Optimizing credit card fraud detection: a deep learning approach to imbalanced datasets

Oussama Ndama, Ismail Bensassi, El Mokhtar En-Naimi
DSAI2S Research Team, Computer Science and Smart Systems Laboratory, FST of Tangier, Abdelmalek Essaâdi University, Tetouan, Morocco

Article Info

Article history:

Received Dec 22, 2023
Revised Apr 30, 2024
Accepted May 18, 2024

Keywords:

Deep neural networks
Fraud detection
Imbalanced datasets
Optimization
Resampling

ABSTRACT

Imbalanced datasets pose a significant challenge in credit card fraud detection, hindering the training effectiveness of models due to the scarcity of fraudulent cases. This study addresses the critical problem of data imbalance through an in-depth exploration of techniques, including cross-entropy loss minimization, weighted optimization, and synthetic minority oversampling technique-based resampling, coupled with deep neural networks (DNNs). The urgent need to combat class imbalances in credit card fraud datasets is underscored, emphasizing the creation of reliable detection models. The research method delves into the application of DNNs, strategically optimizing and resampling the dataset to enhance model performance. The study employs a dataset from October 2018, containing 284,807 transactions, with a mere 492 classified as fraudulent. Various resampling techniques, such as undersampling and SMOTE oversampling, are evaluated alongside weighted optimization. The results showcase the effectiveness of SMOTE oversampling, achieving an accuracy of 99.83% without any false negatives. The study concludes by advocating for flexible strategies, integrating cutting-edge machine learning methods, and developing adaptive defenses to safeguard against emerging financial risks in credit card fraud detection.

Corresponding Author:

Oussama Ndama
DSAI2S Research Team, Computer Science and Smart Systems Laboratory, FST of Tangier, Abdelmalek Essaâdi University
Tetouan, Morocco
Email: oussama.ndama@etu.uae.ac.ma

1. INTRODUCTION

The issue of credit card fraud presents a persistent obstacle within the realm of finance, encompassing both legitimate and illegitimate transactions. The successful identification of these unlawful acts relies on the implementation of robust automated systems. The foundation of supervised credit card fraud detection is rooted in the complex construction of machine learning models. These models utilize historical transactional data to discern the subtle differences between authentic and fraudulent transactions, serving as a crucial method for promptly and accurately identifying fraud in real-time situations. However, imbalanced datasets are a pervasive and significant barrier that hinders the efficacy of machine learning algorithms in this field. In the realm of credit card fraud detection, there exists a notable contrast between genuine transactions and fraudulent ones, with the former greatly surpassing the latter in terms of quantity. The existence of this imbalance is a significant obstacle to attaining the level of precision required for the effective identification of fraudulent activities. Addressing the issue of class imbalance requires the development of novel approaches aimed at enhancing the algorithms' effectiveness in detecting fraud consistently and reliably.

To tackle this difficulty, a comprehensive solution that incorporates diverse data-level methodologies is necessary. The approaches discussed in this study involve a range of strategies, which include oversampling, undersampling, and the incorporation of customized loss functions [1]–[5]. The primary objective of these strategies is to readjust the distribution of the dataset, aiming to alleviate the negative effects of class imbalance on the learning processes of algorithms. This work aims to critically evaluate the efficacy of complex data-level approaches. The evaluation criteria encompass fundamental measurements such as precision, recall, and F1-score. This study seeks to investigate the complexities associated with imbalanced datasets in the context of credit card fraud detection, utilizing artificial neural networks as the major analytical tool. The primary goal is to enhance the dependability and efficacy of the model, thus establishing a more resilient and precise system for mitigating fraudulent behavior in financial transactions.

2. LITERATURE REVIEW

Numerous studies have focused on detecting credit card fraud and addressing the class imbalance within datasets. Researchers and practitioners have sought innovative strategies to address the skewed distribution of fraudulent versus legitimate transactions. The goal is to develop effective mechanisms to mitigate this imbalance, aiming to create more accurate, reliable, and resilient fraud detection systems in financial transactions.

In their study, Warghade *et al.* [6] highlight the pressing need for strong fraud detection systems in light of increasing financial losses. They investigated different machine learning algorithms to address the issue of imbalanced datasets in credit card fraud detection. Their objective was to improve the precision of fraud detection by avoiding the misclassification of legitimate transactions, with the goal of enhancing the efficiency of fraud detection algorithms in the financial industry.

In their study, Mrozek *et al.* [7] explore the field of credit card fraud detection while acknowledging the challenge that imbalanced datasets present. Their study examines the efficacy of machine learning algorithms in dealing with this type of data, demonstrating the advantage of combining random forest with random undersampling, resulting in a recall score of 100%. This highlights the importance of combining effective machine learning with appropriate resampling methods to identify fraud in imbalanced datasets.

The paper by Ebiaredoh-Mienye [8] introduces a stacked sparse autoencoder (SSAE) method to enhance credit card default prediction, which is crucial for financial institutions. Traditional methods face challenges with the dynamic and imbalanced nature of credit card data. The SSAE, an unsupervised feature learning approach, improves classifier performance by generating superior feature representations. The study demonstrates the SSAE's effectiveness, outperforming raw data-driven methods and previous works. This presents a promising solution for financial institutions seeking robust models for predicting credit card defaulters.

Adityasundar *et al.* [9] conducted a study on credit card fraud detection using machine learning. Their main objective was to differentiate between fraudulent and non-fraudulent transactions. Their study employed algorithms such as logistic regression, specifically emphasizing the significance of these models in efficiently detecting fraud.

The study conducted by Makki *et al.* [10] focused on detecting credit card fraud in datasets with a significant imbalance. Their research emphasized the shortcomings of current methods in addressing data imbalance. The resulting in a significant number of false alarms and presenting difficulties in achieving precise fraud detection.

In their study, Singh *et al.* [11] investigated credit card fraud detection in datasets with severe imbalances. They focused on the efficacy of using a combination of oversampling and undersampling techniques. Their thorough examination of several performance measures provided insights into the most effective ways for addressing skewed data in fraud detection algorithms.

Islam *et al.* [12] introduce a rule-based model (RBM) to combat financial fraud without employing resampling techniques. Addressing the challenge of imbalanced datasets in distinguishing fraudulent transactions, this study evaluates RBM against various established machine learning models. The RBM showcases superior performance, boasting remarkable accuracy and precision of 0.99, promising significant advancements in enhancing financial fraud detection systems.

