Received: May 19, 2024. Revised: June 24, 2024. 

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Credit Card Fraud Detection Using an Autoencoder Model with New Loss Function

**Sumaya S. Sulaiman[1,2]*** **Ibraheem Nadher[1]** **Sarab M. Hameed[2]**

[1]*Computer Science Department, College of Science, Al-Mustansiriyah University, Iraq*
[2]*Computer Science Department, College of Science, University of Baghdad, Iraq*
* Corresponding author's Email: sumasaad@uomustansiriyah.edu.iq

**Abstract:** The threat of credit card fraud to financial institutions and their customers is enormous which makes it essential to improve the fraud detection methods. The detection of credit card fraud is a problem for traditional detection methods. This paper presents a new loss function named Regularized Binary Cross Entropy (RBCE) in conjunction with an Autoencoder model. The RBCE loss function aims to improve the traditional BCE loss by incorporating regularization, enhancing the autoencoder's capability to learn robust and meaningful feature representations. This is particularly useful in situations involving high-dimensional and noisy data. The regularization term helps to reduce overfitting by penalizing model complexity, promoting the learning of more generalizable features. Consequently, the autoencoder's ability to identify meaningful features and detect anomalies becomes more accurate. RBCE improves the model's sensitivity to subtle deviations from normal patterns, leading to more precise detection of fraudulent transactions and other anomalies. Two datasets are used in the experiments: European and simulated credit cards. The results show the efficiency of the proposed model in improving fraud detection in both datasets and capturing complex patterns and anomalies in high-dimensional. Moreover, the proposed model outperforms prior works in terms of accuracy (95.130%), detection rate (91.176%), and area under the curve (93.156%) which produces significant improvement in fraud detection.

Keywords: Autoencoder, Credit card fraud detection, Deep learning, Loss function, Sampling.

## 1. Introduction

The threat of credit card fraud to customers and financial institutions is increasing which makes it important to improve the fraud detection methods. Detecting credit card fraud includes some difficulties. One of the serious problems is the disparity between legitimate and fraudulent transactions in credit card datasets. Second, fraudsters are always coming up with innovative and complex ways to evade discovery. Third, the effectiveness of fraud detection algorithms depends on careful feature engineering and selection. Finally, fraud detection in real-time is crucial. Therefore, cooperation between data scientists and financial organizations is needed to overcome these problems [1-3].

An Autoencoder (AE) is convenient for detecting credit card fraud because it provides a meaningful representation of complex data. An AE's efficacy depends on its ability to capture relevant features and anomalies which makes it an effective approach for improving the accuracy and detection of fraud [3-5]. Earlier studies have mostly concentrated on enhancing the model structure. However, traditional loss functions are not adequate to address the problems related to credit card fraud detection [6-8]. This paper aims to enhance AE model performance for credit card fraud detection, by introducing a new loss function to facilitate AE model convergence to an optimal loss during training and generalization to new data samples. The primary contributions of this paper are as follows.

Propose a new loss function that enhances the performance of the AE in detecting fraudulent transactions.

Improve the ability of an AE to capture and reconstruct complex features related to fraudulent activities by incorporating AE models into credit card

fraud detection to learn patterns and anomalies in transaction data that are indicative of fraud. This can help improve the efficacy of credit card fraud detection by reducing false negatives while retaining a low false positive rate.

The remaining sections of this work are organized as follows. Section 2 discuss several related works. Section 3 explains the fundamental concepts of AE, loss functions, and resampling techniques. Section 3 presents the proposed model. The results of the proposed model are presented in Section 5. Finally, in Section 6, the conclusions and future work are presented.

## 2. Related work

A new approach was proposed in [9] that merges Spark with a deep learning AE approach. Spark accomplishes two tasks: it combines historical transactions to achieve design engineering and it classifies transactions online to return the estimated risk of fraud. Different parameters and ML techniques, including RF, LR, ANN, DT, and SVM, were used in the comparative analysis. An accuracy of over 94 % was achieved for testing datasets. The paper lacked a sufficient range of metrics to adequately assess the approach.

To respond in real-time, an auto-encoder (AE) and Restricted Boltzmann Machine (RBM) were proposed in [10], which can find anomalies from the reconstructed normal patterns by applying backpropagation. The results show that the Area Under Curve (AUC) for AE and RBM were 96.03 and 95.05, respectively. But, the training model was computationally intensive and time-consuming.

An evolving pattern change was detected by developing an AE model in [11]. This model comprises four hidden layers, with two encoders and two decoders. Both the encoder and decoder use "tanh" and "ReLU" activation functions in adjacent layers. The model's performance was evaluated using three datasets (European, Australian, and Taiwanese). Results indicate a 99% accuracy with the Taiwanese dataset, while the European dataset achieved a lower accuracy of 70%. The study does not address the interpretability of the model and how the decisions are made by the autoencoder.

Lin [7] used oversampling techniques including Adaptive Synthetic Sampling (ADASYN), Tomek link (T-Link), and synthetic minority oversampling techniques (SMOTE) to European cardholders' dataset to balance the numbers of fraudulent and legitimate transactions for improving the proposed AE with probabilistic random forest (AE-PRF) performance. The experimental results demonstrated that the performance of the AE-PRF was steady irrespective of whether oversampling techniques were applied. The results were compared in terms of accuracy, true positive rate (TPR), true negative rate (TNR), Matthews correlation coefficient (MCC), and area under the receiver operating characteristic curve (AUC-OC) 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.

