New Weighted Synthetic Oversampling Method for Improving Credit Card Fraud Detection

Sumaya S. Sulaiman[1] , Ibraheem Nadher[1], Sarab M. Hameed[2]
[1] *Department of Computer Science, College of Science, Al-Mustansiriya University, Baghdad, Iraq*
[2] *Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*

Received: 22/4/2024 Accepted: 5/7/2024 Published: 30/**6**/2025

Abstract

The use of credit cards for online purchases has significantly increased in recent years, but it has also led to an increase in fraudulent activities that cost businesses and consumers billions of dollars annually. Detecting fraudulent transactions is crucial for protecting customers and maintaining the financial system's integrity. However, the number of fraudulent transactions is less than legitimate transactions, which can result in a data imbalance that affects classification performance and bias in the model evaluation results. This paper focuses on processing imbalanced data by proposing a new weighted oversampling method, *w*ADASMO, to generate minor-class data (i.e., fraudulent transactions). The proposed method is based on the Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and weight adjustment to identify specific minority areas while retaining data generalization and accurately identifying patterns associated with fraudulent transactions. Experimental results obtained from two datasets with Autoencoder (AE), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) learning models show that wADASMO surpasses other oversampling methods in three evaluation metrics: accuracy at 95.6%, 98.8%, and 99.2%; detection rate at 90.4%, 93.38%, and 93.38%; and area under the curve (AUC) at 93%, 96%, and 96.3% for AE, CNN, and LSTM models, respectively.

Keywords: Autoencoder; convolutional neural network; Credit card fraud detection; deep learning; long short-term memory; resampling

طريقة جديدة لأخذ العينات الاصطناعية الموزونة لتحسين كشف الاحتيال على بطاقات الائتمان

سميه سعد سليمان[1], ابراهيم نضير[1], سراب مجيد حميد[2]
[1]قسم علوم الحاسوب, كلية العلوم, الجامعة المستنصرية, بغداد, العراق
[2]قسم علوم الحاسوب, كلية العلوم, جامعة بغداد, بغداد, العراق

الخلاصة

لقد زاد استعمال بطاقات الائتمان في عمليات الشراء عبر الإنترنت بشكل ملحوظ في السنوات الأخيرة، ولكنه أدى أيضًا إلى زيادة في الأنشطة الاحتيالية التي تكلف الشركات والمستهلكين مليارات الدولارات سنويًا. يعد اكتشاف المعاملات الاحتيالية أمرًا بالغ الأهمية لحماية العملاء والحفاظ على سلامة النظام المالي. ومع ذلك، فإن عدد المعاملات الاحتيالية أقل من المعاملات المشروعة، مما قد يؤدي إلى عدم توازن البيانات ويؤثر

*Email: sumasaad@uomustansiriyah.edu.iq

على أداء التصنيف والتحيز في نتائج تقييم النموذج. تركز هذه الدراسة على معالجة البيانات غير المتوازنة من خلال اقتراح طريقة جديدة لأخذ العينات الزائدة، wADASMO، لتوليد بيانات من الدرجة الثانوية (أي المعاملات الاحتيالية). تعتمد الطريقة المقترحة على تقنية الإفراط في أخذ العينات للأقليات الاصطناعية(SMOTE) ، وأخذ العينات الاصطناعية التكيفية(ADASYN) ، وتعديل الوزن لتحديد مناطق الأقليات المحددة مع الاحتفاظ بتعميم البيانات وتحديد الأنماط المرتبطة بالمعاملات الاحتيالية بدقة. تم الحصول على النتائج التجريبية على مجموعتي بيانات وثلاثة نماذج للتعلم العميق: التشفير التلقائي(AE) ، والشبكة العصبية التلافيفية(CNN) ، والذاكرة الطويلة قصيرة المدى .(LSTM) توضح النتائج أن wADASMO يتفوق على طرق أخذ العينات الأخرى عبر ثلاثة مقاييس تقييم: الدقة ومعدل الكشف والمساحة أسفل المنحنى.

1. Introduction

As our world becomes increasingly digital, cybercrimes, such as credit card fraud, are becoming more common and pose a significant threat to individuals and businesses. Because businesses and financial institutions handle a large number of credit card transactions, fraudsters frequently target the transactions themselves, resulting in significant financial losses for victims. Credit Card Fraud Detection (CCFD) techniques can prevent Internet criminals and ensure customer safety. To implement CCFD effectively, it is crucial to develop an automated approach to handling large volumes of credit card transactions. Due to its flexibility, deep learning (DL) has been used to identify fraudulent credit card transactions. However, the effectiveness of these techniques depends heavily on the quality of the training data, and addressing data imbalances is a significant challenge [1, 2].

The main problem with imbalanced datasets is that the number of fraudulent transactions is much lower than the number of legitimate transactions. This can significantly affect the classification method. When the classification method is used on an imbalanced dataset, the results become biased against the majority class, which can lead to poor generalization of data against minority data (fraud transactions).

There are two primary methods for tackling data imbalances: cost-sensitive and resampling. The cost-based approach involves modifying the learning algorithm to account for the higher misclassification cost of the minority class, which is typically fraudulent. The cost of missed fraud is often associated with the transaction amount, assigning a high misclassification cost to fraud and prioritizing false alerts over missing fraud [3, 4]. This method can result in a large number of false-positives. However, the sampling approach requires balancing the class distributions in the training set before running the learning algorithm. Undersampling involves removing samples from the majority class to balance class proportions, whereas oversampling involves replicating samples from the minority class [5, 6, 7].

Therefore, this paper utilizes the oversampling technique to address the issue of imbalanced classes in credit card fraud detection, thereby safeguarding against fraudulent activities and protecting the financial interests of individuals and businesses. It proposes a model for credit card fraud detection based on a new weighted oversampling method. This paper primarily contributes to the following areas:

- A new weighted oversampling method is proposed to address the overgeneralization issue that may arise when generating synthetic data in a majority-class region with uniform synthetic data effects. The proposed method employs a combination of SMOTE and ADASYN to leverage the benefits of both methods, thereby improving minority class representation and maintaining fraud patterns.
- Enhancing the oversampling technique to boost the efficacy of fraud detection.

The remainder of this paper is organized as follows: Section 2 discusses the various studies on this topic. Section 3 introduces some basic concepts, and Section 4 outlines the proposed method. Section 5 analyzes the results, while Section 6 presents the conclusions.

2. Related Works

Researchers have explored viable solutions to overcome the imbalanced dataset problem in credit card theft. In general, sampling approaches are the preferred alternatives. The following are state-of-the-art studies that employ sampling approaches to solve this issue.

Lin et al. [8] applied the Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and Tomek Link (T-Link) to the European cardholders' dataset to balance the numbers of legitimate and fraudulent transactions in order to improve the proposed AutoEncoder with Probabilistic Random Forest (AE-PRF) performance. The experimental results showed that the performance of the AE-PRF was consistent regardless of whether resampling schemes were applied. The results were compared in terms of accuracy, true positive rate (PR), true negative rate (TNR), Matthews correlation coefficient (MCC), and area under the receiver operating characteristic curve (AUC-ROC) of 99%, 81%, 99%, 84%, and 96%, respectively. However, the description of the superiority of the proposed method over popular techniques is insufficient, and there is a lack of comparative analysis with current state-of-the-art technologies.

SMOTE and ADASYN resampling approaches have been used to handle an imbalanced European cardholder dataset [9]. Different machine learning (ML) algorithms, namely Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree (DT), and Logistic Regression (LR), were then applied to the balanced dataset to evaluate their performance. The findings showed that resampling the dataset led to better classification of fraudulent activities using the above ML algorithms with an accuracy of over 99%. The LR algorithm, on the other hand, had accuracies of about 94.51% and 88.96% based on SMOTE and ADASYN, respectively. However, it has been discovered that the uneven class distribution in credit application data streams presents difficulties that cannot be solved using typical ML techniques such as RF, KNN, and LR.

The SMOTE oversampling technique was adopted in [10] before implementing a Long Short-Term Memory (LSTM)-based model incorporating an attention mechanism. This model combines three key components: UMAP for feature extraction, LSTM to assimilate transaction sequences, and attention to enhance the effectiveness of the LSTM. By pinpointing crucial transactions within the input sequence, the classifier achieved significantly improved fraud accuracy, reaching 97% and 96% when synthetic bank-SIM and European datasets were used, respectively.

The imbalance issue in [11] was addressed by applying SMOTE to a hybrid Convolutional Neural Network-Support Vector Machine (CNN-SVM) model, which involved modifying the CNN output layer with an SVM classifier. The proposed model is trained end-to-end with a fully connected softmax layer. The hybrid model's efficacy was assessed on a publicly available real credit card transaction dataset and revealed impressive classification performance metrics: accuracy, precision, detection rate, F1-score, and AUC, all exceeding 90%. Nevertheless, employing SVM poses limitations, particularly in handling imbalanced datasets or data with high-dimensional noise.

