UDC 004.056.5:[004.032.2:004.85]:[343.72:336.717]
DOI: 10.15587/2706-5448.2025.330639

Mykola Zlobin,
Volodymyr Bazylevych

DEVELOPMENT OF A PREPROCESSING METHODOLOGY FOR IMBALANCED DATASETS IN MACHINE LEARNING TRAINING

The object of the study is an imbalanced dataset of credit card transactions, where fraudulent cases represent only 0.18% of the total. One of the most problematic places is the inability of standard machine learning models to correctly detect rare fraud events, often resulting in high false-negative rates. This occurs because the models focus on the majority class, which leads to biased outcomes and undetected fraud. The presented analyses used a structured preprocessing pipeline to address this issue. It includes scaling of numeric values to eliminate bias, stratified sampling to preserve class proportions, random undersampling to balance the dataset, and outlier removal to reduce noise. These steps were applied before training three classification models: logistic regression (LR), K-Nearest Neighbors (KNN), and support vector classifier (SVC). The obtained results show that all models performed well in both cross-validation accuracy and ROC-AUC metrics, with SVC achieving the best ROC-AUC score of 0.9787. This is because the proposed preprocessing pipeline has many features customized to the characteristics of imbalanced data, in particular the combination of data balancing with careful filtering of noise and redundancy. This provides the possibility of achieving robust performance when detecting minority class events. Compared with similar known preprocessing workflows, it provides the following advantages: better class separation, reduced model bias, and improved generalization on unseen data. The results are especially relevant for financial institutions, where fraud detection must be both timely and accurate. The approach offers a practical method for improving security systems without requiring complex or high-cost infrastructure. It can also be adapted for use in other domains where rare events must be detected from large datasets. In future research, the pipeline can be extended by integrating synthetic sampling techniques such as SMOTE or GANs. Additional experiments with real-time streaming data will further validate the robustness of the proposed methodology.

Keywords: *imbalanced classification, fraud detection pipeline, stratified sampling, outlier removal, support vector classifier.*

Received: 06.03.2025
Received in revised form: 26.04.2025
Accepted: 19.05.2025
Published: 27.05.2025

How to cite
Zlobin, M., Bazylevych, V. (2025). Development of a preprocessing methodology for imbalanced datasets in machine learning training. Technology Audit and Production Reserves, 3 (2 (83)), 55–61. https://doi.org/10.15587/2706-5448.2025.330639

1. Introduction

In machine learning, imbalanced datasets present significant challenges, especially in applications like credit card fraud detection. An imbalanced dataset occurs when one class is significantly underrepresented compared to another. For example, in credit card transactions, fraudulent activities constitute a very small fraction of the total transactions, which leads to a class imbalance that complicates the training of effective predictive models. Credit card fraud represents a growing concern globally. Financial institutions face substantial losses due to fraudulent transactions. For instance, in 2018, global losses due to credit card fraud were estimated at 27.85 billion USD, with projections reaching 35.67 billion USD by 2023. This escalating trend underscores the urgency for fraud detection mechanisms [1, 2]. These models are employed to detect fraudulent transactions by identifying patterns and anomalies within transaction data. Standard machine learning algorithms tend to be biased towards the majority class, leading to poor predictive performance for the minority class. This bias results in high false-negative rates, where fraudulent transactions go undetected, undermining the effectiveness of fraud detection systems [3, 4]. There are strategies that have been proposed to address the challenges posed by imbalanced datasets. Data-level techniques, such as under-sampling the majority class or over-sampling the minority class, aim to balance the class distribution. Algorithm-level approaches involve modifying existing algorithms to make them more sensitive to the minority class. Hybrid methods combine both data-level and algorithm-level techniques to improve model performance. Cost-sensitive learning assigns higher misclassification costs to the minority class to mitigate bias. Deep learning approaches have also been explored to handle class imbalance effectively [5]. In the context of credit card fraud detection, the application of these techniques has shown promise. For example, a study employing a genetic algorithm for feature selection in a machine learning-based fraud detection system demonstrated improved detection rates. Another approach used a semi-supervised graph neural network to both labeled and unlabeled data, achieving improved fraud detection performance. These studies highlight the potential of machine learning techniques in addressing the challenges posed by imbalanced datasets in fraud detection [6].

In order to effectively optimize unbalanced datasets one common approach is resampling the dataset to balance class distributions. This can be achieved through under-sampling, where instances from the majority class are randomly removed to match the minority class size, or over-sampling, where instances from the minority class are duplicated or synthetically generated to match the majority class size. A widely used over-sampling technique is the Synthetic Minority Over-sampling Technique (SMOTE), which creates synthetic samples by interpolating between existing minority class instances. Studies have shown that

ISSN-L 2664-9969; E-ISSN 2706-5448

combining SMOTE with convolutional neural networks (CNNs) can improve classification performance on imbalanced datasets, achieving accuracy rates as high as 99.08% on certain datasets [7, 8]. Another strategy is the use of ensemble learning methods. By combining multiple models, ensemble methods can improve the accuracy of predictions on imbalanced datasets. Research indicates that integrating data augmentation techniques, such as SMOTE, with ensemble learning can improve classification performance on imbalanced datasets. Traditional data augmentation methods like SMOTE and random over-sampling have been found to be not only effective but also computationally less expensive compared to more complex methods like Generative Adversarial Networks (GANs) [9, 10].