The AE in [12] was utilized by Bayesian hyperparameter optimization to determine the number of nodes in the hidden layers, activation function, epochs, and batch size. The model is then employed to encode the data used to train three additional models: K-nearest neighbours (KNN), logistic regression (LR), and support vector machine (SVM). These three models were applied to an imbalanced European cardholder's dataset. The results showed that the AE had a recall rate of more than 80% and a high accuracy of 99%. This model encountered difficulties in identifying intricate and evolving fraud patterns that were not adequately represented in the training set.

In a prior study [13], suspicious activity in fraudulent financial transactions was identified using deep learning (DL) techniques like Convolutional Neural Networks (CNN), AE, and Recurrent Neural Networks (RNN) to construct a classification model for fraud detection in finance. An ensemble classification model was then created. To address the dataset's imbalance, SMOTE was employed. Subsequently, DL models were applied. Experiments on a public European credit card dataset indicated that among the individual DL models, AE achieved the highest validation accuracy (93.4%), outperforming CNN (91.4%) and RNN (91.8%). The study did not account for the imbalance issue in the dataset before using it to build the classification model. Furthermore, the training and learning process took up a considerable amount of time.

Salekshahrezaee in [14] investigated RUS, SMOTE, and SMOTE-Tomek methods to mitigate class imbalance using a European credit card fraud dataset and four ensemble classifiers: RF, CatBoost, LightGBM, and XGBoost, along with Principal Component Analysis (PCA) and Convolutional - Autoencoder (CAE) methods for feature extraction. The results show that implementing the RUS method followed by the CAE method leads to the best performance for credit card fraud detection, with a 95.4% F1-Score.

In [15], three DL models, namely AE, CNN, and LSTM, were implemented and optimized using hyperparameter tuning techniques, including random and Bayesian methods, as well as RUS, SMOTE, and ADASYN sampling methods. The European credit card fraud dataset was utilized to evaluate the performance of the models, and the results showed that the AE, CNN, and LSTM models with ADASYN outperformed the other methods. The best-achieved results included an accuracy of over 95%, a detection rate of over 90%, and an area under the curve of over 92%.

Previous research has primarily focused on employing various machine learning and deep learning approaches, including AE, CNN, and LSTM, in conjunction with different sampling methods to address class imbalance in credit card fraud detection. However, there has been no investigation into the effects of different loss functions when used with AE. This paper seeks to address this gap by proposing a new loss function specifically designed for autoencoder-based credit fraud detection. This proposed loss function has the potential to significantly impact model performance, especially when the class distribution is unevenly balanced. By optimizing the training process for imbalanced datasets, the proposed approach could lead to more effective fraud detection.

## 3. Preliminary concepts

The theoretical basis of the concept used in this paper is explained briefly in this section.

### 3.1 Loss function

A loss function is an essential component of a DL model that assesses the performance of the model during training by comparing its predicted output values $\hat{y}_i$ with the actual target values $y_i, \forall i = 1,2, \dots, n$. The main goal is to minimize the difference between the values of the two sets. The following are some variations of loss functions [14, 16-18]:

- Mean Squared Error (MSE) function calculates the mean of the squared differences between the output values and the model's predictions as described in Eq. (1). However, this method is susceptible to outliers, so it's important to handle them carefully when using the MSE loss function.

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

- Binary Cross Entropy loss (BCE) function, also referred to as log loss, measures the disparity between the predicted probability of a class and the true class label as in Eq. (2). This loss function is frequently used for binary classification tasks and provides several advantages such as differentiability, computational simplicity, and a probabilistic interpretation of the model's output.

$$\mathcal{L}_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(y_i\, log(\hat{y}_i) + (1 - y_i)\, log(1 - \hat{y}_i)) \quad (2)$$

- Weighted Binary Cross Entropy (WBCE) function is a version of BCE where weights, $w_i$ are assigned to each sample, and the loss for each sample is determined by this weight as formulated in Eq. (3). This approach was especially helpful in cases when the distribution of the sample was non-uniform.

$$\mathcal{L}_{wBCE} = -\sum_{i=1}^{n}(w_i y_i\, log(\hat{y}_i) + w_i(1 - y_i)\, log(1 - \hat{y}_i)) \quad (3)$$

- Categorical Cross Entropy Loss (CCE) function is used in multi-class classification scenarios to measure the difference between the predicted probability distribution and the actual distribution as defined in Eq. (4). This is also known as a negative log-likelihood loss or multi-class log loss.

$$\mathcal{L}_{CCE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_{i,j}\, log(\hat{y}_{i,j}) \quad (4)$$

- Binary Focal loss (BFL) is a modified version of standard cross-entropy loss that addresses class imbalance as in Eq. (5). This issue arises when the number of positive samples is significantly smaller than the number of negative samples, which causes the model to prioritize negative samples over positive ones, resulting in poor performance. Focal Loss solves this problem by assigning higher weights to challenging positive samples and lower weights to easily negative samples.

$$\mathcal{L}_{BFL} = -\sum_{i=1}^{n}\big(\alpha(1 - \hat{y}_i)^{\gamma}\, y_i log(\hat{y}_i) - (1 - \alpha)\, \hat{y}_i^{\gamma}\, (1 - y_i) log(1 - \hat{y}_i)\big) \quad (5)$$

$\gamma$ a focusing parameter that controls the focus on hard-to-classify samples.

