a preprocessing module in existing machine learning pipelines used by financial institutions, provided adequate computational resources and batch-processing strategies are employed.

It is important to note that the proposed SynKGen method requires various tests to optimize the configuration of the parameters used until a better representation of the original dataset is achieved, such as the type of kernel used (kernel="linear"), the number of components (n_components=8), the alpha value (0.01), and the number of neighbors (n_neighbors=5). This could be a limitation compared to other evaluated methods; however, it could be overcome by using hyperparameter optimization techniques.

For future research, it is recommended to explore the application of the proposed method on other datasets, as well as its implementation with other classifiers or more advanced methods like convolutional neural networks. Additionally, it would be beneficial to investigate the application of real-time models and their ability to adapt to new fraud strategies as they emerge. Collaboration between financial and academic institutions is of great importance, as it could facilitate access to more diverse and up-dated datasets, contributing to improving the accuracy and effectiveness of credit card fraud detection models.

FINANCING

The author did not receive sponsorship to carry out this study-article.

CONFLICT OF INTEREST

There is no conflict of interest related to the subject matter of the work.

AUTHORSHIP CONTRIBUTION

Conceptualization, formal analysis, software, supervision, validation, and writing – review and editing were carried out by Fray L. Becerra-Suarez. Data curation, methodology, and writing – original draft were the responsibility of Fray L. Becerra-Suarez and Luciani J. Jiménez-Fernández. The investigation was conducted by Fray L. Becerra-Suarez and Luciani J. Jiménez-Fernández. Acquisition of funds and project administration were managed by Luciani J. Jiménez-Fernández, Estrella D. Ticona-Tapia, José R. Cardenas-Gonzales, and Pepe H. Bustamante-Quintana. Resources and visualization were provided by Estrella D. Ticona-Tapia, José R. Cardenas-Gonzales, and Pepe H. Bustamante-Quintana.

REFERENCES

Adil, M., Yinjun, Z., Jamjoom, M. M., & Ullah, Z. (2024). OptDevNet: A Optimized Deep Event-Based Network Framework for Credit Card Fraud Detection. *IEEE Access*, 12, 132421–132433. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3458944

Alatawi, M. N. (2025). Detection of fraud in IoT based credit card collected dataset using machine learning. *Machine Learning with Applications*, 19, 100603. https://doi.org/10.1016/j.mlwa.2024.100603

Alfaiz, N. S., & Fati, S. M. (2022). Enhanced Credit Card Fraud Detection Model Using Machine Learning. *Electronics*, 11(4), Article 4. https://doi.org/10.3390/electronics11040662

Attouri, K., Mansouri, M., Hajji, M., Kouadri, A., Bensmail, A., Bouzrara, K., & Nounou, H. (2024). Improved fault detection based on kernel PCA for monitoring industrial applications. *Journal of Process Control*, 133, 103143. https://doi.org/10.1016/j.jprocont.2023.103143

Becerra-Suarez, F. L., Fernández-Roman, I., & Forero, M. G. (2024). Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing. *Mathematics*, 12(9), Article 9. https://doi.org/10.3390/math12091294

Charizanos, G., Demirhan, H., & İçen, D. (2024). An online fuzzy fraud detection framework for credit card transactions. *Expert Systems with Applications*, 252, 124127. https://doi.org/10.1016/j.eswa.2024.124127

Chatterjee, P., Das, D., & Rawat, D. B. (2024). Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements. *Future Generation Computer Systems*, 158, 410–426. https://doi.org/10.1016/j.future.2024.04.057

Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University - Computer and Information Sciences*, 35(1), 145–174.

https://doi.org/10.1016/j.jksuci.2022.11.008

Coello, K., Zhou, K., Nutalapati, H., & Tiglao, N. M. C. (2023). Performance Analysis of Credit Card Fraud Analysis and Detection Machine Learning Algorithms. *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6. https://doi.org/10.1109/ISNCC58260.2023.10323945

Dastidar, K. G., Caelen, O., & Granitzer, M. (2024). Machine Learning Methods for Credit Card Fraud Detection: A Survey. IEEE Access, 12, 158939–158965. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3487298

Hasan, M., Hoque, A., & Le, T. (2023). Big Data-Driven Banking Operations: Opportunities, Challenges, and Data Security Perspectives. *FinTech*, 2(3), Article 3. https://doi.org/10.3390/fintech2030028

Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Systems with Applications*, 193, 116429. https://doi.org/10.1016/j.eswa.2021.116429

Ileberi, E., & Sun, Y. (2024). A Hybrid Deep Learning Ensemble Model for Credit Card Fraud Detection. IEEE Access, 12, 175829–175838. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3502542

Interbank. (2024). *Banca por Internet: Es tiempo de ir por más - Interbank*. https://interbank.pe/comunicado

Jain, Y., Tiwari, N., Dubey, S., & Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques. *International Journal of Recent Technology and Engineering*, 7, 402–407.

Juniper Research. (n.d.). Online Payment Fraud Losses to Exceed $206 Billion Over the Next Five Years | Press. Retrieved December 16, 2024, from https://www.juniperresearch.com/press/online-payment-fraud-losses-to-exceed-206-billion/

Kaggle. (2024). Retrieved December 22, 2024, from https://www.kaggle.com/datasets/bhadramohit/credit-card-fraud-detection

Kaib, M. T. H., Kouadri, A., Harkat, M. F., Bensmail, A., & Mansouri, M. (2025). Data size reduction approach for nonlinear process monitoring refinement using Kernel PCA technique. *Expert Systems with Applications*, 274, 126975. https://doi.org/10.1016/j.eswa.2025.126975

Lazcano, A., & Jaramillo-Morán, M. A. (2025). Data preprocessing techniques and neural networks for trended time series forecasting. *Applied Soft Computing*, 174, 113063. https://doi.org/10.1016/j.asoc.2025.113063

Le, T.-T.-H., Hwang, Y., Kang, H., & Kim, H. (2024). Robust Credit Card Fraud Detection Based on Efficient Kolmogorov-Arnold Network Models. *IEEE Access*, 12, 157006–157020. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3485200

Mondal, I. A., Haque, Md. E., Hassan, A.-M., & Shatabda, S. (2021). Handling Imbalanced Data for Credit Card Fraud Detection. *2021 24th International Conference on Computer and Information Technology (ICCIT)*, 1–6. https://doi.org/10.1109/ICCIT54785.2021.9689866

Ranganatha, H. R., & Syed, A. (2025). Enhancing fraud detection efficiency in mobile transactions through the integration of bidirectional 3d Quasi-Recurrent Neural network and blockchain technologies. *Expert Systems with Applications*, 260, 125179. https://doi.org/10.1016/j.eswa.2024.125179

Rb, A., & Kr, S. K. (2021). Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, 2(1), 35–41. https://doi.org/10.1016/j.gltp.2021.01.006

Sulaiman, S. S., Nadher, I., & Hameed, S. M. (2024). Credit Card Fraud Detection Using Improved Deep Learning Models. *Computers, Materials and Continua*, 78(1), 1049–1069. https://doi.org/10.32604/cmc.2023.046051

Tang, B., & He, H. (2015). KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning. *2015 IEEE Congress on Evolutionary Computation (CEC)*, 664–671. https://doi.org/10.1109/CEC.2015.7256954

Tang, Y., & Liu, Z. (2024). A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning. IEEE Access, 12, 182547–182560. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3491175

Wijaya, M. G., Pinaringgi, M. F., Zakiyyah, A. Y., & Meiliana. (2024). Comparative Analysis of Machine Learning Algorithms and Data Balancing Techniques for Credit Card Fraud Detection. *Procedia Computer Science*, 245, 677–688. https://doi.org/10.1016/j.procs.2024.10.294

Yang, Z., Wang, Y., Shi, H., & Qiu, Q. (2024). Leveraging Mixture of Experts and Deep Learning-Based Data Rebalancing to Improve Credit Fraud Detection. *Big Data and Cognitive Computing*, 8(11), Article 11. https://doi.org/10.3390/bdcc8110151

Zhang, C., Nie, F., & Xiang, S. (2010). A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomputing*, 73(4), 959–967. https://doi.org/10.1016/j.neucom.2009.08.014

Zhao, X., Liu, Y., & Zhao, Q. (2024). Improved LightGBM for Extremely Imbalanced Data and Application to Credit Card Fraud Detection. *IEEE Access*, 12, 159316–159335. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3487212