Additionally, modifying the loss function to assign higher penalties to misclassifications of the minority class can guide the model to pay more attention to these instances. Techniques such as weighted cross-entropy loss and focal loss have been proposed to address class imbalance by adjusting the loss function to focus more on hard-to-classify minority instances [11].

Despite these findings, challenges remain in handling imbalanced datasets. Issues such as loss of predictive accuracy, biased models, and unreliable evaluation metrics persist.

In this paper, the optimization of preprocessing techniques for handling highly imbalanced datasets in machine learning is presented. The paper focuses on credit card fraud detection, where fraudulent transactions represent only a small fraction of the total dataset. The paper examines the impact of various preprocessing techniques, including scaling, stratified sampling, under-sampling, and outlier removal, in improving classification performance. Additionally, three widely used classification models: LR, KNN, and SVC are evaluated to determine their effectiveness in identifying fraudulent transactions. The results highlight the significance of preprocessing in mitigating the challenges posed by class imbalance. This shows that machine learning models can accurately learn from minority class instances and generalize effectively in fraud detection applications. The problem addressed in this research is the poor performance of traditional machine learning models on highly imbalanced datasets, particularly in detecting credit card fraud, where fraudulent transactions represent a very small fraction of all transactions. The purpose of the analysis is to develop and validate an optimized data preprocessing methodology that improves the classification performance of standard machine learning models under extreme class imbalance conditions. This includes evaluating the impact of scaling, stratified sampling, under-sampling, and outlier removal techniques on the detection of fraudulent transactions.

Thus, *the aim of research* is to develop and validate a structured preprocessing methodology that addresses the challenges of extreme class imbalance in credit card fraud detection, with the aim of improving the classification performance of traditional machine learning models through effective scaling, sampling, and noise reduction techniques.

2. Materials and Methods

2.1. Dataset description

The object of this study is a real-world dataset of credit card transactions, which is imbalanced. It contains a total of 284,807 transaction records, of which only 492 are labeled as fraudulent (labeled as 1), while all others are regular transactions (labeled as 0). This represents just 0.18% of all transactions, with the severity of class imbalance in the dataset. The data was originally collected as part of a research collaboration between Worldline and the Machine Learning Group of Université Libre de Bruxelles on fraud detection [12]. All transactions were conducted by European cardholders over 48 hours (Fig. 1). The subsequent subsections describe the preprocessing methodology applied to this dataset and the construction of classification models for fraud detection.

The dataset is predominantly anonymized due to confidentiality requirements and includes 28 transformed, randomly labeled features (V1 to V28). The only known and untransformed features are "Time" and "Amount". "Time" represents the time elapsed between the ongoing transaction and the initial transaction in the dataset, while "Amount" indicates the monetary value of each transaction. Data will be provided upon reasonable request. Fig. 2, 3 illustrate the skewed distributions of these two features, highlighting the density of transaction amounts and the time periods with the highest transaction activities.

	scaled_amount	scaled_time	V1	V2	...	V26	V27	V28	Class
166305	0.632991	0.391170	1.933247	-0.110500	...	-0.117379	-0.022554	-0.003557	0
137933	-0.222874	-0.026915	1.174063	0.309198	...	-0.582201	0.014215	0.018129	0
85772	0.330189	-0.279127	1.012552	0.111077	...	0.174260	0.038842	0.032227	0
137164	-0.254873	-0.031145	-1.513005	0.935000	...	0.531532	-0.549607	-0.292961	0
61523	0.083840	-0.409004	-0.780675	-0.172942	...	0.241945	0.021715	0.090899	0