$\alpha$ Weighting parameter that controls the importance of each sample.

### 3.2 Autoencoder

An AE is a type of artificial neural network primarily employed for unsupervised learning tasks. The fundamental idea behind an AE is to learn a compact representation or encoding of input data by mapping it to a lower-dimensional space and then reconstructing the input data from this reduced representation. It consists of two parts: an encoder E and a decoder D (see Fig. 1) [10-12, 19]. The encoder aims to encode input into encoding vectors using a set of recognition weights based on the mapping function f as in Eq. (6), whereas the decoder obtains an approximation to the output features back from the encoding vector using a set of generative weights through mapping function g, as in Eq. (7). The encoder is designed so that the output produces a latent feature representation $Z \in \mathbb{R}^{d_h * n}$, i.e. a compressed version of the input $Y \in \mathbb{R}^{d_y * n}$. So, it is designed such that the inputs have much larger dimensions than the output. The decoder is designed to decompress the latent variables back to the original dimensions $\hat{Y} \in \mathbb{R}^{d_y * n}$.

$$z = f_{\varphi}(y) = E_f(wy + b) \tag{6}$$

where $w$ is a recognition weight $W \in \mathbb{R}^{d_y * d_h}$, b is a bias, and $E_f$ is the encoder $E$ activation function (typically the element-wise sigmoid, hyperbolic tangent, or Relu non-linearity functions).

$$\hat{y} = g_{\varphi}(z) = D_g(\grave{w}z + \grave{b}) \tag{7}$$

where ẁ is a generative weight $\grave{W} \in \mathbb{R}^{d_y * d_h}$, $b$` is a bias, $D_g$ is the decoder $D$ activation function, and $\varphi = \{w, b, \grave{w}, \grave{b}\}$.

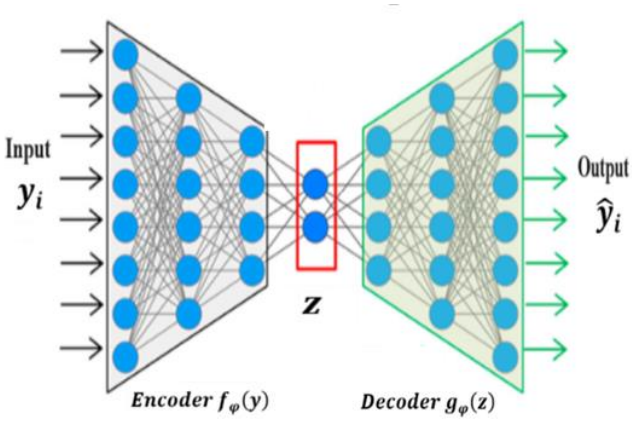


Figure. 1 Components of AE

### 3.3 Regularization

Regularization is used to issue overfitting by modifying the architecture of the model and adjusting the training process. There are different types of regularization methods including L1, L2, and dropout regularizations [20, 21].

In L1 regularization, also called Lasso regression, the absolute value of the weights multiplied by a regularizer term is used as a penalty. In L2 regularization, also called Ridge regression, the squared magnitude of the weights multiplied by a regularizer term is used as the penalty. The main difference between L1 and L2 regularization is that L1 regularization tries to approximate the data median data, while L2 regularization tries to approximate the data mean to avoid overfitting. In dropout regularization, some neurons are randomly deactivated during training, allowing the model to extract more robust and useful features. Eq. (9) and (10) show the formulations of L1 and L2 respectively.

$$L1 = \sum_{i=1}^{m} |w_i| \tag{9}$$

$$L2 = \sum_{i=1}^{m} w_i^2 \tag{10}$$

where $m$ is the number of features; $w_i$ is the model's trainable weight.

### 3.4 Resampling techniques

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 that may look similar to the original data are generated, but these replicates may not be the same. Fig. 2 depicts the resampling techniques, on the left of the figure is undersampling, SMOTE in the middle, and ADASYN on the right. A description of each technique is presented as follows [14, 15, 22-24]:

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. The technique works by drawing a line between the data samples of the minority class and 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 as in Eq. (11). 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 risk of overlapping data points for the minority class in SMOTE, which may weaken the boundary and increase the possibility of misclassification of the boundary samples.

$$y_{new} = y_i + rand(0,1) * \left( y_{ij} - y_i \right) \quad (11)$$

where $x_i$ and $x_j$ are two minority adjacent samples.

ADASYN considered as SMOTE extension that overcomes some of the traditional SMOTE limitations. ADASYN was used to address data imbalance by increasing the representation of minority classes in the datasets. This creates minority data samples that reflect the distributions of underrepresented groups to generate more data. 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 also reduce the learning bias of the actual dataset. However, this algorithm may suffer from reduced precision due to its adaptability. Additionally, the neighbourhoods created by ADASYN contain only one minority example for minority samples that are sparsely distributed.