Baesens *et al.* [13] tackled the problem of identifying fraud in datasets that have a significant imbalance between the number of fraudulent and non-fraudulent instances. They proposed an approach called RobROSE, which is specifically designed to handle imbalanced data and outliers. Their investigation demonstrated the effectiveness of robROSE in enhancing fraud detection by overlooking anomalies, highlighting its potential in comprehending intricate data structures. In addition, they provided open access to the source code of the RobROSE algorithm without charge.

3. RESEARCH METHOD

The research method section plays an essential role in presenting a well-constructed framework and procedural approaches aimed at addressing the ongoing issue of class imbalance in credit card fraud detection datasets. This chapter provides a comprehensive description of the dataset used, including the application of stringent preprocessing techniques that are necessary for enhancing the usability of the data for classifying fraudulent and non-fraudulent transactions. Deep neural networks are used on purpose as the basic model architecture in this method. This makes it possible to fully understand and effectively handle the complex dynamics that arise from imbalanced datasets when spotting fraudulent behavior.

Furthermore, this section provides a detailed description of the strategic optimization and resampling techniques that have been carefully incorporated to address and align the imbalanced data distribution. These approaches provide a full grasp of the methodological changes necessary for creating effective credit card fraud detection mechanisms. In addition, Figure 1 visually demonstrates the proposed framework architecture for addressing imbalanced datasets in credit card fraud detection.

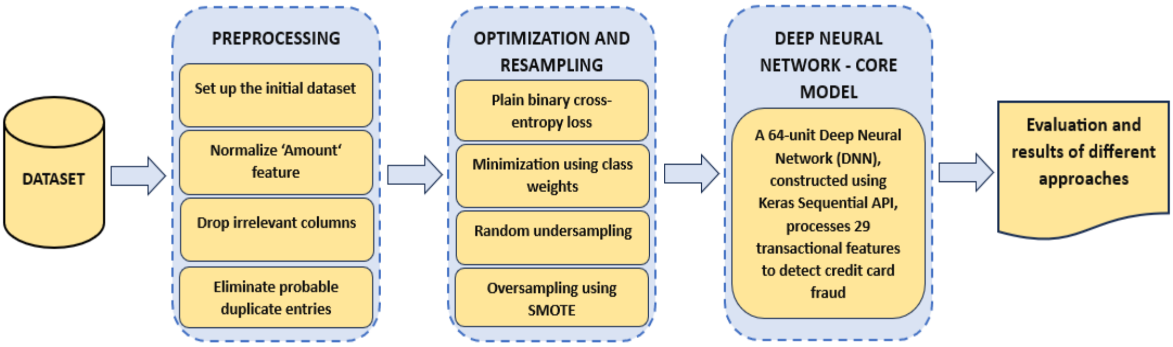
DATASET
PREPROCESSING
Set up the initial dataset
Normalize 'Amount' feature
Drop irrelevant columns
Eliminate probable duplicate entries
OPTIMIZATION AND RESAMPLING
Plain binary cross-entropy loss
Minimization using class weights
Random undersampling
Oversampling using SMOTE
DEEP NEURAL NETWORK - CORE MODEL
A 64-unit Deep Neural Network (DNN), constructed using Keras Sequential API, processes 29 transactional features to detect credit card fraud
Evaluation and results of different approaches

Figure 1. Proposed framework architecture for handling imbalanced dataset in credit card fraud detection

3.1. Dataset description and preprocessing

The dataset utilized in this study encapsulates credit card transactions made by European cardholders in October 2018, comprising a total of 284,807 transactions [14]. This dataset presents a formidable challenge due to its highly imbalanced nature, with a majority of legitimate transactions vastly outnumbering instances of fraudulent activities. Within this extensive dataset, a mere 492 transactions are classified as fraudulent, while the remaining transactions represent legitimate activities. The dataset has 31 columns and includes different features, such as information about time, features marked as V1–V28 (which are anonymized numerical features derived from PCA), transaction amounts, and class labels that show which transactions are fraudulent and which are not. Addressing the profound class imbalance prevalent in this dataset is crucial due to the inherent complexity caused by the disproportion between the limited occurrences of fraudulent transactions and the overwhelming majority of legitimate ones. Table 1 provides a comprehensive breakdown of the dataset's columnar structure, detailing the attributes encapsulated within each column. The highly skewed distribution between fraudulent and non-fraudulent transactions poses a significant challenge in training accurate and reliable machine learning models, necessitating careful preprocessing and specialized methodologies to address this severe class imbalance.

Table 1. Data description

Features	Description
Time	Time in seconds representing the duration between the current transaction and the first transaction
V1, V2, V3, …, V27, V28	Numerical features resulting from PCA transformation, ensuring user privacy
Amount	Transaction amount
Class Label	Binary classification labels: 1 denotes non-fraudulent transactions, 0 denotes fraudulent transactions

Following the presentation of the table describing the characteristics of the dataset and their corresponding explanations, a sequence of thorough preprocessing procedures was undertaken to enhance the dataset's suitability for analysis. The 'Amount' column was initially subjected to normalization, wherein its values were standardized to a range of [-1,1] using the StandardScaler function from the Scikit-Learn library.

In order to improve the efficiency of the dataset's attributes, the initial 'Amount' column and the subsequent 'Time' column were both removed from the dataset. In order to maintain the integrity of the data, any instances of duplicate entries present within the dataset were methodically removed. The preprocessing operations undertaken in this study played a crucial role in enhancing the dataset, hence facilitating further analyses and model creation inside the credit card fraud detection framework.

3.2. Class imbalance problem

The issue of class imbalance has a significant impact on the efficacy of predictive models in the field of credit card fraud detection. The imbalanced distribution of non-fraudulent and fraudulent transactions introduces a bias in machine learning models, leading to a tendency to reliably forecast the majority class while encountering difficulties in accurately identifying the minority class. As a result, these models demonstrate a notable level of accuracy in classifying the majority class, although they encounter challenges in achieving satisfactory precision and recall rates for detecting fraudulent transactions.

In order to rectify this disparity, many approaches are implemented. Resampling strategies, such as oversampling the minority class, undersampling the majority class, or employing a hybrid approach, specifically address this difficulty. Additionally, by lowering the classification threshold, the optimization of the decision threshold increases the accuracy of predicting the minority class. The evaluation of models using measures such as accuracy, recall, and F1-score provides a more comprehensive assessment that takes into account both groups. In machine learning, class imbalance risks biasing models and impacting their reliability. Traditional algorithms favor accuracy, leading to bias towards the majority class and hindering understanding when datasets are imbalanced [15], [16]. In credit card fraud detection, relying solely on accuracy is misleading due to an imbalance. Metrics like recall and F1-score offer better insights and prevent the accuracy paradox, where high accuracy masks poor performance in crucial areas like false negatives [17], [18]. Using diverse metrics helps evaluate models comprehensively, identifying areas needing improvement.