Fig. 1. The example columns of the dataset

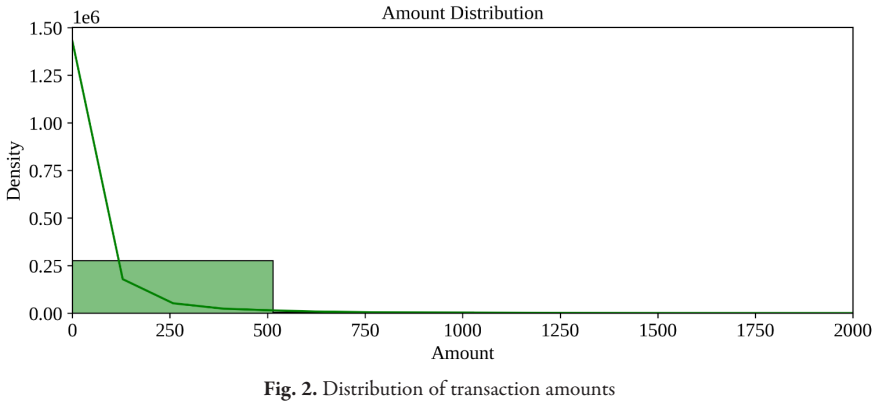

Fig. 2. Distribution of transaction amounts

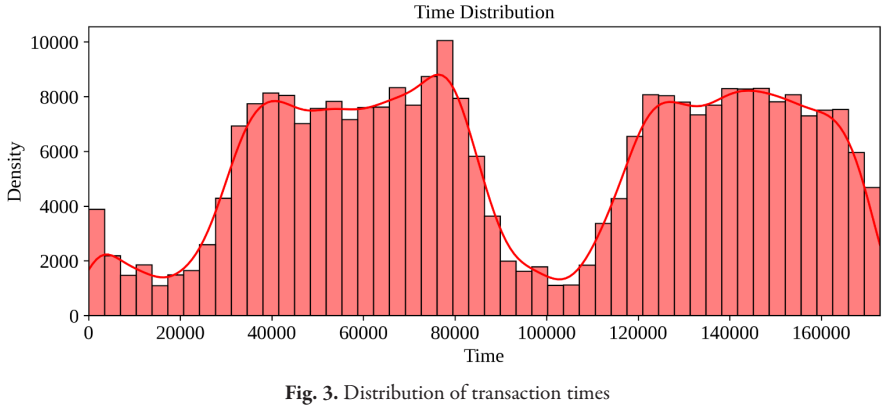

Fig. 3. Distribution of transaction times

The presented Fig. 2 shows that the majority of transactions fall within a low monetary range, with a steep decline beyond 250 units. This distribution highlights the importance of scaling in order to ensure that large values do not dominate the training process.

In Fig. 3, the transaction activity varies significantly across the 48-hour window. Multiple peaks indicate time periods of increased transaction frequency, which may correspond to patterns of user behavior or fraud attempts.

2.2. Optimizing unbalanced data

As confirmed in the previous subsection, the dataset used in this research is highly unbalanced. To optimize it and make it suitable for training ML models, this section will present a systematic approach to handling this data:

1. Scaling.

The first step is to scale the "Amount" and "Time" columns using the following approach

$$X_{scaled} = \frac{X - median(X)}{IQR(X)}, \quad (1)$$

where X is a raw value, $median(X)$ is the median of all X, and $IQR(X)$ is the interquartile range, calculated as the difference between the 75th and 25th percentiles of the values. By applying the scaling process, all features are properly scaled, preventing biases caused by large differences. Additionally, this serves as the first step in balancing the unbalanced dataset.

2. Creating initial training and test data.

After scaling the features, the next step is to split the data into training and test sets while maintaining the original data distribution. Given that fraudulent transactions account for only 0.18% of the overall data, a random split could result in a critically imbalanced test set, potentially containing no fraud cases at all. This would make it nearly impossible to evaluate a model's performance in detecting fraudulent transactions. To prevent such an occurrence, the following process ensures that both datasets preserve the initial class proportions and maintain balance between the classes, allowing for a fair evaluation of the model's performance. The dataset is divided into input variables (X), which contain all columns except "Class", while y includes the fraud labels. The dataset is then split into five folds while preserving class distributions, preventing cases where either the training or test set contains too few fraud instances. The Stratified K-Fold approach resulted in a balanced proportion of fraud and non-fraud cases compared to the original dataset: fraud cases account for 0.18% (0.00184) in the training set, while the test set maintains an almost identical proportion (0.00185). However, given the extreme imbalance of one class, an additional technique – undersampling – will be applied next.

3. Under-sampling process.

Training a model on data where only 0.18% of instances belong to one class would result in a highly skewed classifier that predominantly predicts non-fraudulent transactions in almost all cases. To prevent this, a balanced dataset is created next by under-sampling the non-fraudulent transactions, ensuring an equal distribution of both classes. To do that, all fraudulent transactions are isolated, and comparable number of non-fraudulent transactions are selected. This results in an almost 50:50 class distribution in the new dataset (Fig. 4), which is then shuffled to randomize the order of instances. A disadvantage of this Random Under-Sampling approach is the potential loss of information, as many non-fraudulent transactions are removed, which may impact the model's ability to generalize effectively.

4. Outliers removal.

The next step is to remove outliers from anonymized features, as they can significantly distort the performance of a model, leading to poor predictions. To address this, the Interquartile Range (IQR) method is applied to all features. The process calculates the 25th percentile (Q_1) and the 75th percentile (Q_3) for each feature, where the difference between them represents the IQR (the spread of the middle 50% of the data). Then, the lower and upper bounds for detecting outliers are calculated as:

$$Lower\ Bound = Q_1 - 1.5 \cdot IQR, \quad (2)$$

$$Upper\ Bound = Q_3 + 1.5 \cdot IQR. \quad (3)$$

Finally, all values x that fall below or above the following bounds are identified and removed as outliers

$$x < Q_1 - 1.5 \cdot IQR \text{ or } x > Q_3 + 1.5 \cdot IQR. \quad (4)$$

5. Preparing data for ML models.

The new dataset, created through the previous four steps, is now optimized and ready for training the models. Eighty percent (80%) of the data is used as the training set, while twenty percent (20%) is allocated for testing. All features, except the target labels, serve as inputs to the models, while the single output indicates whether a fraudulent transaction has occurred.