The K-CGAN model was introduced in [1] to address the imbalanced data issue in the European cardholder dataset. The model employs a customized generator loss function of

Kilberg divergence, as well as a conditional generative adversarial network (GAN). Several classification approaches were used to evaluate the efficacy of different data sampling techniques. The results show that Borderline-SMOTE (B-SMOTE), K-CGAN, and SMOTE have the highest precision (99%) among the sampling techniques. Additionally, the highest F1 Score was acquired by the proposed K-CGAN model with 99% and an accuracy of 99%.
A new method based on analyzing credit card transaction patterns in past purchases was proposed to improve fraud detection in the CCFD system [12]. To address the problem of imbalanced data in credit card fraud detection, the Synthetic Minority Oversampling Technique Edited Nearest Neighbor (SMOTE-ENN) method was used. Moreover, the LSTM-based sequential model was applied to capture the behavior of credit card holders' historical purchases. The proposed model's results achieved a high precision of 99.7% for detecting credit card fraud.

In [13], several machine learning techniques, including DT, RF, KNN, LR, and Naive Bayes (NB), were used for classification purposes. The resampling techniques, namely Random Under-Sampling (RUS), SMOTE, and ADASYN, were used to mitigate the problem of the European credit card fraud dataset imbalance class. The best results were achieved by the LR model (99.94%) when RUS was used. In addition, when oversampling was applied, the RF model achieved 99.964%.

In [14], the features extraction method, including principal component analysis (PCA) and convolutional autoencoder (CAE) methods, was used, and RUS, SMOTE, and SMOTE-Tomek techniques were applied to mitigate class imbalance. To distinguish fraudulent credit card transactions from legitimate ones, four ensemble classifiers—RF, CatBoost, LightGBM, and XGBoost—were used. The best results were obtained in the F1-score (95.4%) in the case of applying RUS, followed by CAE.

In [15], the researchers introduced the new Navo Minority Over-Sampling Technique (NMOTe) to address the imbalance problem in three datasets: European, University of California San Diego-FairIsaac (UCSD-FICO), and Small Credit Dataset (SCD), using the Convolutional Neural Network-Gated Recurrent Unit (CNN-GRU) classifier. The experiments were performed using 80%–20% and 60%–40% of the dataset. The proposed NMOTe sampling model achieved 98.45% accuracy, whereas the existing SMOTE achieved 94.19% accuracy, which shows that the imbalance problem is effectively solved using the NMOTe sampling technique.

In [16], they proposed a resampling algorithm based on GAN and B-SMOTE as a balancing module to synthesize better minority samples and enhance the performance of the classification model. Experiments were conducted on two real-world credit card datasets (the University of California (UCI) dataset and a private dataset), and the Multi-Factor Generative Adversarial Network (MFGAN) model achieved a higher AUC of 77.28% and 91.08% and a detection rate of 65.06% and 82.30%, respectively, without reducing the F1 score by 53.98% and 47.06%, respectively, which proved that the MFGAN model was feasible and effective.

In [17], AutoEncoder (AE), CNN, and LSTM deep learning models were applied, and hyperparameter tuning techniques, including random and Bayesian methods, were employed to optimize the model structures with RUS, SMOTE, and ADASYN sampling methods. The European credit card fraud dataset was used as an evaluation dataset, and the results showed that AE, CNN, and LSTM models with ADASYN surpassed other methods. The best results achieved are accuracy exceeding 95%, DR over 90%, and AUC exceeding 92%.

Several methods have been used in previous studies to address the problem of class imbalance in credit card datasets. Nevertheless, there remains a requirement to minimize overfitting and generate fraudulent transactions that simulate fraudulent activity to address class imbalance problems in credit card datasets. To achieve this, a hybrid oversampling method is proposed to balance the datasets and improve the DL classifier's ability to detect fraudulent transactions.

3. Preliminary Concepts

This section describes the basic concepts used in this paper.

3.1 Sampling Methods

Sampling methods are used to address imbalanced dataset distributions, either by reducing the samples of the majority class (undersampling) or by increasing the samples of the minority class (oversampling). In the case of oversampling techniques, new samples are generated that may look similar to the original data, but these replicates may not be the same. The two common oversampling methods are as follows:

1. SMOTE is a data sampling technique used to increase the representation of minority classes in datasets. This is achieved by generating new synthetic components of the minority class based on those that already exist and are close to each other. It works by drawing a line between the minority class data samples, then creating a new data sample at a point on the line. Thus, SMOTE selects data samples that are close together in the minority class. SMOTE is an effective way to address the overfitting problem caused by random oversampling. It works particularly well for small-sized datasets, although it can be slower for larger datasets. However, there is a chance that some data points for the minority class will overlap in SMOTE. This could weaken the boundary and make it more likely that the boundary samples will be misclassified [1, 14, 18].
2. ADASYN is a technique used to address the imbalance problem by increasing the minority class representation in the datasets. This creates minority data samples that reflect the distributions of underrepresented groups to generate more data. ADASYN calculates the level of imbalance for each minority instance by looking at the number of majority instances among its k nearest neighbors. It then determines the density ratio for each minority instance and generates synthetic samples based on this ratio. The number of synthetic samples produced for each minority instance is determined adaptively by ADASYN. This method can be used to generate data samples for minority-class samples that are difficult to learn. ADASYN's generated data points not only balance the dataset well, but they also reduce the learning bias of the actual dataset. However, this algorithm may suffer from reduced precision due to its adaptability. In addition, the neighborhoods created by ADASYN contain only one minority example for minority samples that is sparsely distributed [1, 19].

3.2 Deep Learning Methods

Deep learning, a form of neural network, tackles complex issues through modeling. This paper explored three DL models: AE, CNN, and LSTM.

AE operates as an unsupervised learning method, employing backpropagation by aligning the inputs and outputs to be the same [20, 21]. The primary objective of AE is to approximate the input value's distribution accurately [22]. Through automatic decoding and encoding of input data during training, the AE effectively extracts crucial features. When a transaction deviates notably from what the model recognizes as "legitimate," it incurs a higher reconstruction error. This capability enables the identification of anomalies in the data by learning the model to reconstruct the input data [23, 24, 25].

CNNs, a particular kind of artificial neural network (ANN), possess a distinct structure, enabling them to extract more complex information from data at every layer, thereby

contributing to the ultimate decision. CNN comprises multiple layers (including convolutional and fully connected layers), pooling, shared weights, and local connections. It operates as a feedforward neural network [11, 26, 27]. The CNN reduces human experience interference with the model by learning derivative features that benefit the classification results [28, 29]. Convolutional layers are a useful tool that CNN may use to extract significant local information from transactions that might be signs of fraud. Furthermore, by sharing parameters, it reduces the number of parameters that need to be trained, increasing CNN's computational efficiency and ability to handle large-scale credit card transaction datasets. In addition, it effectively suppresses noise and focuses on the most discriminative features by utilizing convolutional and pooling operations [11, 29, 30].

LSTM, a specialized form of recurrent neural network (RNN), excels at capturing time series data and detecting extended sequential patterns. Unlike feedforward neural networks, LSTM incorporates feedback connections among its hidden units, which are tailored for handling time-series data. This unique design lets LSTM find long-term sequential patterns, which helps predict transaction labels by looking at the order of past transactions [10, 30, 31].

4. Proposed Credit Card Fraud Detection

The majority of legitimate transactions can pose difficulties for DL models in accurately identifying fraudulent instances owing to an imbalance in the data. Falsely categorizing a legitimate transaction as fraudulent can easily be resolved by verifying it with the customer. However, misclassifying fraudulent transactions as legitimate can lead to significant financial harm to banks. To prevent this, a new method of oversampling is proposed that generates additional fraudulent data to balance the predominance of non-fraudulent cases within the dataset.

Consider dataset C, described as $\mathbb{C} = \{C_1, C_2 \ldots, C_n\}$, that contains credit card transactions, each of which has m features. Each transaction is labeled legitimate (0) or fraudulent (1), $L = \{0,1\}$. The number of transactions in the minority class (fraudulent) and the majority class (legitimate) can be denoted as n_f and n_l, respectively, with $n = n_f + n_l$. The objective of the proposed method is to generate more fraudulent transactions to increase the number of fraudulent transactions, n_f, thereby effectively detecting fraudulent transactions and minimizing false negatives.