3.3. Deep neural network as core model

The study utilizes a deep neural network (DNN) as the fundamental component for detecting credit card fraud. The DNN, built using the Keras Sequential API, incorporates a complex structure designed to understand nuanced patterns present in transactional data. The first layer, consisting of 64 units, processes the input data, which includes 29 features. Every neuron in this tightly linked layer calculates the weighted total of its inputs, which is then sent through a rectified linear unit (ReLU) activation function [19]. Mathematically, this can be expressed as (1):

$$h1 = ReLU(W1 \cdot X + b1) \tag{1}$$

The output of the first layer $h1$ serves as the input for subsequent layers, denoted as (2).

$$h2 = ReLU(W2 \cdot h1 + b2) \tag{2}$$

where $W1$ and $W2$ represent the weights, X denotes the input data, $b1$ and $b2$ represent the bias terms, and $h1$ and $h2$ are the outputs of the respective hidden layers. Following the same pattern, there are additional hidden layers with 32 and 16 units, respectively. In order to address the issue of overfitting, dropout layers are incorporated after every hidden layer, where 30% of the units are randomly deactivated during the training process. In addition, an L2 regularization technique is employed to regulate the complexity of the model and prevent overfitting [20]. This regularization involves applying a penalty of 0.001 to the weights of each layer, as described mathematically. The term is represented by (3).

$$\lambda \sum n_{(i=1)} W_i^2 \tag{3}$$

The regularization term is defined as the sum of the squared individual weights multiplied by the regularization coefficient λ, where λ represents the regularization coefficient, indicating the strength of regularization. n is the total number of weights in the model, and Wi denotes the i^{th} weight in the model. A sigmoid function drives the final layer, which calculates the likelihood of fraud for each transaction. During the training process, the model goes through 10 epochs, where it iterates over the dataset in batches of 64 samples. It adjusts its internal parameters using the Adam optimizer to minimize the binary cross-entropy loss function and optimize the performance of classification [21]. Figure 2 provides a clear explanation of the intricacies of our deep neural network structure. In this visual representation, the many layers and design ideas of our approach are shown, offering a clear and easily understandable summary.

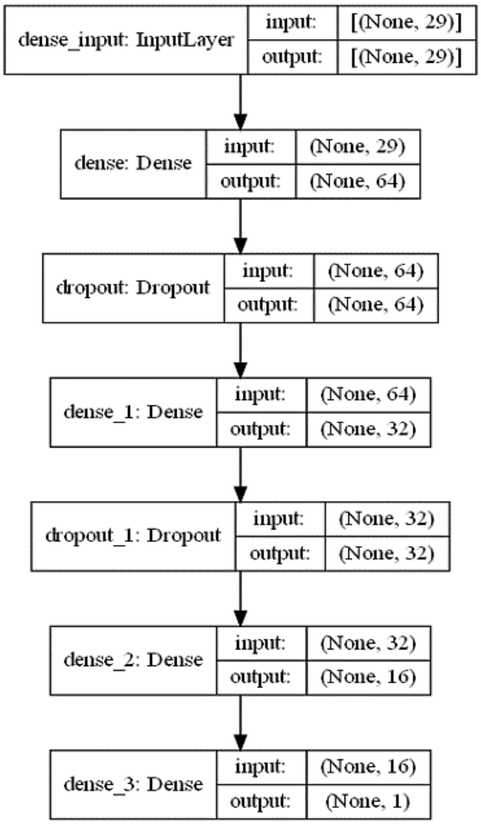

Figure 2. Our deep neural network architecture

3.4. Tackling imbalance-optimization and resampling

This section focuses on techniques aimed at rectifying the imbalance within datasets, particularly in credit card fraud detection. It explores how optimization and resampling methodologies are utilized to enhance model performance. Ensuring more accurate identification of fraudulent transactions despite the inherent imbalance between classes.

3.5. Binary cross-entropy loss

Binary cross-entropy loss is a basic function in machine learning, particularly in tasks that include binary classification, such as detecting the legality of credit card transactions. Within the field of fraud detection, the goal is to predict whether a transaction is fraudulent or not, which is a typical scenario of binary classification [22]. This loss function calculates the difference between projected probabilities and actual labels, which is crucial for adjusting the neural network's parameters (weights and biases) to improve predictive accuracy [23]. Examine a dataset of credit card transactions that is distinguished by several attributes, such as transaction amount, location, and time. Using binary cross-entropy loss helps train a neural network to sort these transactions into groups based on their features, telling the difference between transactions that are fraudulent and those that are not. The binary cross-entropy loss function is defined as (4).

$$L\,(y,\,\hat{y}) = -\left(y * log(\hat{y}) + (1-y) * log\,(1-\hat{y})\right) \tag{4}$$

where y is the true label (either 0 or 1), and $\hat{y}$ is the predicted probability of the positive class (i.e., the probability that the transaction is fraudulent).

To illustrate this concept practically, consider a dataset with credit card transactions. Our goal is to train a neural network to predict the probability of each transaction being fraudulent. Input attributes like amount, location, and time represent each transaction. We also have binary labels (0 or 1) indicating fraudulence. Training the network involves utilizing binary cross-entropy loss, where the network processes input features per transaction to predict the likelihood of fraud. The loss function measures the difference between predicted probabilities and actual labels. Based on this disparity, the network adjusts its weights and biases during training [24].

For instance, let's consider a transaction that includes input features such as an amount of 150.00 USD, a location of “Chicago”, a time of “11am”, and a true label indicating fraudulence with a value of 1. The binary cross-entropy loss for this transaction would be calculated based on the network's prediction probability of 0.8.

$$L\,(y = 1,\, \hat{y} = 0.8) = -\begin{pmatrix} 1 \,*\, log\,(0.8) \\ +\,(1\,-\,1)\,*\,log\,(1\,-\,0.8) \end{pmatrix} = -\,log\,(0.8) \approx\, 0.223 \tag{5}$$

This loss value indicates how well the network is doing at predicting the true labels for the dataset. During the training process, the network adjusts its weights and biases in order to minimize the overall binary cross-entropy loss over all transactions in the dataset. This helps to enhance its capability to forecast whether a certain transaction is fraudulent or not [25].