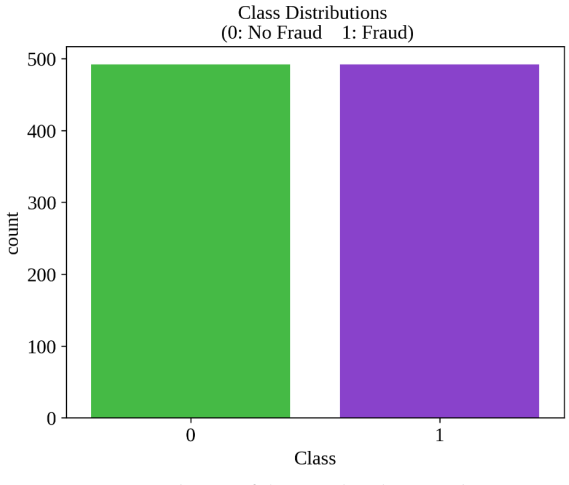

Fig. 4. Distribution of classes within the new subset: class 0–55%; class 1–45%

2.3. Building classification models

For classification purposes in this research, three well-known and efficient models are utilized: LR, KNN, and SVC.

LR represents one of the most commonly used statistical models for binary classification problems. In imbalanced datasets, such as the one analyzed in this research, traditional ML models show difficulties to effectively classify the minority class, which is fraud. LR provides a probabilistic framework that is useful in these cases due to its ability to model the relationship between the predictor variables and the probability of a transaction being fraudulent. Fraud detection is an important area in financial risk management. Fraudulent transactions account for less than 0.2% of all transactions [13]. Traditional ML models trained on imbalanced data tend to favor the majority class, failing to correctly identify fraud cases. Logistic regression, when optimized with preprocessing methods such as undersampling, feature scaling, and outlier removal, provides a solution supporting probability estimation but also maximize model interpretability. It predicts the probability of a binary outcome using a sigmoid function applied to a linear combination of input features. Given an input feature set $X = (x_1, x_2, ..., x_n)$ the logistic function is [3, 14, 15]

$$p(y = 1|X) = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad (5)$$

ISSN-L 2664-9969; E-ISSN 2706-5448

where z is the linear combination of input variables

$$z=\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_nx_n, \quad (6)$$

where β_0 is the intercept term, while $\beta_1,\beta_2,...,\beta_n$ are the coefficients learned by the model to optimize classification. The probability that a given transaction is fraudulent is calculated using the logistic function

$$p\left(y=1|X\right)=\frac{1}{1+e^{-\left(\beta_0+\sum_{i=1}^{n}\beta_ix_i\right)}}. \quad (7)$$

The probability that a transaction is legitimate is

$$p\left(y=0|X\right)=1-p\left(y=1|X\right). \quad (8)$$

To determine the best-fitting parameters (β), the model optimizes the log-likelihood function

$$L\left(\beta\right)=\Pi_{i=1}^{m}p\left(y_i|X_i\right)^{y_i}\left(1-p\left(y_i|X_i\right)\right)^{1-y_i}. \quad (9)$$

Taking the natural logarithm, the log-likelihood function transforms into

$$l\left(\beta\right)=\sum_{i=1}^{m}\left[y_i\log\log p\left(y_i|X_i\right)+\left(1-y_i\right)\log\log\left(1-p\left(y_i|X_i\right)\right)\right]. \quad (10)$$

Since this function is convex, gradient descent or Newton-Raphson methods are used to estimate the coefficients β. Fraud detection models require regularization to prevent overfitting due to the high-dimensional nature of financial transaction data. Two common regularization techniques used in LR are L1 (Lasso) and L2 (Ridge) regularization [16].

L2 regularization, also known as Ridge Regression, adds a penalty proportional to the sum of squared coefficients

$$J\left(\beta\right)=-l\left(\beta\right)+\lambda\sum_{i=1}^{n}\beta_i^2. \quad (11)$$

This approach discourages large coefficient values, reducing model variance and improving generalization. The dataset in this paper uses L2 regularization to guarantee stability and prevent overfitting. LR is seen as an effective model for fraud detection due to its ability to output probabilistic predictions. It allows financial institutions to set probability thresholds for fraud classification. The decision boundary is established based on a threshold value (τ), where

$$Classify\ as\ fraud\ if\ p\left(y_i|X_i\right)>\tau. \quad (12)$$

Since overall accuracy is misleading in imbalanced datasets, evaluation relies on precision, recall, and F1-score, defined as:

$$Precision=\frac{True\ Positives}{True\ Positives+False\ Positives}. \quad (13)$$

$$Recall=\frac{True\ Positives}{True\ Positives+False\ Negatives}. \quad (14)$$

$$F1_{score}=2\cdot\frac{Precision\cdot Recall}{Precision+Recall}. \quad (15)$$

KNN model, on the other hand, represents a non-parametric method for classification and regression tasks. Its simplicity has led to wide application in different domains, including fraud detection [17]. KNN operates on the principle that data points with similar features are likely to belong to the same class. In classification, the algorithm assigns a class to a new data point based on the majority class among its k nearest neighbors in the feature space. For regression tasks, KNN predicts the value of a new data point by averaging the values of its k nearest neighbors. Given a dataset with n instances, each represented by a feature vector x_i and an associated label y_i, the goal is to predict the label y for a new instance x. The steps are given as [18–20]:

1. Compute the distance between x and all instances x_i in the dataset. A common distance metric is the Euclidean distance

$$d\left(x,x_i\right)=\sqrt{\sum_{j=1}^{m}\left(x_j-x_{ij}\right)^2}, \quad (16)$$

where m is the number of features, x_j is the j-th feature of x, and x_{ij} is the j-th feature of x_i.