RUS is a technique that handles issues of class imbalance by randomly deleting instances of the majority class, hence bringing the number of samples in the majority class (i.e., legitimate transactions) down to match those in the minority class. However, the main issue with RUS is that this random removal of data can lead to the loss of crucial information contained in the removed samples.

## 4. Proposed AE credit card fraud detection model

An AE is commonly used for fraud detection in credit card transactions and offers robust capabilities for identifying anomalies and learning new features. However, there are limitations in adopting an AE for fraud detection. The proposed AE model with a new loss function dedicated to credit card fraud detection aims to address limitations including sensitivity to anomalies that lead to distorted reconstructions and deter the learning process, managing imbalanced data, and generalization that can make the AE unable to identify new fraud.

### 4.1 Proposed fraud-custom loss function

BCE loss function is normally formulated with Eq. (2) in AE models to measure the significant difference between the reconstructed output and the input data. In this way, AE models attempt to accurately predict legal or fraud transactions based on their distinct characteristics, thus reducing BCE loss values. Despite their acceptable direction, they suffer from two main shortcomings. AE model issues are overfitting and failing to generalize for new transactions of fraud. To address this issue, this study integrates BCE loss function and L2 regularization into the training process of a new loss function, called Regularized BCE loss function (RBCE). The incorporation of BCE loss function and L2 regularization, in turn, would likely empower the proposed model in twofold. First, the proposed model would be able to hold off irrelevant features and work on ranks of relevant features that are enough to discriminate between fraudulent and legitimate transactions. Second, the proposed model would be able to avoid overfitting that results from too tuning the model to normal transactions at the expense of fraud. Here, large weights in the proposed AE model are penalized with L2 regularization, letting it learn simpler and smoother representations. These new representations can help better generalize new frauds not seen during training.

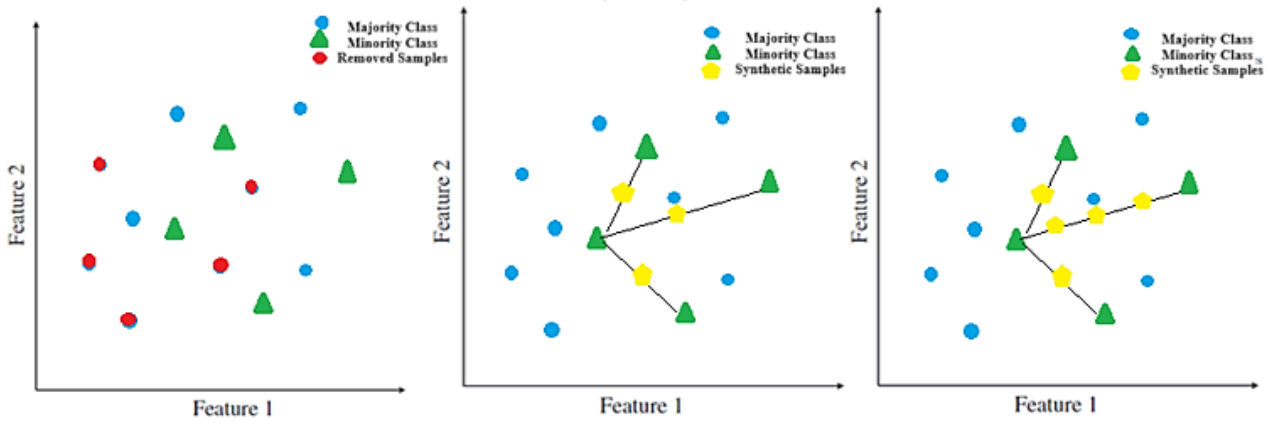


Figure. 2 Resampling methods [15]

The mathematical formulation of the proposed RBCE can be expressed in Eq. (12):

$$\mathcal{L}_{RBCE}(\varphi, y, \hat{y}) = \mathcal{L}_{BCE}(y, \hat{y}) + \lambda\, L_2 \qquad (12)$$

where $\lambda \in [0, \infty)$ hyperparameter weights measure the qualified contribution of the penalty term and the larger value means more regularization. The penalty term is computed based on the correct weights and is included in the computation of the gradients during backpropagation to decrease the weight magnitude and help prevent overfitting as in Eq. (13).

$$\mathcal{L}_{RBCE} = -\frac{1}{n}\sum_{i=1}^{n}(y_i\, log(\hat{y}_i) + (1 - y_i)\, log(1 - \hat{y}_i)) + \lambda\, \sum_{i=1}^{m} w_i^2 \qquad (13)$$

The objective is to minimize $\mathcal{L}_{\mathrm{BCE}}$ on the training data and the weights w that are used in calculating the gradients during backpropagation in $\varphi$, as formulated in Eq. (14).

$$\min_{w} \mathcal{L}_{BCE}(w) + \lambda\, L_2(w) \qquad (14)$$

### 4.2 AE model for credit card fraud detection

The AE model proposed for credit card fraud detection is intended to address several limitations, including a heightened sensitivity to anomalies that can result in distorted reconstructions and hinder the learning process, the management of imbalanced data, and the generalization of the AE model, which can lead to its identification of new fraud. The approach entails three stages. Initially, credit card transactions are pre-processed, followed by the determination of the optimal set of hyperparameters for the AE using a Bayesian hyperparameter optimization technique. Finally, the AE model is applied to detect credit card transactions. The proposed AE model for detecting credit card fraud is depicted in Fig. 3.