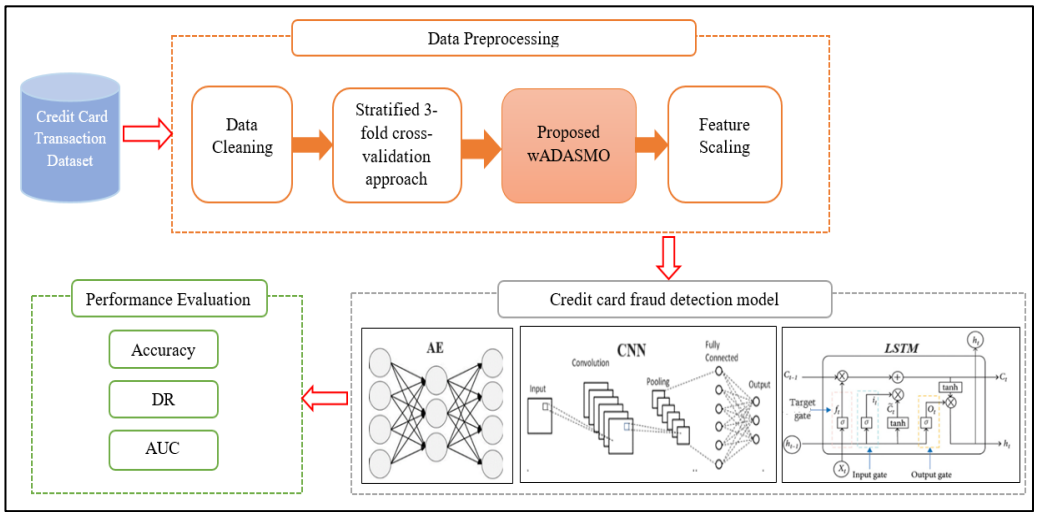

Figure 1: The general layout of the proposed credit card fraud detection mode

The proposed model involves two main stages: data preprocessing and the detection of credit card fraud. Data preprocessing is a critical step in credit card fraud detection, and this paper focuses on four critical preprocessing procedures, namely: data cleaning, stratified k-fold approach, oversampling method, and feature scaling, that are necessary for preparing data in a suitable format for training and evaluation. After preprocessing, the credit card data is fed into AE, CNN, or LSTM deep learning models to detect credit card fraud by classifying credit card transactions as legitimate or fraudulent. Figure 1 shows the block diagram of the proposed model. The subsequent subsections present the specifics of each step:

4.1 Data Cleaning

The proper handling of missing values in a dataset is critical to ensuring that the performance of the model is not negatively affected. Missing values can be removed or imputed; however, it is crucial to experimentally determine a threshold. If the percentage of missing values is below the threshold, imputation can be performed by using the mean value of the missing feature. However, removing transactions with missing feature values is a better approach for missing values that exceed the threshold.

4.2 Stratified K-fold Approach

The stratified k-fold approach is used for splitting the credit card dataset into training and test sets. The reason for choosing this approach is to confirm a uniform distribution of legitimate and fraudulent transactions in each fold. The dataset is shuffled such that each fold represents a miniature of the entire dataset by keeping the class distribution in each fold, and the model is exposed to a representative sample of each class throughout training and testing, as shown in algorithm 1. The reason for using this approach in this paper is that we are dealing with minority class datasets in which the fraud class has much fewer occurrences than the legitimate class (see Figure. 2).

Let $(C_{f1}, L_{f1}), (C_{f2}, L_{f2})$, and (C_{f3}, L_{f3}) represent credit card transactions and their corresponding labels belong to fold f_i, $i \in \{1,2,3\}$. During each iteration, one fold is used as a test set, C_{tst} while the other two are used as the training set, C_{trn}. Stratified k-fold cross-validation ensures that each fold retains the class distribution of the original dataset (That is, each fold contains a similar percentage of samples from both the minority and majority classes as the original dataset). The stratification condition is shown in Eq. (1).

$$i \in \{1,2,3\}, \forall\, l \in L: \frac{count(L_{fi,l})}{count(L_{fi})} \approx \frac{count(L_l)}{n} \quad (1)$$

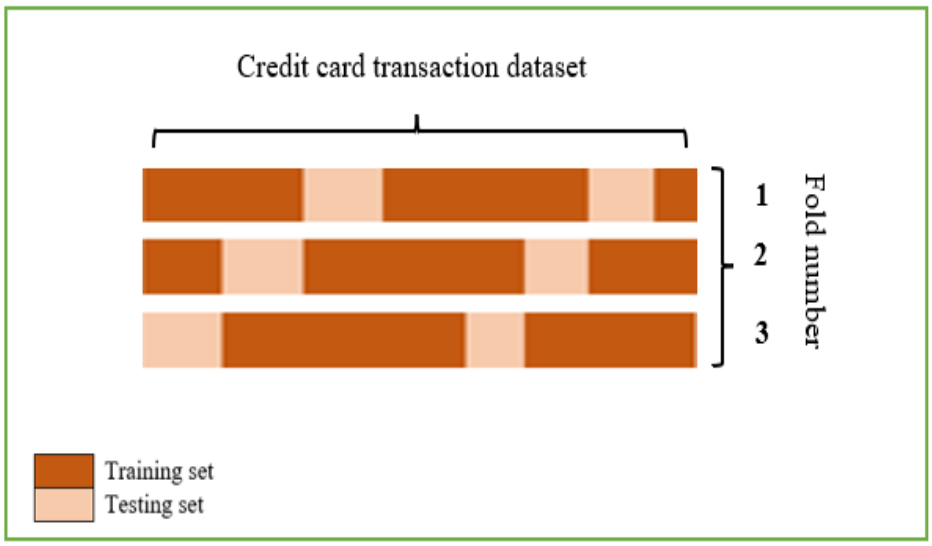

Figure 2: Stratified 3-fold cross-validation

Algorithm 1: Fold Generation with Stratified 3-fold CV

Input:

- Credit card transaction dataset: $\mathbb{C} = \{C_1, C_2 \ldots, C_n\}$
- Number of classes: c
- Label of classes: L= $\{L_0,\ L_1\ \}$
- Number of folds K=*3*

Output:

- Generated folds $f_1,\ f_2,\ f_3$

1. Set $f_1 = \emptyset,\ f_2 = \emptyset,\ f_3 = \emptyset$

2. $For\ i = 1\ to\ c$ // two classes legitimate and fraudulent
Count the number of credit card transactions in class L_i
$t = Count\ (L_i)/K$

3. If $(Count\ (L_i)\ mod\ K) > 0$ then $t = t + 1$

4. $For\ j = 1\ to\ K$ // number of folds is 3
Set $R = \emptyset$

5. Select randomly t credit card transactions from $\mathbb{C}$ with class label L_i and add them to set R

6. Add the selected credit card transactions to f_j
$f_j = f_j\ U\ R$

7. Remove R credit card transactions from $\mathbb{C}$

8. End for

9. End for

4.3 The Proposed Oversampling Method

Class imbalances often affect credit card fraud datasets. To address this issue, a new sampling method is proposed that effectively balances the number of legitimate and fraudulent transactions. The proposed oversampling method, coined as (*w*ADASMO), is a weighted combination of ADASYN and SMOTE, as shown in Figure 3. This method adopts equal weights for both ADASYN and SMOTE to generate synthetic for minority-class transactions that mimic the patterns and characteristics of fraudulent transactions by generating new data

points informed by existing minority-class data, resulting in a more robust and accurate fraud detection model.