2. Identify k instances $x_1,x_2,...,x_k$ with the smallest distances to x.

3. Assign the class that is most frequent among the k nearest neighbors, for classification. Or calculate the average value of the k nearest neighbors, for regression

$$\hat{y}=\frac{1}{k}\sum_{i=1}^{k}y\left(i\right). \quad (17)$$

The choice of k influences KNN's performance. A smaller k can lead to models sensitive to noise, while a larger k may smooth out class boundaries. Selecting an optimal k often involves heuristic methods or cross-validation methods. Additionally, the algorithm's effectiveness can be compromised by irrelevant or redundant features. Feature selection or scaling methods are employed to mitigate this issue.

SVCs derived from Support Vector Machines (SVMs), represent supervised learning models widely used for classification tasks. They are used in high-dimensional spaces and applied in bioinformatics, text categorization, and image recognition. Given a training dataset with n instances, each represented by a feature vector x_i and a corresponding label y_i (where y_i is either 1 or – 1), the objective of SVC is to find a hyperplane that best separates the data into two classes. The equation can define this hyperplane

$$w^Tx-b=0, \quad (18)$$

where w is the normal vector to the hyperplane, and b is the bias term. The goal is to maximize the margin, which is the distance between the hyperplane and the nearest data points from either class. This can be achieved by solving the following optimization problem [21–23]

$$minimize\frac{1}{2}\left\|w\right\|^2\ subject\ to\ y_i\left(w^Tx-b\right)\geq1\ \forall i\in\left\{1,2,...,n\right\}. \quad (19)$$

Here the hyperplane not only separates the classes but does so with the maximum possible margin, improving the model's generalization capabilities. In real scenarios, data is often not perfectly linearly separable. To handle such cases, the soft-margin SVC introduces slack variables ξ_i to provide some misclassifications. The optimization problem is modified

$$minimize\frac{1}{2}\left\|w\right\|^2+C\sum_{i=1}^{n}\xi_i\ subject\ to$$
$$y_i\left(w^Tx-b\right)\geq1-\xi_i,\xi_i\geq0,\ \forall i\in\left\{1,2,...,n\right\}, \quad (20)$$

where C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error. SVC finds a balance between fitting the training data and maintaining a large margin, thereby improving its ability to generalize to unseen data. In order to handle non-linearly separable data, SVCs employ the kernel trick, which involves mapping the input features into a higher-dimen-

sional space where a linear separator can be found. This is achieved using kernel functions, such as the radial basis function kernel, which measures the similarity between data points in the transformed feature space. The kernel enables SVCs to perform classification in complex datasets without explicitly computing the high-dimensional transformations. SVCs have been effectively used for imbalanced datasets, such as those encountered in fraud detection, medical diagnosis, and rare event prediction. In such cases, the minority class is often of greater interest but is underrepresented in the data.

All three models are implemented in Python with their default setups to ensure a fair environment for comparing their performances. The LR model uses L2 regularization to prevent overfitting, with the "lbfgs" solver applied and the number of iterations set to 100. KNN model is configured with 5 nearest neighbors, Minkowski distance as the distance metric, and each neighbor contributes equally to the classification. Finally, SVC is based on the radial basis function kernel, with the regularization parameter C set to 0.1 to control the trade-off between margin size and tolerance for misclassification. The "one-vs-one" strategy is used as the solver for this binary fraud detection case.

Regarding the hyperparameters for the models, GridSearchCV was applied as an automated search method to find the optimal combinations. The method systematically tested all combinations and selected the best ones based on cross-validation. For LR, the search included 11 different regularization penalties and a range of C values from 0.001 to 1000. For KNN, various numbers of neighbors were tested through the GridSearch. Finally, for SVC, the search iterated through different values of C and kernels before identifying the optimal combination.

3. Results and Discussion

3.1. Preprocessing effectiveness

To evaluate the impact of each preprocessing step: scaling, stratified sampling, under-sampling, and outlier removal, the changes in data distributions and class balance were first analyzed. The effectiveness of the preprocessing steps was evaluated by examining the changes in data structure and feature distribution. As shown in Fig. 4, the new dataset achieved an almost balanced class distribution after the under-sampling process. This distribution improves upon the original dataset, where the fraudulent class accounted for only 0.18% of all transactions. The balancing process ensures that the training phase is not dominated by the majority class and allows models to learn more informative patterns associated with fraudulent behavior. The scaling of the "Amount" and "Time" features is also important in improving model stability. Fig. 2 shows that the original "Amount" values were highly skewed, with most transactions clustered at low monetary values and a sharp decline after 250 units. Without normalization, such disparities could have biased model training by giving excessive weight to high-value transactions. The use of IQR scaling normalized these values, reducing their influence and helping to maintain consistent feature contributions. Similarly, the distribution of the "Time" feature, as shown in Fig. 3, showed peaks corresponding to specific intervals of transaction activity. While these fluctuations may reflect behavioral patterns, they also represent a risk of temporal bias. With application of IQR scaling to the "Time" feature, the distortion was reduced caused by extreme values and ensured that temporal features did not disproportionately affect classification outcomes. Outlier removal contributed to model stability by filtering extreme deviations in the anonymized features (V1 to V28). Using IQR thresholds, values lying outside of the $1.5 \times$ IQR range were excluded. This step mitigated the impact of noisy or anomalous records, which could otherwise have introduced instability or misleading patterns during training. As a result, the cleaned dataset offered a more consistent and reliable foundation for model development. These preprocessing steps produced an improved dataset in terms of balance, scale, and data quality. This optimized data environment supports the development of more accurate classification models, as discussed in the following subsections.