The process of data cleaning consists of handling missing values, normalization, and data splitting. Missing values can be handled by removal or imputation. If the missing values percentage less than a pre-set threshold, the missing values will be filled in using the mean value of the feature. However, if the percentage of missing values exceeds the threshold, the transactions containing those missing values will be removed from the dataset entirely. Following, Z-score normalization is applied to improve model convergence and prevent issues caused by varying feature scales. The feature values are standardized by setting their mean to 0 and standard deviation to 1. Finally, the dataset is divided into training and testing sets by applying stratified K-fold cross-validation since the dataset of fraud transactions is imbalanced (i.e., fraudulent transactions are significantly more numerous than legitimate transactions). The dataset is divided into K subsets while ensuring that each subset maintains the percentage of samples for each class. This is beneficial when dealing with imbalanced datasets, such as fraud detection.

The Bayesian optimization is applied in the second stage which involves hyperparameter tuning of AE model by developing a probabilistic model of the performance metric. The F1-score of the AE model, as shown in Eq. (15) [25] is used in this paper.

$$F1\ score = 2 * (DR + P)\ /(DR + P) \qquad (15)$$

where, DR stands for detection rate, sometimes known as the true positive rate (TPR), which indicates the proportion of fraudulent behaviour that the model correctly identified as fraudulent, as defined in Eq. (16).

$$DR = TP/(TP + FN) \qquad (16)$$

$$P = TP/(TP + FP) \qquad (17)$$

where, TP (true positive) denotes the number of fraudulent transactions correctly classified as fraudulent, TN (true negative) describes the number of legitimate transactions correctly classified as legitimate, FN (false negative) defines the number of fraudulent transactions misclassified as legitimate, and FP (false positive) is the number of legitimate transactions misclassified as fraud.

The third stage uses the proposed AE with RBCE to detect credit fraud. The AE is trained on the cleaned and normalized credit card dataset, learning a compressed representation of the data samples (latent space) through learning the encoder network's weight w. The latent space captures the essential features or patterns present in the input data in a more compact form. The decoder reconstructs the original input data from the compressed representation by training the decoder network's weight $\dot{w}$. The training process employs $\mathcal{L}_{\mathrm{RBCE}}$, as described in Eq. (12), which penalizes the model for large deviations between the input and the reconstructed output while also incorporating regularization techniques. The unique aspect of using this custom loss function with an AE is its potential to enhance the model's ability to reconstruct data accurately and generalize well to unseen data, benefiting various applications such as feature learning, data denoising, and anomaly

detection. This ensures that the regularization penalty is applied to the weights during training, and the gradients of the loss function concerning the weights include the gradient of the regularization term. Consequently, during backpropagation, the regularization term contributes to the gradients used to update the weights, effectively penalizing large weights and preventing overfitting.

## 5. Results discussion

The proposed AE with RBCE model is evaluated by comparing its effectiveness with the baseline loss functions including MSE, AE with BCE, and AE with focal loss. In addition, the experiments, which are conducted on two credit card fraud datasets: the European credit card dataset and the synthetic credit card dataset. Table 1 provides a detailed description of the two benchmark credit card fraud datasets.

- The European credit card dataset [26], composed of transactions made by European cardholders in September 2013, is used to evaluate fraudulent credit card transaction detection models. The dataset consists of 284,807 transactions. Each transaction has 31 features. The dataset is highly imbalanced, with only 492 fraudulent transactions, which account for 0.172% of the total data.
- The simulated dataset is synthetic data of credit card transactions generated by the Sparov tool which was developed by Brandon Harris [27]. The simulation was held over two years, from January 1, 2019, to December 31, 2020. It consists of 1,842,743 legitimate transactions and 9,651 fraudulent transactions. The performance of the proposed AE-RBCE model is evaluated with accuracy (Acc), detection rate, and AUC.

Accuracy, as expressed in Eq. (18), measures the overall performance of the model, while the detection rate measure (Eq. 16) evaluates the model's ability to correctly identify fraudulent transactions. Finally, the overall performance of the proposed model in terms of AUC (Eq. (19)) is measured to show the ability of the proposed model to distinguish between fraudulent and legitimate transactions [25, 28].