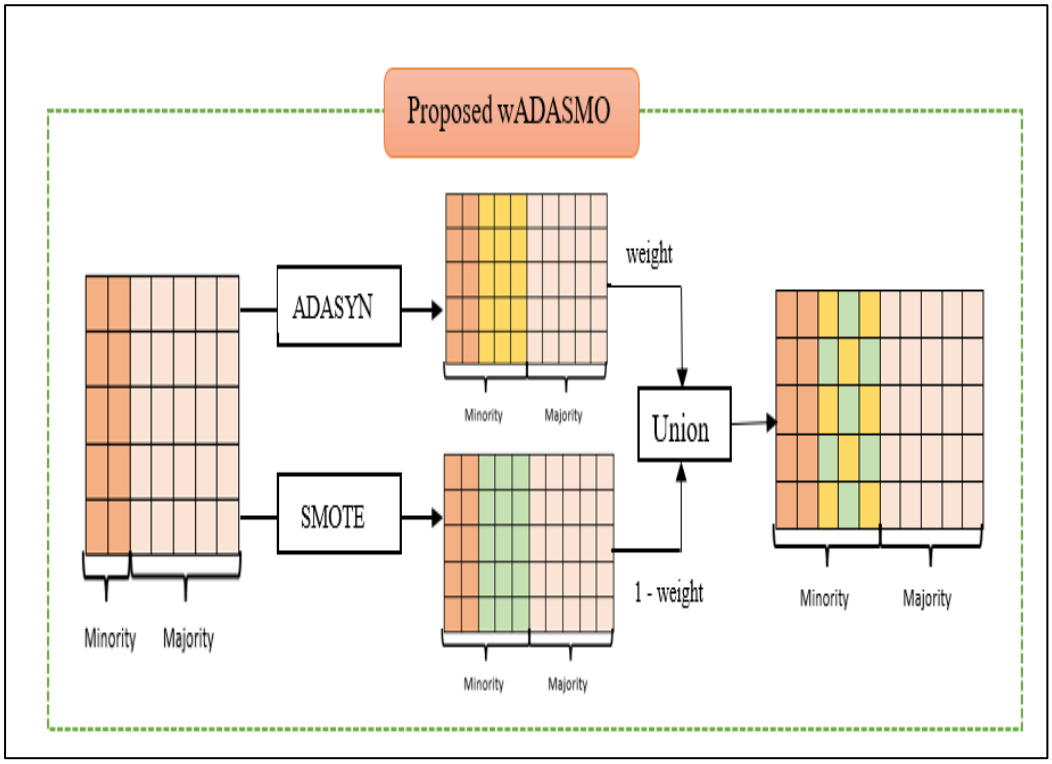

Figure 3: Proposed *w*ADASMO general layout

Algorithm 2 presents the details of wADASMO, which is based on the weighted samples of SMOTE and ADASYN. It takes a training dataset, C_{trn}, and split it into a fraud set, $C_{fraud} = \{frd_1, frd_2, \dots, frd_{n_f}\}$ and legitimate set, $C_{leg} = \{leg_1, leg_2, \dots, leg_{n_l}\}$ Then, the fraud set, C_{fraud}, is oversampled using SMOTE and ADASYN. The Euclidean distance between two credit card transactions is calculated using Eq. (2)

$$\forall\, fr_i \in C_{fraud}, \forall\, c_j \in C_{trn},\ diff = \sqrt{(fr_i - c_j)^2} \quad (2)$$

For SMOTE, the number of synthetic transactions per_{smote} is determined based on Eq. (3), considering the desired percentage of the total legitimate and fraud transactions:

$$per_{smote} = (n_l - n_f) * 100/n_f \quad (3)$$

Subsequently, some fraud transactions are randomly selected from C_{fraud} according to n_{smote}. Then, new synthetic transactions, $SMOTE_{new}$ generated along the line between the minority example and its selected nearest neighbors k_{near}; and these samples are added to the $SMOTE_{set}$

The ADASYN oversampling method functions similarly to SMOTE but adjusts the creation of synthetic transactions based on the density distribution of the minority class, C_{fraud}. ADASYN focuses on generating samples for instances that are more challenging to classify by introducing varying levels of imbalance. The number of synthetic transactions is computed is shown in Eq. (4).

$$n_{adasyn} = \left(n_l - n_f\right)\beta \quad (4)$$

where $\beta \in [0,1]$ is used to determine the intended balance level after generating the synthetic data. For each transaction, frd_i within C_{fraud} ADASYN calculates its nearest neighbors. Then, it computes the difference in the number of neighboring majority class instances to the current transactions, generating a density distribution ratio, as in Eq. (5):

$$d_{adasyn} = \frac{n_l}{n_f} \in (0,1] \quad (5)$$

This ratio guides the creation of synthetic examples, with a higher emphasis on transactions in regions with a greater class imbalance.
By leveraging this approach, ADASYN dynamically adapts the generation of synthetic samples, placing more emphasis on instances that are more challenging to classify accurately because of their proximity to the majority class. These newly created transactions are appended to $ADASYN_{set}$.

Finally, a weighted fraud transaction from $ADASYN_{set}$ and $SMOTE_{set}$ are used to generate synthetic fraud transactions set, $wADASMO_{set}$ as shown in Eq. (6). In this way, the proposed method enhances the transaction selection process by using both ADASYN and SMOTE, where ADASYN generates new samples in areas with low density of minority class, and SMOTE generates synthetic instances by interpolating among existing minority class data points. By leveraging the power of both SMOTE and ADASYN methods, we can achieve a comprehensive representation of the minority class distribution. This not only helps to prevent overfitting but also provides a diverse range of data points for the model to learn from.

$$wADASMO_{set} = w \times ADASYN_{set} \cup (1-w) \times SMOTE_{set} \quad (6)$$

Where $w \in (0,1)$ is a fraud transaction weight

Algorithm 2: The proposed *w*ADASMO

Input:

- Training dataset C_{trn},
- Number of nearest neighbors k =5

Output:

- New training dataset with synthesized fraud transactions, C_{trn_new}

1. Divide C_{trn}, into two sets: fraud, C_{fraud}, and legitimate, C_{leg}
Calculate the percentage of new synthetic transactions per_{smote} using equation 3
2. $For\ i = 1\ to\ n_f$
Choose a random transaction from C_{fraud} and determine its k nearest neighbors according to Euclidean distance using equation 2;
$n_{smote} = per_{smote}/100$
3. While (n_{smote}!=0)
4. Select the k_{near} neighbor transaction from k.
5. Generate the new synthetic transaction $SMOTE_{new}$ and add it to $SMOTE_{set}$
6. $n_{smote} = n_{smote} - 1$
7. End while
8. End for

9. Calculate the imbalance degree d_{adasyn} depending on the transactions number in C_{fraud} and C_{leg} using equation 5
10. Determine imbalance threshold th ;
11. if $d_{adasyn} < th$ go to step 5
12. Choose a random transaction from C_{fraud} and determine its k nearest neighbors according to the Euclidean distance in equation 2
13. Calculate the ratio $r_i,\ i = 1..n_f$, of the chosen transaction depending on n_l , n_f
14. Calculate the density distribution r of the chosen transaction based on $r = r_i / \sum_i r_i$, and $r_i \in [0, 1]$;
15. Determine the amount of synthetic transactions that are generated for each chosen minority transaction $g_i = r * n_{adasyn}$;
16. $For\ i = 1\ to\ g_i$
17. Select the k_{near} neighbor transaction from k.
18. Generate the new synthetic transaction c_{ADASYN_new} and add it to $ADASYN_{set}$
19. End for
20. Generate synthetic fraud transactions set, $wADASMO_{set}$ using equation 6
21. Concatenate $wADASMO_{set}$ and C_{leg} to generate a new balanced training set, C_{trn_new}
$C_{trn\_new} = wADASMO_{set} \cup C_{leg}$

4.4 Feature Scaling

Feature scaling is a critical step in the preprocessing of numerical features. It ensures faster convergence of the model's optimization algorithm and prevents numerical instability. The standardization method of z-score scaling, as in Eq. (7), is used to standardize numerical features with a mean of 0 and a standard deviation of 1.

$$z = (x - \mu)/\sigma \tag{7}$$

where μ is the feature mean, σ is the feature standard deviation, and x is a single raw data value.

4.5 Detection of Credit Card Fraud

The proposed DL model, as outlined in algorithm 3, is accomplished through the integration of wADASMO with three DL models. First, the proposed AE model comprises an encoding layer E and a decoding layer D. The encoding layer E transforms the input C into a hidden vector C' to capture essential features, which is then used by the decoding layer D to reconstruct C" from C'. The model evaluates the difference between the original input C and the reconstructed output C" to calculate the mean squared error (MSE). This MSE is then compared to a threshold, α. If the MSE exceeds α, the system predicts that the credit card transaction is fraudulent; otherwise, it classifies it as normal. The choice of the threshold α significantly impacts the AE model's performance, and several studies in the existing literature focus on determining an optimal α. However, the threshold value in this paper is determined by calculating the percentile as expressed in Eq. (8).

$$Percentile = (V_x/n) \times 10 \tag{8}$$

where V_x: is the number of data values less than x, and n is the overall credit card data transactions in the dataset.

CNN is the second deep learning model proposed in this paper. The architecture of CNN includes two convolutional layers, two pooling layers, one flattened layer, one dropout layer, and two dense layers designed to process one-dimensional data. The number of time steps is added as a third dimension to sequential data, transforming them into a three-dimensional vector that resembles an image. In the convolutional layers, size-three kernels are employed to multiply inputs and create a new feature map. A pooling layer reduces the number of trainable parameters. The flattened layer then transforms the three-dimensional vector into a one-dimensional input for the dense layers.

Finally, the proposed LSTM model includes a single LSTM layer to capture long-range dependencies in sequences and one dense layer. The sequential credit card dataset C is prepared to be processed by the LSTM layer, which involves transforming the data into a three-dimensional form of sequences that represent a time series or a sequence of events.