3.2. Model training and evaluation

The three classification models: LR, KNN, and SVC were developed and trained using the Python programming language and the scikit-learn library. These models were selected due to their interpretability, efficiency, and documented performance in binary classification tasks such as fraud detection. Before training, the dataset was preprocessed using the custom pipeline described in Section 2, which included scaling, stratified sampling, under-sampling, and outlier removal. To ensure fair and reproducible model comparisons, all models were implemented using the same train-test split function, and 80% of the data was used for training while 20% was reserved for testing. GridSearchCV Python function was applied for hyperparameter tuning, using five-fold stratified cross-validation to ensure that class proportions were preserved across all folds. This avoids the risk of overfitting and ensures that the models generalize well across various data subsets. Each model was configured as: The LR model used L2 (Ridge) regularization with the "lbfgs" solver and a maximum of 100 iterations. The KNN classifier was implemented with $k = 5$ neighbors and used the Minkowski distance metric. The SVC used the radial basis function kernel with a regularization parameter $C = 0.1$. The "one-vs-one" classification was used, suitable for binary problems like fraud detection.

The performances of the models are presented in Table 1 in the form of cross-validation accuracy, obtained from GridSearch. LR achieved an accuracy of 94.46%, KNN achieved 93.40%, while SVC attained the highest accuracy of 94.85%.

Table 1

Models' performances

Model	Cross-validation accuracy
LR	94.46%
KNN	93.40%
SVC	94.85%

These results show that the preprocessing methodology had a positive impact on classifier performance. The near-identical performance of LR and SVC further supports the conclusion that the dataset, after noise reduction and balancing, enabled the models to capture meaningful patterns associated with fraudulent activity. The application of GridSearchCV also ensured optimal parameter selection, which contributed to model reliability. By customized model configurations to the structure of the processed data, the training phase benefited from reduced noise, improved class representation, and normalized feature contributions. These conditions supported effective learning, especially in a highly imbalanced context where default model parameters would otherwise lead to suboptimal detection of the minority class.

3.3. Comparative performance of models

In addition to cross-validation accuracy, the models were evaluated using the ROC AUC (Receiver operating characteristic – Area under curve) metric, which provides a more reliable assessment of classification performance in imbalanced datasets. ROC AUC measures a model's ability to distinguish between classes across various threshold settings, where a score of 1.0 indicates perfect separation and 0.5 represents random guessing. As shown in Table 2, the SVC model achieved the highest ROC AUC score of 0.9787, followed closely by Logistic Regression with 0.9769, while KNN achieved a lower but still respectable score of 0.9317. These scores confirm the consistency and reliability of SVC and LR in detecting fraudulent transactions, indicating their improved sensitivity to the minority class. The higher scores of SVC and LR also reflect the benefits of careful preprocessing, including

ISSN-L 2664-9969; E-ISSN 2706-5448

balanced sampling and noise reduction, which allowed the models to better differentiate between fraudulent and legitimate instances.

Table 2

ROC AUC scores

Model	Scores
LR	0.9769
KNN	0.9317
SVC	0.9787

As was the case with the cross-validation analysis, the three models show similar performances from the ROC AUC scores' perspective as well. A similar trend is also visible in Fig. 5, where the LR and SVC graphs more dominantly converge towards the ideal 1.0 in the upper-left corner of the figure and the 1.0 position on the *y*-axis.

As shown in Fig. 5, the ROC curves of all three models show strong classification performance. The SVC model achieves the best overall curve, maintaining a high true positive rate across almost all thresholds, followed closely by logistic regression. KNN performs slightly worse, with a more gradual rise, indicating lower sensitivity to minority class detection. These trends confirm that proper preprocessing improves the ability of classifiers to distinguish between fraudulent and legitimate transactions.