$$Acc = (TP + TN)/(TP + FN + TN + FP) \quad (18)$$

$$AUC = (1 + \frac{TP}{TP+FN} - \frac{FP}{TN+FP})/2 \quad (19)$$

The experiments are conducted on European and simulated credit card datasets using AE models with different loss functions and sampling techniques. From Tables 2 and 3, one can observe that the Acc, DR, and AUC results of the proposed loss function, $\mathcal{L}_{\mathrm{RBCE}}$, are far better than other loss functions. This points to $\mathcal{L}_{\mathrm{RBCE}}$ has enabled the capturing of fraudulent transaction patterns despite class imbalance. Furthermore, the proposed model attains the highest values of AUC with SMOTE and ADASYN oversampling techniques. This means that oversampling techniques can generate synthetic instances for minority class (i.e., fraud transactions), thereby improving the ability of the model to distinguish fraudulent from legitimate transactions. The proposed loss function, $\mathcal{L}_{\mathrm{RBCE}}$, andoversampling harnesses the detection rate of the proposed model to fraudulent patterns that lead to higher AUC values. The higher performance of the proposed highlights the importance of designing a proper loss function considering dataset characteristics of credit card fraud transactions. The comparison results of the proposed AE-RBCE model against other works [7, 13, 15]. The hyperparameter value for each model is presented in Table 4. The results, as shown in Table 5, demonstrate the superior performance of the AE-RBCE model across all sampling techniques. The accuracy ranges from approximately 95.1% to 95.13%. In addition, fraud detection ranges from approximately 90.4% to 91.176% and outperforms other models. Moreover, the AE-RBCE model shows superior AUC values ranging from approximately 92.7% to 93.156%. results confirm its effectiveness in distinguishing between fraudulent and legitimate transactions and the superior performance of the AE-RBCE model compared to previous studies in [7, 13, 15] across all sampling techniques demonstrates its potential as an advanced solution for detecting fraud in credit card transactions.

Table 1. Dataset Description

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

## 6. Conclusion

A new loss function called RBCE with an AE model for credit card fraud detection is presented in this paper. As AE model is designed to learn the condensed representation of legitimate transactions while effectively highlighting anomalies associated

with fraudulent activities. Its performance is enhanced by utilizing the suggested RBCE loss function due to its ability to capture complex patterns and anomalies in high-dimensional data making it suitable for fraud detection tasks.

Comparing the experimental results of the proposed to conventional methods indicates a notable decrease in false positives and a considerable improvement in fraud detection rates on the European and simulated credit card datasets. The RBCE with AE model, which combines BCE, L2 regularization, and AE, exhibits the potential for developing an accurate and robust system that provides a promising method for enhancing credit card fraud detection systems. The result indicates that the AE with RBCE achieves an accuracy of 95.13%, surpassing the detection rate of 91.1% and attaining a higher AUC of 93.156%. This ensures accurate detection and outperforms other existing methods. Future studies can focus on refining the suggested model and exploring ways to reduce the number of features, enhancing its practicality in real-world applications. These efforts could lead to developing more efficient and reliable credit card fraud detection systems, benefiting both financial institutions and consumers.

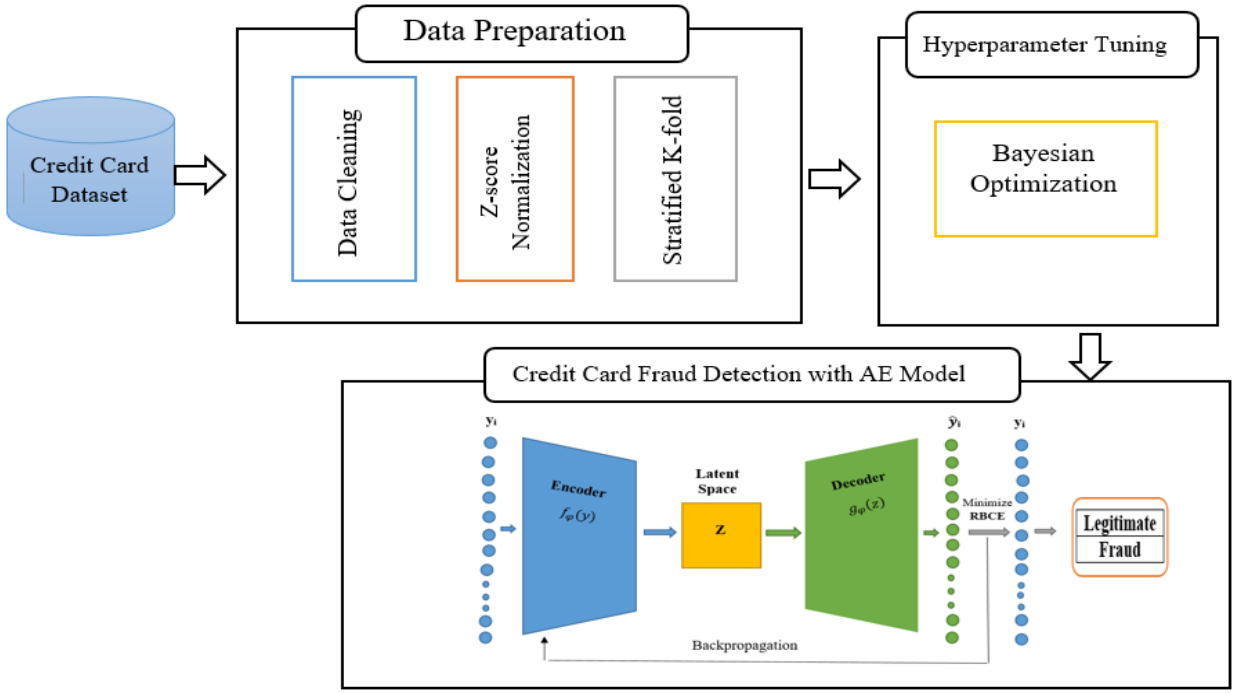


Figure. 3 The proposed AE with RBCE loss function block diagram

Table 2. Performance comparison of the proposed AE with RBCE against MSE, BCE, and BFL when European credit cards as an evaluation dataset.