Algorithm 3: the proposed credit card detection using DL with *w*ADASMO

Input:

- $\mathbb{C} = \{C_1, C_2 \ldots, C_n\}$: Credit card transaction dataset

Output:

- Trained Model

1. Process missing $\mathbb{C}$ values by employing the imputation method
2. Use stratified k-fold cross-validation to split the dataset, $\mathbb{C}$ into training and testing sets (C_{trn}, C_{tst})
3. Oversample the training dataset $\mathbb{C}_{trn}$ using *w*ADASMO to generate a new training set C_{trn_new}
4. Normalize credit card transaction dataset, C_{trn_new} using equation (7).
5. Build and compile the DL model.
6. Apply the chosen hyperparameters to train the model on the training data, C_{trn_new}
7. Assess the performance of the trained model on the test data, $\mathbb{C}_{tst}$ in terms of Acc, DR, and AUC.

5. Experiments and Result Analysis

The proposed wADASMO method is compared with the ADASYN and SMOTE methods to evaluate its effectiveness, which serves as a baseline model for credit fraud, and the experiments are conducted on two benchmark credit card fraud datasets: European credit cards and simulated credit cards. Table 1 provides a detailed description of the two benchmark credit card fraud datasets.

- The European credit card dataset [32] is used for evaluating the performance of models designed for detecting fraudulent credit card transactions. This dataset is made up of 284,807 transactions generated by European cardholders over two days in September 2013. Each transaction is composed of thirty-one features, including the amount of the transaction, the time of the transaction, and the type of card used, among others. Additionally, each transaction is associated with a class label that indicates whether it is fraudulent (1) or not. However, the dataset is highly imbalanced, with only 492 fraudulent transactions, which constitute 0.172% of the total data.
- The Sparkov data generation tool developed by Brandon Harris was utilized to create a synthetic dataset of credit card transactions [33]. The simulation was carried out over two years, from January 1, 2019, to December 31, 2020. The dataset contains records of 1,842,743 legitimate transactions and 9,651 instances of fraudulent transactions. 1,000 customers and 800 merchants conducted the transactions.

Also, to prevent bias towards any particular model, hyperparameter optimization was conducted, and the best setting was selected for each model in the testing, as shown in Table 2.

Table 1: Dataset Description

Dataset	Number of transactions	Features	Legitimate transactions	Fraudulent transactions	Fraud Ratio
European credit card dataset	284,807	31	284,3150.5210	492	0.172%
Simulated credit card dataset	1,852,394	23	1,842,743	9,651	0.521%

Table 2: The proposed DL models with *w*ADASMO optimal hyperparameter values

DL	Hyperparameter Values						
	Number of neurons per layer	Batch size	Optimization Function	Activation function	Dropout	Loss function	Learning rate
AE	512	64	Adam	Tanh	not needed	MSE	0.001
CNN	32	128	Adam	sigmoid	0.2	BFL	0.0001
LSTM	128	32	Adam	sigmoid	not needed	BFL	0.0001

5.1 Evaluation metrics

The evaluation metrics used to assess the credit card fraud detection models consider the class imbalance and the importance of correctly identifying both fraudulent and legitimate transactions.

The performance of the proposed model is evaluated using accuracy (Acc), detection rate (DR), and area under the curve (AUC). The accuracy measure measures the model's overall performance, as shown in Eq. (9), whereas the detection rate metric measures the model's ability to detect fraudulent transactions correctly, as shown in Eq. (10). AUC measures the overall performance of the model in terms of how well it can discriminate between fraudulent and non-fraudulent transactions, as shown in Eq. (11) [34].

$$Acc = \frac{TP+TN}{TP+FN+TN+FP} \tag{9}$$

$$DR = \frac{TP}{TP+FN} \tag{10}$$

$$AUC = \frac{1+\frac{TP}{TP+FN}-\frac{FP}{TN+FP}}{2} \tag{11}$$

where

True Negative (TN) is the number of legitimate transactions that are correctly classified as legitimate; True Positive (TP) is the number of fraudulent transactions that are correctly classified as fraud; False Positive (FP) is the number of legitimate transactions incorrectly classified as fraud; and False Negative (FN) is the number of fraud transactions incorrectly classified as legitimate transactions.

5.2 Impact of *w*ADASMO on Dataset Distribution

To visualize the data distribution of the high-dimensional European and simulated credit card datasets, principal component analysis (PCA) was used to reduce the dimensionality of the datasets to two. This allows for a deeper understanding of the underlying patterns and structures in the data. It is clear from the illustration in Figs. 4 and 5 that there is an uneven distribution of credit card fraud transactions in the European and simulated credit card datasets. The primary aim of SMOTE and ADASYN is to balance the class distribution in such datasets by generating synthetic transactions for the fraudulent minority class. SMOTE can potentially make the minority class distribution denser or introduce patterns that were not present in the original data. SMOTE balances the class distribution but may also introduce synthetic data, leading to overfitting and boundary issues.

Figures 4 (b) and 5 (b) indicate that synthetic samples in SMOTE are generated in the feature space of the minority class. Their positions are determined by the distribution and density of the minority class instances, rather than aiming to target areas of complexity or challenge for the classifier. This approach sometimes leads to oversampling in noisy regions or areas that are not necessarily more difficult to classify. ADASYN, on the other hand, creates artificial samples that change based on the local density of minority and majority classes. It focuses on creating samples for minority cases that are harder to classify, which makes the minority class distribution denser. This adaptiveness makes ADASYN potentially more effective in handling complex, imbalanced datasets. Figures 4(c) and 5(c) show that the ADASYN algorithm gives more importance to the areas where the classifier is likely to make errors due to class imbalance. This means that ADASYN focuses on generating synthetic samples in regions that are challenging for the classifier to classify accurately. Putting ADASYN and SMOTE together in a weighted way, as shown in Figures 4(d) and 5(d), changes how minority class instances are chosen for making synthetic samples. This approach assigns higher importance to instances that are critical in addressing the imbalance or more challenging for the classifier. This adaptation aims to improve the effectiveness of oversampling by generating synthetic samples in areas of the feature space that are more difficult to classify accurately, resulting in better generalization and classification in imbalanced datasets.

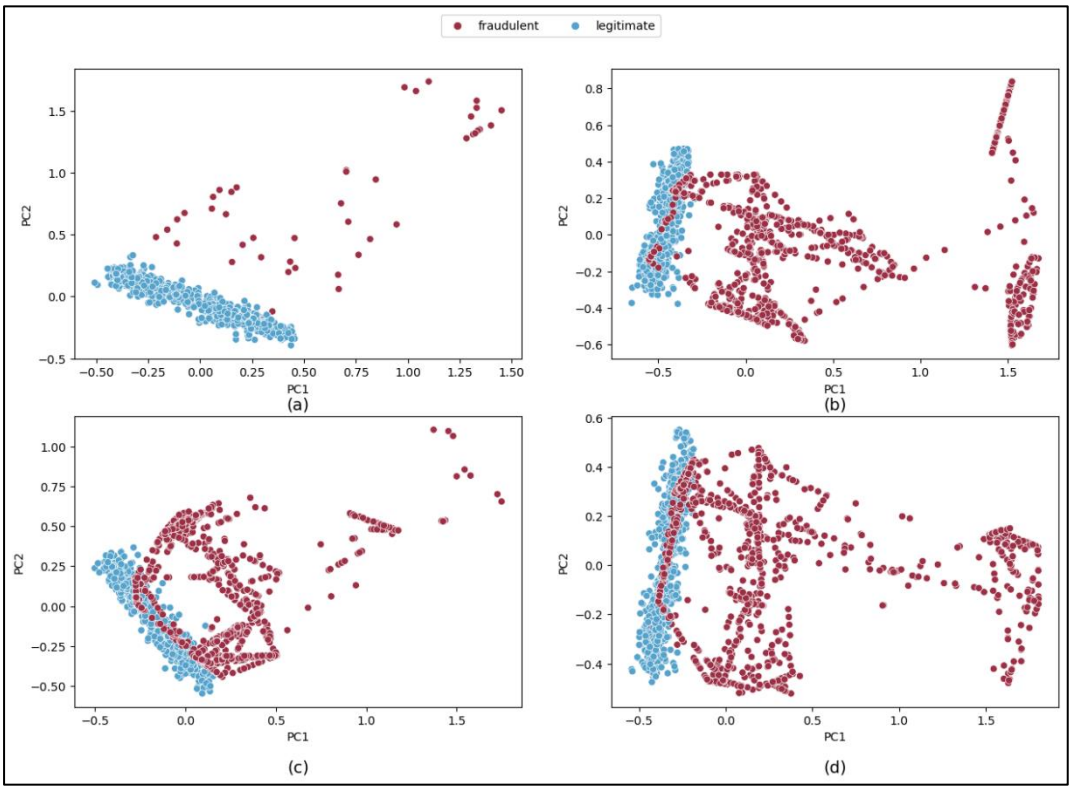

Figure 4: The distribution of European dataset with the red is fraud class and the blue is legitimate class. (a) Original European cardholder's dataset, (b) European cardholder's dataset after SMOTE, (c) European cardholder's dataset after ADASYN, and (d) European cardholder's dataset after *w*ADASMO