In summary, plain binary cross-entropy loss minimization is a technique often used in credit card fraud detection to train neural networks to identify transactions as fraudulent or not, depending on a set of input features. The loss function quantifies the disparity between the projected probabilities and the actual labels, and the network's weights and biases are adjusted accordingly using this disparity. The network enhances its predictive accuracy for identifying fraudulent transactions by minimizing the overall binary cross-entropy loss over the entire dataset.

3.6. Minimization using class weights

The challenge of class imbalance in machine learning demands effective strategies like weight minimization to address this issue. This method involves assigning weights to samples based on their class frequency in the dataset, emphasizing the significance of the minority class during model training [26], [27]. In credit card fraud detection, prioritizing accurate identification of fraud, despite its rarity, is critical. Using modified loss functions during training helps the model focus on correctly recognizing fraudulent transactions, minimizing imbalanced dataset effects, and potentially improving performance. Implementing class weights in logistic regression involves understanding data imbalance, choosing suitable weights, integrating them into the loss function, training the model, and evaluating its performance. For instance, modifying the binary cross-entropy loss function includes class weights to penalize misclassifications in the less common but important class, determined through techniques like inverse class frequency, which elevates the significance of the minority class during training [28].

Let's take an example of a dataset containing 1,000 samples, out of which 100 are classified as belonging to the minority class, representing fraudulent transactions. The determination of class weights can be achieved in the following manner: Let's denote the class weights for the majority class as w_0 and for the minority class as w_1. The class weights can be determined as (6), (7):

$$w_0 = \frac{Total\ samples}{Number\ of\ majority\ class\ samples} = \frac{1000}{900} \approx 1.11 \tag{6}$$

$$w_1 = \frac{Total\ samples}{Number\ of\ minority\ class\ samples} = \frac{1000}{100} = 10 \tag{7}$$

During the training process, these class weights can be used to adjust the contribution of each class to the overall loss function [29]. Specifically, they are applied in the calculation of the loss for each training example. This way, the model gives higher importance to the minority class, helping it learn from the relatively fewer instances of fraudulent transactions.

3.8. Undersampling techniques

Undersampling techniques are essential in addressing class imbalances in datasets, particularly in credit card fraud detection scenarios. Within these datasets, the vast majority of transactions are non-fraudulent, greatly surpassing the minor portion of fraudulent transactions. The presence of this imbalance frequently distorts machine learning models, causing them to prioritize accuracy in the dominant class. As a result, their capacity to accurately identify the minority class, namely fraudulent transactions in this case, is compromised. To tackle this imbalance, undersampling reduces instances in the majority class, aiming for a more equitable distribution between classes in the dataset [30], [31]. For instance, in a dataset where 99.83% are non-fraudulent and 0.17% are fraudulent, random undersampling selects a subset of non-fraudulent transactions equal to the fraudulent ones. This rebalancing creates a new dataset, allowing models to train more fairly on both classes. This adjustment minimizes bias toward the larger class, notably refining the model's ability to detect fraudulent transactions in credit card fraud detection [32], [33]. However, while opting for random undersampling, there's a risk of losing crucial information from excluded majority instances. Despite this limitation, it is a valuable method for our fraud detection issue, fostering a balanced dataset that enhances model training and comprehension of both class characteristics. Experimentation will gauge its effectiveness in rectifying class imbalances and improving overall model performance.

3.9. Oversampling techniques

Oversampling techniques are instrumental in addressing the issue of class imbalance within machine learning datasets, especially in scenarios where one class is significantly underrepresented compared to the other. This method primarily focuses on amplifying the minority class by either replicating existing samples or generating synthetic instances to level the class distribution [34]. Among the assortment of oversampling methods available, synthetic minority oversampling technique (SMOTE) stands out as a well-regarded algorithm for combating class imbalance. SMOTE operates by creating synthetic samples for the minority class through an interpolation process, thus diversifying the representation of this class. The procedure involves selecting an instance from the minority class and identifying its k nearest neighbors in the feature space. Subsequently, new synthetic instances are produced by interpolating between the chosen instance and its neighboring samples [35].

Mathematically, let's consider a scenario where X represents the feature space and $X_{minority}$ denotes the instances from the minority class. For each $x_i \in X_{\text{minority}}$, SMOTE identifies its k nearest neighbors. A synthetic sample x_{new} is then generated by combining x_i with a randomly chosen neighbor x_j using (8):

$$x_{new} = x_i + \lambda \times \left(x_j - x_i\right) \tag{8}$$

Here, x_i and x_j represent two feature vectors from the minority class, and λ $(0 < \lambda < 1)$ signifies a random value determining the extent of interpolation between x_i and x_j the synthetic sample x_{new} is then added to the dataset, effectively augmenting the representation of the minority class.

The benefit of SMOTE is that it can produce synthetic instances that not only replicate existing data but also add brand-new observations in areas of the feature space where the minority class is underrepresented. This process aims to rectify the imbalance, enabling machine learning models to glean a more comprehensive understanding of the minority class, subsequently improving their discriminatory abilities between classes [36]. However, it is essential to note that while SMOTE and oversampling techniques in general contribute to rectifying class imbalance, their indiscriminate application might lead to overfitting issues and potentially inflate the model's performance on the training data, thus impacting its generalization capabilities on unseen data [37]. Therefore, judicious utilization of oversampling methods, considering the specific nuances of the dataset and the machine learning problem, is crucial.

4. EVALUATION

4.1. DNN–plain binary cross-entropy loss

In order to improve the performance of our model, we utilized the Adam optimizer in conjunction with the binary cross-entropy loss function. The Adam optimizer is a method for optimizing the learning rate that combines the advantages of the AdaGrad and RMSProp algorithms. Adam, a widely recognized algorithm in the field of deep learning, has shown faster convergence rates when compared to other optimization strategies. Consequently, we selected the binary cross-entropy loss function for our work, as it is commonly used in binary classification tasks similar to ours. This function works by minimizing the discrepancy between anticipated and actual class labels, fitting well with our goal to improve model accuracy in distinguishing between two separate classes. Figure 3 is a depiction of the confusion matrix associated with the DNN trained using the plain binary cross-entropy loss.