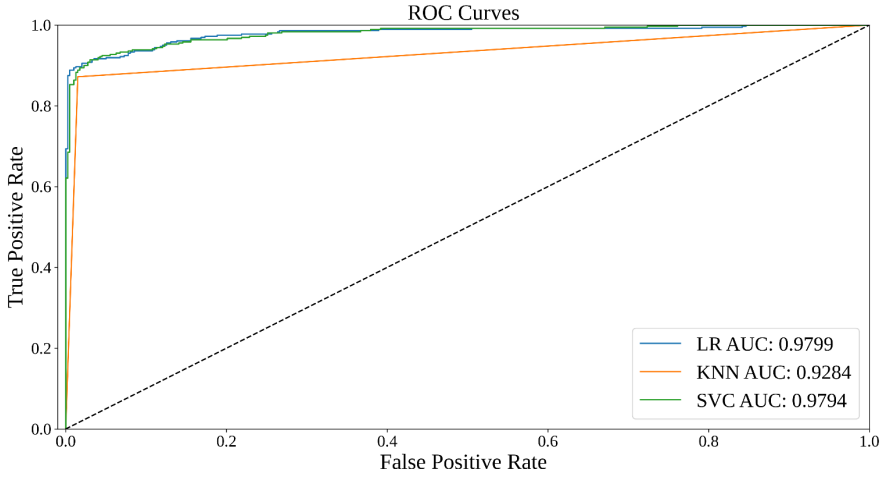

Fig. 5. ROC curves of three classification models

When compared to other studies in the field, the results obtained in this work show competitive and, in some cases, superior performance. For instance, in paper [3] used undersampling and logistic regression to achieve an ROC AUC of approximately 0.95, while our model reached 0.9769, indicating better minority class recognition. Similarly, in paper [7] reported an ROC AUC around 0.97 using SMOTE with CNN, comparable to the 0.9787 ROC AUC achieved by our SVC model, despite using simpler preprocessing and classification techniques. Furthermore, in paper [8] applied SMOTE with an ensemble of CNNs and reported a maximum accuracy of 97.53%, which is in the same range as our SVC accuracy of 94.85%, considering conventional classifiers. These comparisons suggest that the structured preprocessing pipeline presented in this study achieves state-of-the-art performance with reduced computational complexity and implementation cost. Thus, the proposed method provides a practical but powerful alternative to more resource-intensive approaches in the literature.

3.4. Practical implications

The practical significance of this analysis lies in its applicability to real-world fraud detection systems. Financial institutions can implement the proposed preprocessing pipeline to improve the performance of machine learning models in identifying fraudulent transactions, which are underrepresented in historical datasets. By the process of addressing class imbalance, this methodology improves sensitivity to rare fraudulent events, reduces false negatives, and strengthens the reliability of transaction monitoring systems. The results show that a constructed preprocessing pipeline: combining scaling, stratified sampling, under-sampling, and outlier removal, can improve the sensitivity and overall effectiveness of standard machine learning models. In operational terms, this means that institutions can improve their fraud detection capabilities without investing in complex or computationally intensive architecture. The approach relies on open-source tools and interpretable models and, therefore, is accessible to a broad range of users, including those in resource-constrained environments. By improving the ability of models to detect minority class instances, the methodology contributes directly to reducing false negatives, which is important in high-stakes contexts such as credit card fraud detection, where undetected fraud can result in substantial financial losses. Moreover, the pipeline is adaptable. While this analysis focuses on credit card transactions, the same principles can be applied in other domains that face similar imbalanced data challenges, such as insurance claims processing, network intrusion detection, and medical diagnosis. The ease of implementation and the observed model performance also suggest potential for real-time applications, provided that future work incorporates mechanisms for handling streaming data.

3.5. Limitations of research

The research has several limitations. The dataset used covers only a short time frame and originates from a single region, which may affect the generalizability of results to other financial environments. The under-sampling technique, while effective in balancing classes, can lead to information loss, especially when legitimate transaction diversity is important. In addition, the study focused on static datasets, without evaluating real-time or streaming scenarios. Future research should explore the integration of synthetic sampling methods, such as SMOTE or GANs, to complement under-sampling without data loss.

It would also be beneficial to test the proposed approach in real-time environments or extend it to other domains such as insurance claims. Finally, combining preprocessing with ensemble methods or deep learning architectures may further improve scalability.

4. Conclusions

The results show the effectiveness of the proposed preprocessing methodology for handling highly unbalanced datasets in machine learning applications. The analysis applied an approach that included scaling, creating balanced training and test sets, implementing under-sampling, and removing outliers to optimize the dataset for model training. The success of this preprocessing approach was validated through the performance evaluation of three classification models. The evaluation results indicate that the proposed methodology effectively addressed the challenges associated with imbalanced datasets. The cross-validation accuracy scores for LR, KNN, and SVC were 94.46%, 93.40%, and 94.85%, respectively, showing strong predictive performance. The models were evaluated using the ROC-AUC metric, which confirmed their ability to distinguish between fraudulent and

non-fraudulent transactions, with SVC achieving the highest ROC-AUC score of 0.9787, followed closely by LR with 0.9769. These findings are consistent with the improvements observed during preprocessing, particularly the improved class balance and noise reduction that contributed to better learning conditions for all models. The interpretation of ROC curves further supported these results, showing superior sensitivity of the SVC and LR models across various thresholds. This confirmed the suitability of the selected models when trained on a systematically optimized dataset.

These findings suggest that the preprocessing techniques improved the representation of the minority fraud class and enabled the models to generalize effectively. The results can be explained by the application of data preprocessing techniques that improved the quality of input data fed into the models. The scaling process eliminated biases introduced by large numerical differences in feature values. The stratified sampling strategy ensured that the class distributions in both the training and test sets were maintained, preventing the models from being biased toward the majority class. The under-sampling method balanced the dataset and achieved that fraudulent transactions were not overshadowed by the overwhelming majority of legitimate transactions. The removal of outliers contributed to modeling stability by mitigating the effects of extreme values that could distort predictions. From a practical perspective, these results offer insights into financial institutions and other industries dealing with fraud detection, where class imbalance is a common challenge. Implementing these techniques in real fraud detection systems can lead to improved accuracy in identifying fraudulent activities, reducing financial losses, and strengthening the security of financial transactions. From an applied perspective, the methodology offers a scalable and accessible approach for fraud detection tasks, specifically in financial institutions where timely and accurate classification of rare events is important.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Data will be provided upon reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the presented work.