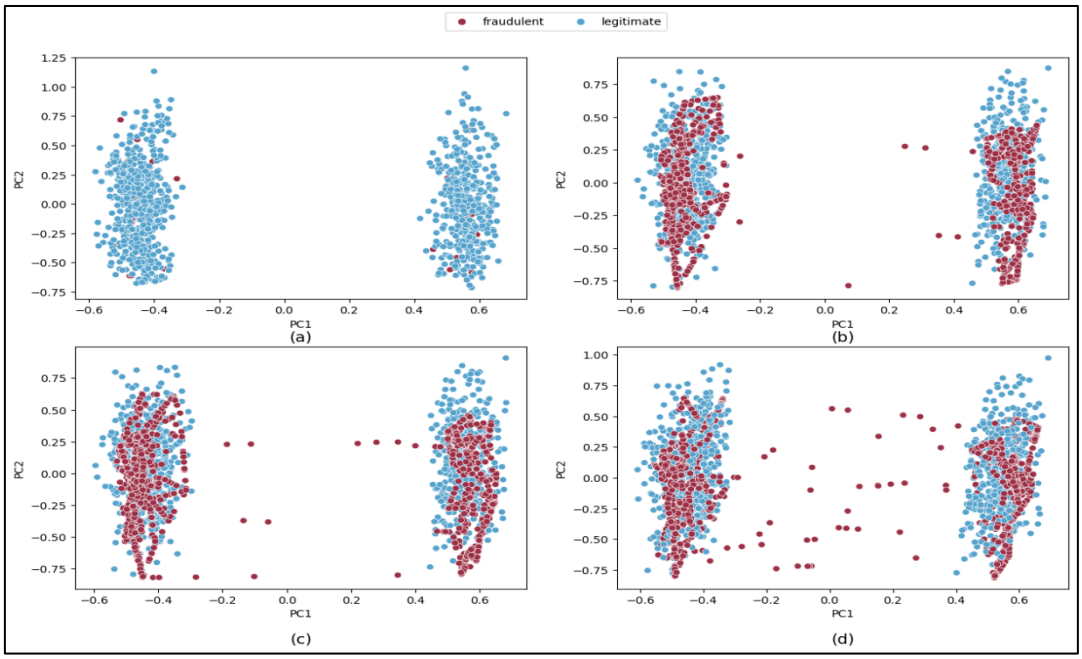

Figure 5: The distribution of the simulated dataset with the red as fraud class and the blue as legitimate class. (a) Original simulated credit card dataset, (b) simulated credit card dataset after SMOTE, (c) simulated credit card dataset after ADASYN, and (d) simulated credit card dataset after *w*ADASMO method

5.3 Impact of Weights on the Proposed Method

The impact of different weight assignments on wADASMO was investigated to provide insight into the optimal weight. The performance in terms of accuracy, DR, and AUC results with different weights w ={0.1,0.2,...,0.9} of wADASMO with three DL models AE, CNN, and LSTM is analyzed on both the European cardholder's dataset and the simulated credit card dataset, as shown in Figures 6 and 7.

The results indicated that the proposed wADASMO produced varying results depending on the weight value. In minority-class cases, the weight value plays a critical role in maintaining diversity and adaptability. When the weight value is less than 0.5, it emphasizes general oversampling techniques, while a weight value greater than 0.5 focuses on adapting to complex minority class instances. However, setting a weight value of 0.5 creates a balance between diversity and adaptability in generating synthetic samples, resulting in improved accuracy, DR, and AUC. Consequently, the weight w is set to 0.5, which effectively captures the complexities present in the minority class while preventing overfitting. It is essential to balance oversampling techniques while generating synthetic samples to achieve the best possible outcomes and avoid overfitting. This balance is critical because it ensures that the results do not favor the majority class at the expense of the minority class.

When compared to CNN and LSTM, AE's performance differed slightly with different weights. This is because AE's goal is to reduce the reconstruction error by capturing important features of data in a compressed form, which reduces the reliance on individual instances or variations in samples.

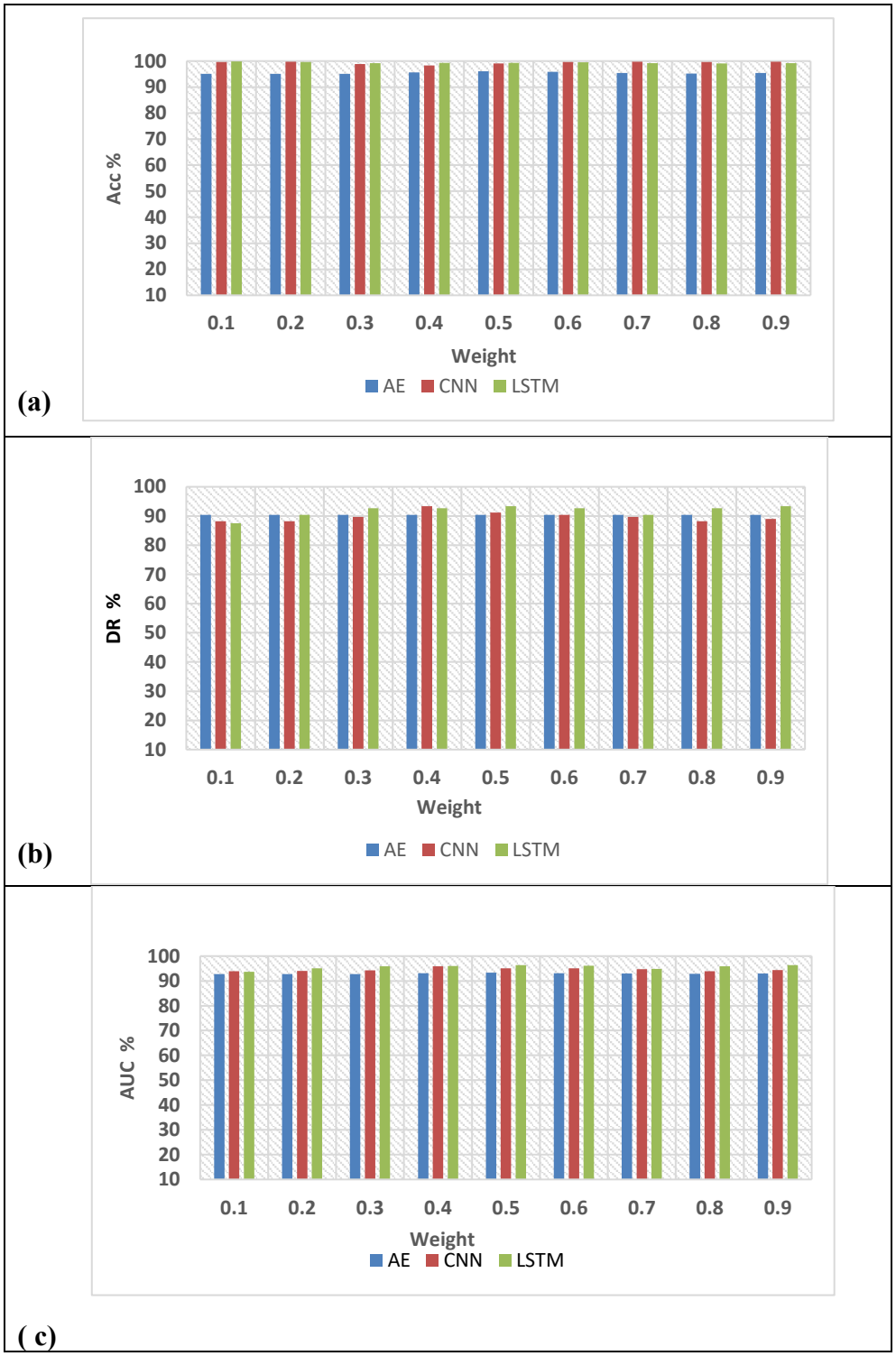

Figure 6: DL models performance of in terms of Acc, DR, and AUC with *w*ADASMO with different weights and European credit card dataset