| Model | Sampling Techniques | Acc% | DR% | AUC% |
|---|---|---|---|---|
| AE-MSE | SMOTE | 95.127 | 90.44 | 92.78 |
| | ADASYN | 95.127 | 90.44 | 92.788 |
| | Undersampling | 95.125 | 89.70 | 92.41 |
| AE-BCE | SMOTE | **95.123** | **88.970** | **92.051** |
| | ADASYN | **95.12** | **89.705** | **92.419** |
| | Undersampling | 95.125 | 89.70 | 92.41 |
| AE-Focal | SMOTE | 95.127 | 90.441 | 92.788 |
| | ADASYN | 95.1278 | 90.441 | 92.800 |
| | Undersampling | **95.127** | **90.440** | **92.780** |
| AE-RBCE | SMOTE | **95.130** | **91.176** | **93.156** |
| | ADASYN | **95.130** | **91.176** | **93.156** |
| | Undersampling | **95.127** | **90.441** | **92.788** |

Table 3. Performance comparison of the proposed AE with RBCE against MSE, BCE, and BFL when simulated credit card as evaluation dataset

| Model | Sampling Techniques | Acc% | DR% | AUC% |
|---|---|---|---|---|
| AE-MSE | SMOTE | **93.900** | **40.572** | **67.377** |
| | ADASYN | 94.974 | 38.155 | 66.715 |
| | Undersampling | 94.876 | 38.325 | 66.751 |
| AE-BCE | SMOTE | 94.846 | 35.500 | 65.331 |
| | ADASYN | 94.876 | 43.500 | 69 |
| | Undersampling | 94.865 | 37.304 | 66.237 |
| AE-focal | SMOTE | **94.654** | **17.358** | **56.212** |
| | ADASYN | 94.752 | 17.120 | 56.142 |
| | Undersampling | 94.657 | 17.562 | 56.314 |
| AE-RBCE | SMOTE | **94.933** | **43.669** | **69.437** |
| | ADASYN | **94.931** | **43.5** | **69.351** |
| | Undersampling | **94.886** | **39.244** | **67.213** |

Table 4. Hyperparameter values of the proposed AE-RBCE model and the models presented in [7, 13, 15]

| DL | Sampling Techniques | Hyperparameter Values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | No. of neurons per layer | Batch size | Optimization function | Activation function | Threshold (cut_off) | Loss Function | Learning rate |
| Proposed AE-RBCE | SMOTE | 512 | 64 | Adam | Relu | 0.95 | RBCE | 0.0001 |
| | ADASYN | 512 | 64 | Adam | Relu | 0.95 | RBCE | 0.0001 |
| | Undersampling | 512 | 128 | Adam | Relu | 0.95 | RBCE | 0.001 |
| [7] | SMOTE | 26 | 64 | Adam | Relu | 0.11 | MSE | - |
| | ADASYN | 26 | 64 | Adam | Relu | 0.13 | MSE | - |
| [13] | SMOTE | | 256 | - | - | 0.002 | MSE | 0.01 |
| AE Model [15] | SMOTE | 512 | 64 | Adam | Relu | 0.95 | BFL | 0.001 |
| | ADASYN | 512 | 64 | Adam | Relu | 0.95 | BFL | 0.001 |
| | Undersampling | 512 | 128 | Adam | Relu | 0.95 | BCE | 0.001 |

Table 5. Performance comparison of the proposed AE-RBCE against [7, 13, 15]

| Model | **Dataset** | Sampling Techniques | Acc% | DR% | AUC% |
|---|---|---|---|---|---|
| AE-RBCE | European credit card | SMOTE | 95.130 | **91.176** | **93.156** |
| | | ADASYN | 95.130 | **91.176** | **93.156** |
| | | Undersampling | **95.127** | **90.441** | **92.788** |
| [7] | European credit card | SMOTE | 99.65 | 85.83 | - |
| | | ADASYN | 99.6 | 86.13 | - |
| | | Undersampling | - | - | - |
| [13] | European credit card | SMOTE | 93.4 | - | - |
| | | ADASYN | - | - | - |
| | | Undersampling | - | - | - |
| AE Model [15] | European credit card | SMOTE | 95.1 | 90.4 | 92.7 |
| | | ADASYN | 95.1 | 90.4 | 92.8 |
| | | Undersampling | 95.1 | 89.7 | 92.4 |