The deep neural network, evaluated using the binary cross-entropy loss function, demonstrated strong performance in accurately predicting non-fraudulent transactions (true negatives: 82,548) with remarkable accuracy. However, when identifying fraudulent transactions (true positives: 97), the model encountered a slight challenge. While the count of (false negatives: 32) was relatively low, indicating instances where fraudulent activities went undetected, there were notable (false positives: 22), misclassifying legitimate transactions as fraudulent. This points to a need for further enhancement to minimize both false negatives and especially false positives, which are pivotal in fraud detection scenarios. The model's reliance on the binary cross-entropy loss function facilitated a focused optimization toward reducing the difference between predicted and actual class labels, although fine-tuning remains necessary to enhance its accuracy in identifying fraudulent transactions.

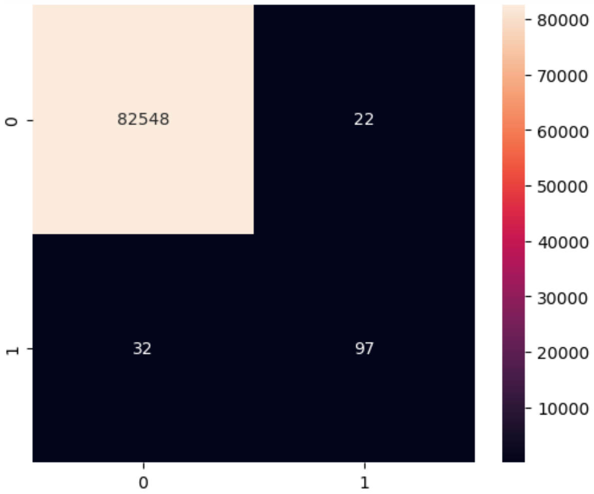

Figure 3. DNN–plain binary cross-entropy loss confusion matrix

4.2. DNN–minimization using class weights

The credit card fraud detection model, using a DNN, effectively addresses class imbalances by using a weighted loss technique. The *'class_weight'* function in scikit-learn is used to assign greater weight to the loss incurred by misclassifying fraudulent transactions. This improves the model's capability to reliably recognize minority classes, which is vital in the context of fraud detection. The function produced a dictionary containing class weights for two classes: 0 (non-fraudulent) and 1 (fraudulent). The calculated class weight for class 0 was 0.5008929498494445, but for class 1, it was 280.47093023255815. As a result, the model assigned greater importance to errors related to fraudulent transactions, hence improving its capacity to reliably identify such instances. Figure 4 is imperative to introduce as a depiction of the confusion matrix linked to the DNN trained using the minimization using weights technique.

The confusion matrix for our model evaluation demonstrates a notable performance: accurately classifying a substantial number of non-fraudulent transactions (true negatives: 81,022). However, it is accompanied by a discernible count of (false positives: 1,548), where legitimate transactions were misclassified as fraudulent. Although the model exhibited a remarkable reduction in missed fraudulent cases (false negatives: 14), correctly identifying 115 cases of fraudulent transactions (true positives), it also overlooked a few instances. This analysis underscores the model's strength in minimizing missed fraudulent activities while highlighting the necessity to further refine its precision in flagging legitimate transactions. Achieving a balanced approach between precision and recall remains imperative to enhance the model's fraud detection capabilities.

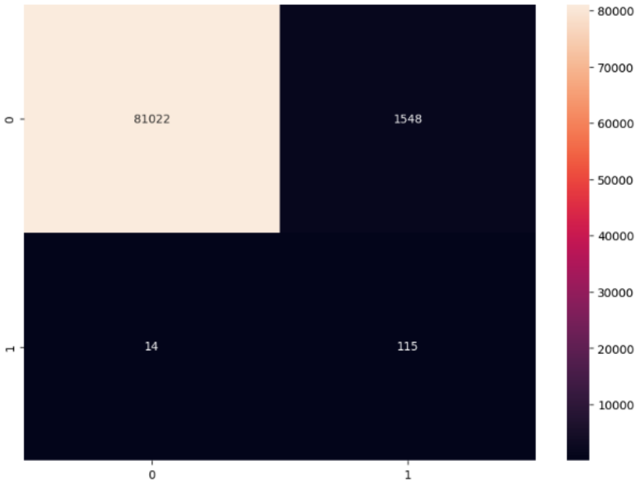

Figure 4. DNN–minimization using weights confusion matrix

4.3. DNN–random undersampling

Undersampling is an important approach used to tackle class imbalances in datasets, especially in situations such as credit card transaction analysis. Our study employed random undersampling as a strategy to address the substantial imbalance between fraudulent and non-fraudulent transactions. The study aimed to improve machine learning models by creating a more equitable dataset by randomly selecting non-fraudulent transactions and fraudulent transactions. This approach ensured a fair representation of both classes, improving classification accuracy. Random undersampling can improve classification accuracy but may lead to knowledge loss from discarded data. Despite this, it remains beneficial in situations where accurate minority class categorization is crucial. Table 2 provides a snapshot of the class distribution resulting from the application of random undersampling. This adjustment is instrumental in enhancing the model's training process and improving its accuracy in identifying both non-fraudulent and fraudulent transactions.

Figure 5 illustrates the confusion matrix related to the DNN trained using the random undersampling technique. The confusion matrix visually represents the classification performance of the model after employing random undersampling. After applying random undersampling to our DNN evaluation, the confusion matrix shows encouraging results. The model correctly classified 142 transactions as non-fraudulent (true negatives) and incorrectly classified 5 legitimate transactions as fraudulent (false positives). However, it demonstrated remarkable accuracy in detecting fraudulent transactions, properly identifying 128 cases as (true positives). Nevertheless, there were 9 instances of fraudulent transactions that were not detected (false negatives), suggesting situations where the model did not successfully recognize them as fraudulent. The results demonstrate the effectiveness of the DNN's random undersampling technique in accurately classifying non-fraudulent transactions and its ability to identify a significant number of fraudulent cases. However, additional improvement is necessary to decrease the occurrence of undetected fraudulent transactions and minimize the misidentification of valid ones.