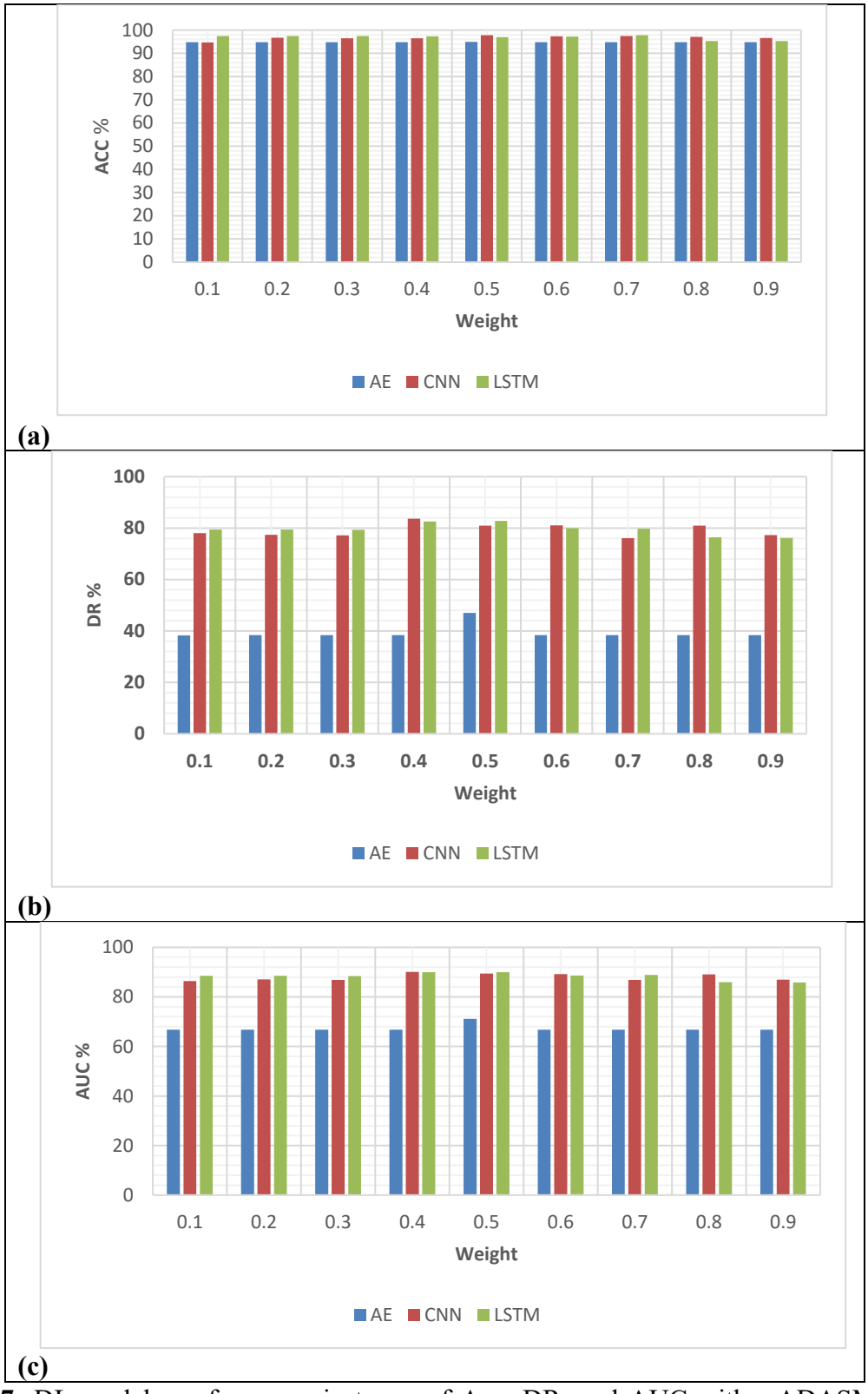

Figure 7: DL models performance in terms of Acc, DR, and AUC with *w*ADASMO with different weights and simulated credit card dataset

5.4 Results Comparison

Table 3 presents the performance of the wADASMO method in comparison to ADASYN and SMOTE. The results confirm that wADASMO is superior to the rest of the methods for both

credit card datasets. This is due to the proposed wADASMO method combining the strengths of ADASYN and SMOTE, offering improved adaptiveness and a more balanced class distribution, which makes it potentially more effective in handling credit card fraud imbalanced datasets by capturing the true positives (fraudulent cases) while controlling false positives. The DR and AUC are higher for European and simulated credit card datasets when AE, CNN, and LSTM models are used. This means that the models are better at finding fraudulent transactions and generalizing the results. The DR results for the European dataset are 90.4%, 93.3%, and 93.3%, respectively, while for the simulated dataset the results are 47%, 83.6%, and 78.8%, respectively.

It's important to emphasize that even though there is a slight difference in the CNN model's accuracy when it comes to analyzing the European and simulated credit card data, the difference in accuracy is only 1.0 and 0.1, respectively, and it still performs competitively with the best results. The difference in values could be due to the dataset's characteristics, as well as the required level of adaptability and sophistication for addressing class imbalance.
The proposed wADASMO results prove that it is a promising oversampling method for handling imbalanced data problems across different deep-learning models. This indicates that wADASMO is not affected by the type of DL model and is effective in addressing the issue of imbalanced data in credit card fraud detection and improving the ability of the models to distinguish fraudulent from legitimate transactions.

Table 3: Performance comparison between deep learning models with *w*ADASMO against SMOTE and ADASYN techniques.

Dataset	DL model	Sampling method	Acc%	DR%	AUC%
European credit card	AE	SMOTE	95.127	90.441	92.788
		ADASYN	95.1278	90.441	92.800
		*w*ADASMO	**95.628**	**90.441**	**93.039**
	CNN	SMOTE	99.745	90.441	95.100
		ADASYN	**99.823**	88.970	94.405
		*w*ADASMO	98.801	**93.382**	**96.096**
	LSTM	SMOTE	99.222	93.3	96.307
		ADASYN	**99.740**	89	94.363
		*w*ADASMO	99.292	**93.382**	**96.342**
Simulated credit card	AE	SMOTE	93.900	40.572	67.377
		ADASYN	94.876	43.500	69
		*w*ADASMO	**94.968**	**47.039**	**71.131**
	CNN	SMOTE	96.653	78.284	87.456
		ADASYN	**96.469**	77.263	87
		*w*ADASMO	96.468	**83.662**	**90.099**
	LSTM	SMOTE	93.397	76.276	84.882
		ADASYN	97	77.944	87.962
		*w*ADASMO	**97.495**	**78.897**	**88.245**

In Table 4, the suggested model's performance outcomes are compared with previous studies [8, 11, 10, and 17] when using the European credit card dataset as evaluation data. The results indicate that the proposed models outperform previous works [8, 11, 10, and 17] in terms of detection rate and AUC, meaning that they are effective at making correct predictions and capturing fraud transactions. Among the proposed models, LSTM with wADASMO has the highest accuracy, detection rate, and AUC, while CNN with wADASMO and AE with wADASMO both perform well. The innovative wADASMO sampling method is attributed to this success, as it generates synthetic samples in challenging areas of the feature space, thus

enhancing generalization in imbalanced datasets. Unlike earlier studies, the proposed wADASMO generates instances with different density distributions in regions of greater class imbalance. This is reflected in the particularly higher values of DR and AUC compared to the referenced studies [8, 10, 11, and 17].

Table 4: Performance comparison among the proposed DL models against previous works

DL Model	Acc%	DR%	AUC%
The proposed AE with *w*ADASMO	95.628	**90.441**	**93.039**
AE-PRF with ADASYN [8]	99.6	86.13	-
AE with ADASYN [17]	95.1278	90.441	92.800
The proposed CNN with *w*ADASMO	98.801	**93.382**	**96.096**
CNN with SMOTE [11]	89.74	76.9	89.49
CNN with SMOTE [17]	99.745	90.441	95.100
The proposed LSTM with *w*ADASMO	**99.292**	**93.382**	**96.342**
LSTM with SMOTE [10]	96.72	91.91	-
LSTM with SMOTE [17]	99.222	93.3	96.307

6. Conclusion

In this paper, a new oversampling technique called wADASMO is introduced to address the issue of overgeneralization that often arises when synthetic data is created with an even distribution in the majority class area. This method comprises three steps: firstly, synthetic data are generated using SMOTE; secondly, synthetic data are generated using ADASYN; and finally, a weighted method is employed to generate new synthetic data. Through extensive experiments conducted on two benchmark credit card fraud datasets, it was observed that the proposed method, wADASMO, generates synthetic samples by making a balance between the ADASYN and SMOTE techniques when the value of w is set to 0.5. This emphasizes the adaptive nature of ADASYN, which focuses on harder-to-learn instances while also leveraging the general oversampling technique of SMOTE. The results provide strong evidence that wADASMO outperforms SMOTE and ADASYN with accuracy of 95.6%, 98.8%, and 99.2%, detection rate of 90.4%, 93.38%, and 93.38%, and AUC of 93%, 96%, and 96.3% for the AE, CNN, and LSTM models, respectively. This contributes to improving the detection of credit card fraud by addressing the challenges related to imbalanced classes and offering valuable insights for future research in this area. For future work, there is excellent potential to further improve credit fraud detection by combining ADASYN with a variant of SMOTE.