## Notation

| Notation | Description |
|---|---|
| $\mathcal{L}_{MSE}$ | Mean Squared Error loss function |
| $\mathcal{L}_{BCE}$ | Binary Cross Entropy loss function |
| $\mathcal{L}_{wBCE}$ | Weighted Binary Cross Entropy loss function |
| $\mathcal{L}_{CCE}$ | Categorical Cross Entropy loss function |
| $\mathcal{L}_{BFL}$ | Binary Focal loss function |
| $\mathcal{L}_{RBCE}$ | Regularized Binary Cross Entropy loss function |
| $m$ | Number of features |
| $n$ | Number of dataset samples |
| $\gamma$ | Focusing parameter |
| $\alpha$ | Weighting parameter |
| $y$ | Actual target value |
| $\hat{y}$ | Predicted output value |
| $Z$ | Latent feature representation |
| $d_h$ | Hidden layer dimension |
| $d_y$ | Input layer dimension |
| $w$ | Recognition weight |
| $\grave{w}$ | Generative weight |
| $b$ | Bias parameter |
| $E_f$ | Encoder activation function |
| $D_g$ | Decoder activation function |
| $\varphi^*$ | Trainable parameter $\{w, b, \grave{w}, \grave{b}\}$ |
| $L1$ | Lasso regression |
| $L2$ | Ridge Regression |
| $\lambda$ | Penalty hyperparameter |
| $DR$ | Detection rate |
| $P$ | Precision |
| $Acc$ | Accuracy |
| AUC | Area under the curve |
| $TP$ | True positive |
| $TN$ | True negative |
| $FN$ | False negative |
| $FP$ | False positive |

## Conflicts of Interest

The authors declare that there is no conflict of interest.

## Author Contributions

Sarab M. Hameed, Sumaya S. Sulaiman, and Ibraheem Nadher designed the study and collected the data. Sarab M. Hameed and Sumaya S. Sulaiman were responsible for the analysis and interpretation of the results, as well as the production of the draft manuscript. The results were evaluated by all authors, who then approved the final version of the manuscript.

## Acknowledgments

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

## References

[1] S. S. Sulaiman, I. Nadher, and S. M. Hameed, "Credit Card Fraud Detection Challenges and Solutions: A Review", *Iraqi Journal of Science*, Vol. 65, No. 4, pp. 2287–2303, Apr. 2024.

[2] M. Alamri and M. Ykhlef, "Survey of Credit Card Anomaly and Fraud Detection Using Sampling Techniques", *Electronics*, Vol. 11, No. 23, Oct. 2022.

[3] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey", *Artificial Intelligence Review*, Vol. 57, No. 2, p. 28, 2024.

[4] H. Du, L. Lv, A. Guo, and H. Wang, "AutoEncoder and LightGBM for Credit Card Fraud Detection Problems", *Symmetry*, Vol. 15, No. 4, 2023.

[5] 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, 2023.

[6] S. El Kafhali, M. Tayebi, and H. Sulimani, "An Optimized Deep Learning Approach for Detecting Fraudulent Transactions", *Information*, Vol. 15, No. 4, 2024.

[7] T. H. Lin and J. R. Jiang, "Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest", *Mathematics*, Vol. 9, No. 21, p. 2683, 2021.

[8] 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, 2023.

[9] S. Sanober et al., "An enhanced secure deep learning algorithm for fraud detection in wireless communication", *Wireless Communications and Mobile Computing*, Vol. 2021, 2021.

[10] 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.

[11] M. A. Sharma, B. G. Raj, B. Ramamurthy, and R. H. Bhaskar, "Credit Card Fraud Detection Using Deep Learning Based on Auto-Encoder", In: *Proc. of Fourth International Conference on Advances in Electrical and Computer Technologies 2022 (ICAECT 2022), ITM Web of Conferences*, India, Vol. 18, 2022.

[12] 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", In: *Proc. of Proceedings of the International Conference on Computer, Information Technology and Intelligent Computing (CITIC 2022)*. Atlantis Press, pp. 503–512, 2022. [Online]. Available: https://doi.org/10.2991/978-94-6463-094-7_40

[13] S. Al-Faqir and O. Ouda, "Credit card frauds scoring model based on deep learning ensemble", *Journal of Theoretical and Applied Information Technology*, Vol. 100, No. 14, 2022.

[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, p. 6, 2023.

[15] 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, 2024.

[16] 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, 2022.

[17] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Deep learning with focal loss approach for attacks classification", *TELKOMNIKA Telecommunication Computing Electronics and Control*, Vol. 19, No. 4, pp. 1407–1418, 2021.

[18] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42, No. 2, pp. 318–327, 2020.

[19] S. Chen and W. Guo, "Auto-Encoders in Deep Learning—A Review with New Perspectives", *Mathematics*, Vol. 11, No. 8, 2023.

[20] B. Kim, K. H. Ryu, J. H. Kim, and S. Heo, "Feature variance regularization method for autoencoder-based one-class classification", *Computers & Chemical Engineering*, Vol. 161, p. 107776, 2022.

[21] S. Khanam, I. Ahmedy, M. Y. I. Idris, and M. H. Jaward, "Towards an Effective Intrusion Detection Model Using Focal Loss Variational Autoencoder for Internet of Things (IoT)", *Sensors*, Vol. 22, No. 15, 2022.

[22] 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.

[23] 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, 2023.

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

[25] 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, 2023.

[26] "European cardholders Dataset." 2014 2013. Accessed: Jan. 01, 2022. [Online]. Available: https://www.kaggle.com/datasets/mlgulb/creditcardfraud

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

[28] 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, 2023.