Table 2. Class distribution after applying random undersampling

Technique	Class 0	Class 1
Random undersampling	492	492

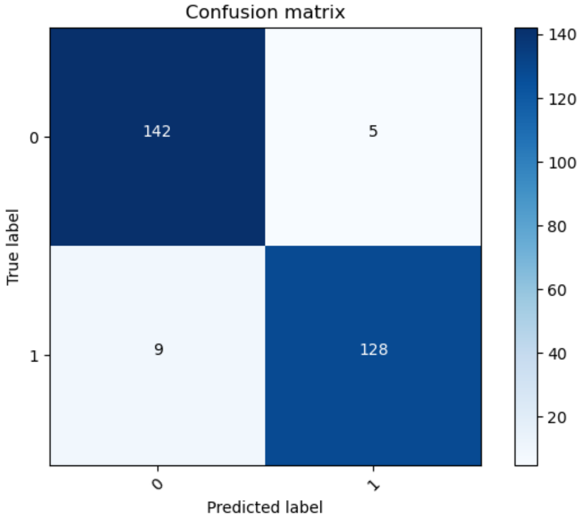

Figure 5. DNN–random undersampling confusion matrix

4.4. DNN–oversampling using SMOTE

During our investigation into credit card fraud detection, we implemented SMOTE, a widely renowned method specifically intended to handle such imbalances in datasets. SMOTE is a technique that creates new instances within the minority class, specifically fraudulent transactions, to improve the representation of these instances. This technique enhances class balance in the dataset by increasing the number of occurrences in the minority class. The goal is to enhance the model's fraud detection capabilities, minimizing the chance of missing true fraudulent activity. Table 3 illustrates the class distribution resulting from the application of SMOTE. The numerical values in the table delineate the count of instances for each class, emphasizing the effectiveness of SMOTE in augmenting the minority class and rectifying the initial imbalance.

In Figure 6, you can see the confusion matrix that is associated with the DNN that was trained using SMOTE. The result matrix showcases a highly promising outcome, especially in terms of classifying non-fraudulent transactions, with 82036 instances accurately predicted (true negatives). However, it seems that the model has not categorized any fraudulent transactions as False Negatives, showcasing a considerable improvement in detecting fraudulent cases (true positives: 82804). While the model displays exceptional capability in flagging non-fraudulent cases, it may require further analysis to confirm the absence of (false positives: 274) or misclassifications of genuine transactions as fraudulent. This result reflects a notably robust performance in identifying fraudulent activities, suggesting that the SMOTE technique has significantly contributed to enhancing the model's sensitivity to detecting such cases without any false negatives.

Table 3. Class distribution after applying SMOTE

Technique	Class 0	Class 1
SMOTE	284315	284315

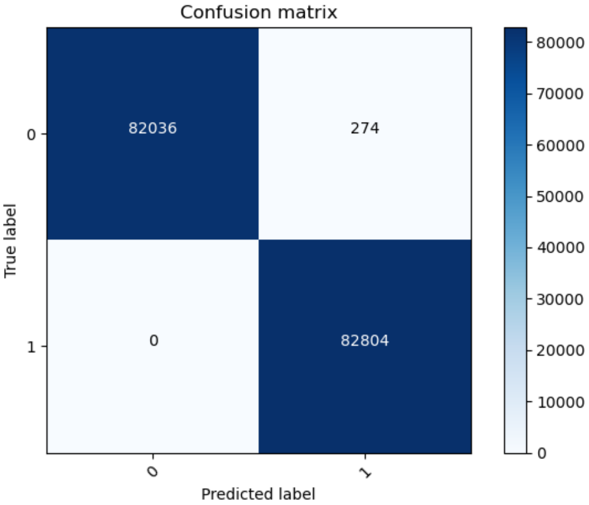

Figure 6. DNN–SMOTE confusion matrix

5. RESULTS AND DISCUSSION

The results from evaluating four different models: the plain binary cross-entropy loss deep neural network, weighted deep neural network, random undersampled deep neural network, and SMOTE deep neural network are detailed in Table 4. Highlighting various evaluation metrics: accuracy, false negative rate, recall, precision, and F1-score. The plain binary cross-entropy loss DNN exhibited an outstanding accuracy of 99.93%. However, it demonstrated a relatively higher false negative rate of 24.81%, implying an inability to identify approximately 25% of the fraudulent transactions. Despite this, its recall value of 75.19% indicates its capacity to recognize actual positive instances, and a Precision of 81.51% demonstrates its ability to label accurate positive predictions. The model's F1-score of 78.23% reflects a balanced performance between precision and recall. The weighted DNN achieved a slightly lower accuracy of 98.11% but significantly reduced the false negative rate to 10.85%, enhancing its capability to identify fraudulent transactions. However, the model's precision of 6.92% was notably low, resulting in a higher rate of false positives.

The random UnderSampled DNN achieved an accuracy of 95.07% and notably minimized the false negative rate to 6.57%, indicating its adeptness in identifying fraudulent transactions. Its high recall of 93.43% and precision of 96.24% underscore its ability to accurately identify true positives while maintaining a strong precision rate. The SMOTE DNN achieved an impressive accuracy of 99.83% and exhibited zero false negatives, signifying its ability to detect all fraudulent transactions. With a perfect recall of 100.00%, it excelled in identifying all positive instances. The model's precision of 99.67% and F1-score of 99.83% suggest a minute number of false positives.

In short, the Random UnderSampled and SMOTE DNNs demonstrated superior performance compared to plain and weighted models in the detection of fraud. Although the undersampled model reduced the occurrence of false negatives, the SMOTE model produced a flawless recall, albeit with a slightly higher number of false positives. Both techniques are highly efficient in detecting fraudulent activities, each with distinct advantages.

Table 5 presents a comparative analysis of crucial performance metrics, highlighting the supremacy of our model over its existing machine learning counterparts: logistic regression-sag, artificial neural network (ANN), and RFC undersampling. Our model attains an exceptional accuracy of 99.83%, surpassing others, including RFC undersampling (97.43%), logistic regression-sag (99.73%), and ANN (96.00%) [7], [9], [10]. Boasting a perfect recall of 1.0000, our model excels in identifying all instances of fraudulent transactions, outperforming alternative models. The notable precision of 99.67% and F1-score of 99.83% further emphasize its efficacy in achieving both accuracy and sensitivity. In conclusion, the amalgamation of deep neural networks with the SMOTE oversampling technique proves to be a robust approach for handling imbalanced datasets in credit card fraud detection, showcasing the potential for more accurate and reliable fraud detection systems in real-world applications.