Acknowledgment

The authors would like to thank Al-Mustansiriyah University in Baghdad, Iraq, for their cooperation with this study (http://uomustansiriyah.edu.iq).

References

[1] E. Strelcenia and S. Prakoonwit, "Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation," *AI*, vol. 4, no. 1, pp. 172–198, Jan. 2023. doi: 10.3390/ai4010008.

[2] N. H. Ali, M. E. Abdulmunem, and A. E. Ali, "Learning Evolution: a Survey," *Iraqi Journal of Science*, vol. 62, no. 12, pp. 4978–4987, Dec. 2021. doi: 10.24996/ijs.2021.62.12.34.

[3] C. Elkan, "The foundations of cost-sensitive learning," *presented at the International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Associates Ltd*, pp. 973–978, Dec. 2021.

[4] S. J. Muhamed, "Detection and Prevention WEB-Service for Fraudulent E-Transaction using APRIORI and SVM," *Al-Mustansiriyah Journal of Science*, vol. 33, no. 4, pp. 72–79, Dec. 2021. doi: 10.23851/mjs.v33i4.1242.

[5] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *EEE Transactions on Neural Networks and Learning Systems,* vol. 29, no. 8, pp. 3784–3797, Sep. 2017. doi: 10.1109/TNNLS.2017.2736643.

[6] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, Jul. 2019. doi: 10.1109/ACCESS.2019.2927266.

[7] W. Falah and I. J. Mohammed, "Hybrid CNN-SMOTE-BGMM Deep Learning Framework for Network Intrusion Detection using Unbalanced Dataset," *Iraqi Journal of Science*, vol. 64, no. 9, pp. 4846–4864, Sep. 2023. doi: 10.24996/ijs.2023.64.9.43.

[8] T. H. Lin and J. R. Jiang, "Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest," *Mathematics*, vol. 9, no. 21, p. 2683, Oct. 2021. doi:10.3390/math9212683.

[9] T. C. Tran and T. K. Dang, "Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection," *in 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea (South),* pp. 1–7, Jan. 2021, doi: 10.1109/IMCOM51814.2021.9377352.

[10] I. Benchaji, S. Douzi, and B. El Ouahidi, "Credit card fraud detection model based on LSTM recurrent neural networks," *Journal of Advances in Information Technology*., vol. 12, no. 2, pp.113-118, May 2021. doi:10.12720/jait.12.2.113-118.

[11] T. Berhane, T. Melese, A. Walelign, and A. Mohammed, "A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model," *Mathematical Problems in Engineering*., vol. 2023, pp. 1-10, Jun. 2023. doi: 10.1155/2023/8134627.

[12] I. D. Mienye and Y. Sun, "A Deep Learning Ensemble with Data Resampling for Credit Card Fraud Detection," *IEEE Access*, vol. 11, pp. 30628–30638, Mar. 2023. doi: 10.1109/ACCESS.2023.3262020.

[13] B. Rabiul Alam, M. Shimu Khatun, M. Taslim, and M. Alam Hossain, "Handling Class Imbalance in Credit Card Fraud Using Various Sampling Techniques," American Journal of Multidisciplinary Research and Innovation, vol. 1, no. 4, pp. 160–168, Oct. 2022. doi: 10.54536/ajmri.v1i4.633.

[14] Z. Salekshahrezaee, J. L. Leevy, and T. M. Khoshgoftaar," The effect of feature extraction and data sampling on credit card fraud detection," *Journal of Big Data*, vol. 10, no. 1, pp. 6, Jan. 2023. doi: 10.1186/s40537-023-00684-w.

[15] J. Karthika and A. Senthilselvi, "An integration of deep learning model with Navo Minority Over-Sampling Technique to detect the frauds in credit cards," *Multimedia Tools Applications,* vol. 82, no. 14, pp. 21757–21774, Jun. 2023. doi: 10.1007/s11042-023-14365-6.

[16] H. Tu, Y. Xie, A. Li, B. Hu, and L. Gao, "A Credit Card Fraud Detection Model Based on Multi-Feature Fusion and Generative Adversarial Network," *Computer Materials. Continua*, vol. 76, no. 3, pp. 2707–2726, Oct. 2023. doi: 10.32604/cmc.2023.037039.

[17] S. S. Sulaiman, I. Nadher, and S. M. Hameed, "Credit Card Fraud Detection Using Improved Deep Learning Models," *Computers, Materials and Continua*, vol. 78, no. 1, pp. 1049–1069, Jan. 2024. doi: 10.32604/cmc.2023.046051.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence. Research*, vol. 16, pp. 321–357, 2002. doi: https://doi.org/10.1613/jair.953.

[19] H. Haibo, B. Yang, E. A. Garcia, and L. Shutao, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong,* pp. 1322–1328. Jun. 2008. doi: 10.1109/IJCNN.2008.4633969.

[20] M. A. Sharma, B. G. Raj, B. Ramamurthy, and R. H. Bhaskar, "Credit Card Fraud Detection Using Deep Learning Based on Auto-Encoder," *Fourth International Conference on Advances*

in Electrical and Computer Technologies 2022 (ICAECT 2022), ITM Web of Conferences, India, vol. 18, Dec. 2022. doi: https://doi.org/10.1051/itmconf/20225001001

[21] A. Pumsirirat and Y. Liu, Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine," *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 1, pp. 18-25, 2018. doi: 10.14569/IJACSA.2018.090103.

[22] N. Rosley, G. K. Tong, K. H. Ng, S. N. Kalid, and K. C. Khor, "Autoencoders with Reconstruction Error and Dimensionality Reduction for Credit Card Fraud Detection," *vol. Proceedings of the International Conference on Computer, Information Technology and Intelligent Computing (CITIC 2022). Atlantis Press*, pp. 503–512, Dec. 2022. [Online]. Available: https://doi.org/10.2991/978-94-6463-094-7_40

[23] R. Ball, H. Kruger, and L. Drevin, "Anomaly detection using autoencoders with network analysis features," *ORiON journal*, vol. 39, no. 1, pp. 1–44, Jun. 2023. doi: https://doi.org/10.5784/39-1-711.

[24] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," *Expert Systems with Applications*, vol. 193, May 2022. doi: 10.1016/j.eswa.2021.116429.

[25] M. S. Kwon, Y. G. Moon, B. Lee, and J. H. Noh, "Autoencoders with exponential deviation loss for weakly supervised anomaly detection," *Pattern Recognition Letters*, vol. 171, pp. 131–137, Jul. 2023. doi: 10.1016/j.patrec.2023.05.016.

[26] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4244–4268, Jul. 2022. doi: 10.1016/j.jksuci.2020.10.004.

[27] T. A. Wotaifi and B. N. Dhannoon, "An Effective Hybrid Deep Neural Network for Arabic Fake News Detection," *Baghdad Science Journal*, vol. 20, no. 4, pp. 1392, Jul. 2022. doi: 10.21123/bsj.2023.7427.

[28] M. H. Al-Tai, B. M. Nema, and A. Al-Sherbaz, "Deep Learning for Fake News Detection: Literature Review," *Al-Mustansiriyah Journal of Science*, vol. 34, no. 2, pp. 70–81, Jun. 2023. doi: 10.23851/mjs.v34i2.1292.

[29] Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang, "A Model Based on Convolutional Neural Network for Online Transaction Fraud Detection," *Security and. Communication Network*, vol. 2018, p. 5680264, Aug. 2018. doi: 10.1155/2018/5680264.

[30] H. H. Ali, J. R. Naif, and W. R. Humood, "A New Smart Home Intruder Detection System Based on Deep Learning," *Al-Mustansiriyah Journal of Science*., vol. 34, no. 2, pp. 60–69, Jun. 2023. doi: 10.23851/mjs.v34i2.1267.

[31] S. Wang, C. Ma, Y. Xu, J. Wang, and W. Wu, "A Hyperparameter Optimization Algorithm for the LSTM Temperature Prediction Model in Data Center," *Scientific Program*ming, vol. 2022, Dec. 2022. doi: 10.1155/2022/6519909.

[32] European cardholders Dataset, 2013. Accessed: Jan. 01, 2022. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[33] Simulated Credit Card Transactions generated using Sparkov, 2020. Accessed: Jan. 10, 2023. [Online]. Available: https://www.kaggle.com/datasets/kartik2112/fraud-detection

[34] S. Szeghalmy and A. Fazekas, "A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning," *Sensors*, vol. 23, no. 4, Feb. 2023. doi: 10.3390/s23042333.