References

1. Tadvi, F., Shinde, S., Patil, D., Dmello, S. (2021). Real time credit card fraud detection. *International Research Journal of Engineering and Technology, 8 (5),* 2177–2180.
2. Ounacer, S., Jihal, H., Bayoude, K., Daif, A., Azzouazi, M. (2022). Handling Imbalanced Datasets in the Case of Credit Card Fraud. *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020).* Cham: Springer International Publishing, 666–678. https://doi.org/10.1007/978-3-030-90633-7_56
3. Pozzolo, A. D., Caelen, O., Johnson, R. A., Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. *2015 IEEE Symposium Series on Computational Intelligence,* 159–166. https://doi.org/10.1109/ssci.2015.33
4. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications, 41 (10),* 4915–4928. https://doi.org/10.1016/j.eswa.2014.02.026
5. Yang, Y., Khorshidi, H. A., Aickelin, U. (2024). A review on over-sampling techniques in classification of multi-class imbalanced datasets: insights for medical problems. *Frontiers in Digital Health, 6.* https://doi.org/10.3389/fdgth.2024.1430245
6. Ileberi, E., Sun, Y., Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data, 9 (1).* https://doi.org/10.1186/s40537-022-00573-8
7. Joloudari, J. H., Marefat, A., Nematollahi, M. A., Oyelere, S. S., Hussain, S. (2023). Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks. *Applied Sciences, 13 (6),* 4006. https://doi.org/10.3390/app13064006
8. Tian, L., Lu, Y. (2021). An Intrusion Detection Model Based on SMOTE and Convolutional Neural Network Ensemble. *Journal of Physics: Conference Series, 1828 (1),* 012024. https://doi.org/10.1088/1742-6596/1828/1/012024
9. Park, J., Kwon, S., Jeong, S.-P. (2023). A study on improving turnover intention forecasting by solving imbalanced data problems: focusing on SMOTE and generative adversarial networks. *Journal of Big Data, 10 (1).* https://doi.org/10.1186/s40537-023-00715-6
10. Liu, Y., Liu, Q. (2025). SMOTE oversampling algorithm based on generative adversarial network. *Cluster Computing, 28 (4).* https://doi.org/10.1007/s10586-024-04980-9
11. Mahmoodi, N., Shirazi, H., Fakhredanesh, M., DadashtabarAhmadi, K. (2024). Automatically weighted focal loss for imbalance learning. *Neural Computing and Applications, 37 (5),* 4035–4052. https://doi.org/10.1007/s00521-024-10323-x
12. *Machine Learning Group.* Machine Learning Group – ULB. Université Libre de Bruxelles. Available at: http://mlg.ulb.ac.be/
13. Bolton, R. J., Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. *Credit scoring and credit control, VII,* 235–255.
14. Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., Bontempi, G. (2018). SCARFF : A scalable framework for streaming credit card fraud detection with spark. *Information Fusion, 41,* 182–194. https://doi.org/10.1016/j.inffus.2017.09.005
15. Lebichot, B., Le Borgne, Y.-A., He-Guelton, L., Oblé, F., Bontempi, G. (2019). Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. *Recent Advances in Big Data and Deep Learning.* Springer International Publishing, 78–88. https://doi.org/10.1007/978-3-030-16841-4_8
16. Abdulhafedh, A. (2022). Comparison between Common Statistical Modeling Techniques Used in Research, Including: Discriminant Analysis vs Logistic Regression, Ridge Regression vs LASSO, and Decision Tree vs Random Forest. *OALib, 9 (2),* 1–19. https://doi.org/10.4236/oalib.1108414
17. Itoo, F., Meenakshi, Singh, S. (2020). Comparison and analysis of logistic regression, Naive Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology, 13 (4),* 1503–1511. https://doi.org/10.1007/s41870-020-00430-y
18. Wang, H., Bell, D. (2004). Extended k-Nearest Neighbours based on Evidence Theory. *The Computer Journal, 47 (6),* 662–672. https://doi.org/10.1093/comjnl/47.6.662
19. Halder, R. K., Uddin, M. N., Uddin, Md. A., Aryal, S., Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data, 11 (1).* https://doi.org/10.1186/s40537-024-00973-y
20. Srisuradetchai, P., Suksrikran, K. (2024). Random kernel k-nearest neighbors regression. *Frontiers in Big Data, 7.* https://doi.org/10.3389/fdata.2024.1402384
21. Hejazi, M., Singh, Y. P. (2013). One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence, 27 (5),* 351–366. https://doi.org/10.1080/08839514.2013.785791
22. Sahin, Y., Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. *Proceedings of the International Multiconference of Engineers and Computer Scientists, 1.*
23. Kumar, S., Gunjan, V. K., Ansari, M. D., Pathak, R. (2022). Credit card fraud detection using support vector machine. *Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2021.* Singapore: Springer, 27–37. https://doi.org/10.1007/978-981-16-6407-6_3

✉ ***Mykola Zlobin,*** *PhD Student, Department of Information and Computer Systems, Chernihiv Polytechnic National University, Chernihiv, Ukraine, e-mail: mykolay.zlobin@gmail.com, ORCID: https://orcid.org/0009-0000-7653-6109*

Volodymyr Bazylevych, *PhD, Associate Professor, Department of Information and Computer Systems, Chernihiv Polytechnic National University, Chernihiv, Ukraine, ORCID: https://orcid.org/0000-0001-8935-446X*

✉ *Corresponding author*