Table 4. Comparison of key performance metrics across different approaches

Model	Accuracy	FalseNegRate	Recall	Precision	F1-score
PLAIN BINARY CROSS-ENTROPY LOSS	0.999347	0.248062	0.751938	0.815126	0.782258
MINIMIZATION USING CLASS WEIGHTS	0.981112	0.108527	0.891473	0.069152	0.128348
RANDOM UNDERSAMPLING	0.950704	0.065693	0.934307	0.962406	0.948148
OVERSAMPLING USING SMOTE	0.998341	0.00000	1.00000	0.996702	0.998348

Table 5. Comparison of key performance metrics with existing ML models

Model	RFC undersampling [7]	Logistic regression-sag [9]	ANN [10]	Our DNN-SMOTE
Accuracy	0.9743	0.9973	0.9600	0.9983
Recall	1.0000	0.8252	0.4700	1.0000
Precision	0.0060	0.3755	-	0.9967
F1-score	0.1182	-	-	0.9983

6. CONCLUSION

As we strive to enhance credit card fraud detection, we face the difficulty of dealing with imbalanced datasets. Using a range of techniques, such as optimization and resampling procedures, our main objective is to correct the inherent imbalance in credit card transaction data. The inherent bias towards non-fraudulent transactions presents a challenging obstacle for models to effectively identify fraudulent activities. The objective of our mission is to rebalance this disparity by enabling algorithms to accurately identify infrequent fraudulent transactions within a large volume of genuine activity. Our commitment to enhancing fraud detection capabilities is demonstrated by our use of advanced approaches like weighted optimization, undersampling, and SMOTE oversampling to handle the complexities of the dataset. Our findings demonstrate that SMOTE oversampling achieves an impressive accuracy of 99.83% without any false negatives. Although alternate techniques such as undersampling resulted in a reduction of false positives, the problem of false negatives continued to persist. In general, oversampling was found to be the most efficient approach. As we advance, promising breakthroughs such as ensemble models and sophisticated neural networks continue to shape progress in credit card fraud detection. In considering future perspectives, it is imperative to embark on exploring alternative datasets and rigorously evaluating the proposed model's robustness and generalizability. Recognizing the integral role of continuous evolution in effective fraud prevention, a flexible strategy becomes crucial, necessitating regular updates to models and the implementation of adaptive defenses. The integration of state-of-the-art machine learning methods with proactive measures empowers organizations to safeguard customers and enterprises against evolving financial risks. Concurrently, our current focus involves identifying the most significant features contributing to credit card fraud, a pivotal step in fortifying the resilience of our model against emerging threats.

REFERENCES

[1] R. Dubey, J. Zhou, Y. Wang, P. M. Thompson, and J. Ye, “Analysis of sampling techniques for imbalanced data: An n=648 ADNI study,” *NeuroImage*, vol. 87, pp. 220–241, Feb. 2014, doi: 10.1016/j.neuroimage.2013.10.005.

[2] S. Bagui and K. Li, “Resampling imbalanced data for network intrusion detection datasets,” *Journal of Big Data*, vol. 8, no. 1, Jan. 2021, doi: 10.1186/s40537-020-00390-x.

[3] M. Yeung, E. Sala, C. B. Schönlieb, and L. Rundo, “Unified focal loss: generalising dice and cross entropy-based losses to handle

class imbalanced medical image segmentation," *Computerized Medical Imaging and Graphics*, vol. 95, p. 102026, Jan. 2022, doi: 10.1016/j.compmedimag.2021.102026.

[4] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, "Learning from imbalanced data sets with weighted cross-entropy function," *Neural Processing Letters*, vol. 50, no. 2, pp. 1937–1949, Jan. 2019, doi: 10.1007/s11063-018-09977-1.

[5] N. Rachburee and W. Punlumjeak, "Oversampling technique in student performance classification from engineering course," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3567–3574, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3567-3574.

[6] S. Warghade, S. Desai, and V. Patil, "Credit card fraud detection from imbalanced dataset using machine learning algorithm," *International Journal of Computer Trends and Technology*, vol. 68, no. 3, pp. 22–28, Mar. 2020, doi: 10.14445/22312803/ijctt-v68i3p105.

[7] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," in *Proceedings - 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing, UCC 2020*, Dec. 2020, pp. 426–433, doi: 10.1109/UCC48980.2020.00067.

[8] S. A. Ebiaredoh-Mienye, E. Esenogho, and T. G. Swart, "Artificial neural network technique for improving prediction of credit card default: A stacked sparse autoencoder approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4392–4402, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4392-4402.

[9] N. Adityasundar, T. SaiAbhigna, B. Lakshman, D. Phaneendra, and N. MohanKumar, "Credit card fraud detection using machine learning classification algorithms over highly imbalanced data," *Journal of Science and Technology*, vol. 5, no. 3, pp. 138–146, May 2020, doi: 10.46243/jst.2020.v5.i3.pp138-146.

[10] 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, 2019, doi: 10.1109/ACCESS.2019.2927266.

[11] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 34, no. 4, pp. 571–598, Apr. 2022, doi: 10.1080/0952813X.2021.1907795.

[12] S. Islam, M. M. Haque, and A. N. M. Rezaul Karim, "A rule-based machine learning model for financial fraud detection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, p. 759, Feb. 2024, doi: 10.11591/ijece.v14i1.pp759-771.

[13] B. Baesens, S. Höppner, I. Ortner, and T. Verdonck, "robROSE: a robust approach for dealing with imbalanced data in fraud detection," *Statistical Methods and Applications*, vol. 30, no. 3, pp. 841–861, Jun. 2021, doi: 10.1007/s10260-021-00573-7.

[14] Machine Learning Group-ULB, "Credit card fraud detection," *Kaggle*, 2018. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (acccessed Dec 22, 2023).

[15] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, Mar. 2019, doi: 10.1186/s40537-019-0192-5.

[16] H. Ali, M. N. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 3, pp. 1552–1563, Jun. 2019, doi: 10.11591/ijeecs.v14.i3.pp1552-1563.

[17] F. J. Valverde-Albacete and C. Peláez-Moreno, "100% classification accuracy considered harmful: the normalized information transfer factor explains the accuracy paradox," *PLoS ONE*, vol. 9, no. 1, Jan. 2014, doi: 10.1371/journal.pone.0084217.

[18] M. F. Uddin, "Addressing accuracy paradox using enhanched weighted performance metric in machine learning," in *ITT 2019 - Information Technology Trends: Emerging Technologies Blockchain and IoT*, Nov. 2019, pp. 319–324, doi: 10.1109/ITT48889.2019.9075071.

[19] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, "Explaining deep neural networks and beyond: a review of methods and applications," in *Proceedings of the IEEE*, Mar. 2021, vol. 109, no. 3, pp. 247–278, doi: 10.1109/JPROC.2021.3060483.

[20] K. Farhana, M. Rahman, and M. Tofael Ahmed, "An intrusion detection system for packet and flow based networks using deep neural network approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5514–5525, Oct. 2020, doi: 10.11591/IJECE.V10I5.PP5514-5525.

[21] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, pp. 1134–1141, Feb. 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.

[22] R. Malhotra, A. Shakya, R. Ranjan, and R. Banshi, "Software defect prediction using binary particle swarm optimization with binary cross entropy as the fitness function," *Journal of Physics: Conference Series*, vol. 1767, no. 1, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012003.

[23] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

[24] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Deconstructing cross-entropy for probabilistic binary classifiers," *Entropy*, vol. 20, no. 3, p. 208, Mar. 2018, doi: 10.3390/e20030208.

[25]. Usha Ruby Dr.A, "Binary cross entropy with deep learning technique for Image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020, doi: 10.30534/ijatcse/2020/175942020.

[26] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan, "An approach for classification of highly imbalanced data using weighting and undersampling," *Amino Acids*, vol. 39, no. 5, pp. 1385–1391, Apr. 2010, doi: 10.1007/s00726-010-0595-2.

[27] M. Du *et al.*, "A skew-sensitive evaluation framework for imbalanced data classification," *arXiv:2010.05995*, Oct. 2020.

[28] J. He and M. X. Cheng, "Weighting methods for rare event identification from imbalanced datasets," *Frontiers in Big Data*, vol. 4, Dec. 2021, doi: 10.3389/fdata.2021.715320.

[29] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2940–2951, Jul. 2022, doi: 10.1109/TNNLS.2020.3047335.

[30] M. A. Arefeen, S. T. Nimi, and M. S. Rahman, "Neural network-based undersampling techniques," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 1111–1120, Feb. 2022, doi: 10.1109/TSMC.2020.3016283.

[31] S. Susan and A. Kumar, "The balancing trick: optimized sampling of imbalanced datasets—a brief survey of the recent state of the art," *Engineering Reports*, vol. 3, no. 4, Oct. 2021, doi: 10.1002/eng2.12298.

[32] J. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "The effects of random undersampling for big data medicare fraud detection," in *Proceedings - 16th IEEE International Conference on Service-Oriented System Engineering*, Aug. 2022,

pp. 141–146, doi: 10.1109/SOSE55356.2022.00023.

[33] C. C. Tusell-Rey, O. Camacho-Nieto, C. Yáñez-Márquez, and Y. Villuendas-Rey, "Customized instance random undersampling to increase knowledge management for multiclass imbalanced data classification," *Sustainability*, vol. 14, no. 21, Nov. 2022, doi: 10.3390/su142114398.

[34] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: using deep learning and SMOTE for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023, doi: 10.1109/TNNLS.2021.3136503.

[35] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, "Stop oversampling for class imbalance learning: a review," *IEEE Access*, vol. 10, pp. 47643–47660, 2022, doi: 10.1109/ACCESS.2022.3169512.

[36] B. S. Raghuwanshi and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowledge-Based Systems*, vol. 187, Jan. 2020, doi: 10.1016/j.knosys.2019.06.022.

[37] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of classification methods on unbalanced data sets," *IEEE Access*, vol. 9, pp. 64606–64628, 2021, doi: 10.1109/ACCESS.2021.3074243.

BIOGRAPHIES OF AUTHORS

Oussama Ndama is a Ph.D. student in data science, artificial intelligence and smart systems research team (DSAI2S), C3S Laboratory, Faculty of Sciences and Technologies (FST), Tangier, Morocco. He had his master in computer science and big data, Laureate of FST of Tangier. He is also a business intelligence Engineer with more than 5 years of experience in different multinational companies. The research topics of interest are smart systems, machine learning, deep learning, NLP, ANN, sentiment analysis, and smart cities. He can be contacted at email: oussama.ndama@etu.uae.ac.ma.

Ismail Bensassi is a Ph.D. student in data science, artificial intelligence and smart systems research team (DSAI2S), C3S Laboratory, Faculty of Sciences and Technologies (FST), Tangier, Morocco. He is an engineer in computer science, Laureate of FST of Tangier. The research topics of interest are smart connection of user profiles in a big data context, multi-agent systems (MAS), case-based reasoning (CBR), ontology, machine learning, smart cities, and e-learning, MOOC/SPOC. He can be contacted at: bensassi.ismail@gmail.com.

El Mokhtar En-Naimi is a full professor in the University of Abdelmalek Essaâdi (UAE), Faculty of Sciences and Technologies of Tangier (FSTT), Department of Computer Sciences. (He was temporary professor: from 1999 to 2003 and permanent professor: since 2003/2004 until now. Actually, He is a full professor in UAE, FST of Tangier). He was a head of Computer Sciences Department, since October 2016 until the end of December 2020. He was responsible for a License of Science and Technology, LST Computer Engineering ("Licence LST-GI"), from January 2012 to October 2016. He is a chief of data science, artificial intelligence and smart systems (DSAI2S) research team since the academic year 2022/2023. He is also a founding member of the Both Laboratories: Laboratoire d'Informatique, Systèmes et Télécommunications (LIST) Laboratory (From 2008 To 2022) and Computer Science and Smart Systems (C3S) Laboratory since the academic year 2022/2023 until Now, the University of Abdelmalek Essaâdi, FST of Tangier, Morocco. He is also an expert evaluator with the ANEAQ, since the academic year 2016/2017 until now, that an expert of the private establishments belonging to the territory of the UAE and also an expert of the Initial or Fundamental Formations and Formations Continuous at the Ministry of Higher Education, Scientific Research and Executive Training and also at the UAE University and the FST Tangier since 2012/2013 until Now. He is an author/co-authors of several articles, published in the international journals in computer sciences, in particular, in multi-agent systems (MAS), cases-based reasoning (CBR), artificial intelligent (AI), machine learning (ML), deep learning (DL), e-learning, MOOC/SPOC, big data, data-mining, wireless sensor network, VANet, MANet, smart city. He was/is also director of several doctoral theses in computer sciences. He has too served as a general chair, technical program chair, technical program committee member, organizing committee member, session chair, and reviewer for many international conferences and workshops. In addition, he is an associate member of the Institute of Complex Systems in Normandy (ISCN), the University of the Havre, France, since 2009 until Now. He can be contacted at email: en-naimi@uae.ac.ma.