Machine Learning Facing Behavioral Noise Problem in an Imbalanced Data Using One Side Behavioral Noise Reduction: Application to a Fraud Detection

Jamal Malki, Salma El-Hajjami, Alain Bouju, Mohammed Berrada

Machine learning facing behavioral noise problem in an imbalanced data using One Side Behavioral Noise Reduction. Application to a fraud detection

1st Salma El Hajjami
IASSE Laboratory
ENSA, USMBA
Fez, Morocco
salma.elhajjami@usmba.ac.ma

2nd Jamal Malki
L3i Laboratory
La Rochelle University
La Rochelle, France
jamal.malki@univ-lr.fr

3rd Alain Bouju
L3i Laboratory
La Rochelle University
La Rochelle, France
alain.bouju@univ-lr.fr

4th Mohammed Berrada
IASSE Laboratory
ENSA, USMBA
Fez, Morocco
mohammed.berrada@gmail.com

***Abstract*—With the expansion of machine learning and data mining in the context of Big Data analytics, the common problem that affects data is class imbalance. It refers to an imbalanced distribution of instances belonging to each class. This problem is present in many real world applications such as fraud detection, network intrusion detection, medical diagnostics, etc. In these cases, data instances labeled negatively are significantly more numerous than the instances labeled positively. When this difference is too large, the learning system may face difficulty when tackling this problem, since it is initially designed to work in relatively balanced class distribution scenarios. Another important problem, which usually accompanies these imbalanced data, is the overlapping instances between the two classes. It's commonly referred as noise or overlapping data. In this article, we propose a novel approach called: One Side Behavioral Noise Reduction (OSBNR). This approach presents a new way to deal with the problem of class imbalance in the presence of a high noise level. OSBNR is based on two steps. Firstly, a cluster analysis is applied to groups similar instances from the minority class into several behavior clusters. Secondly, we select and eliminate the instances of the majority class, considered as behavioral noise, which overlap with behavior clusters of the minority class. The results of experiments carried out on a representative public dataset confirm that the proposed approach is efficient for the treatment of class imbalance in the presence of noise.**

***Index Terms*—Machine learning, Imbalanced data, Data mining, Big data.**

I. Introduction

The problems belong to the class of anomaly detection such as fraud detection, network intrusion detection and medical diagnostics, share a common observation: the captured data is imbalanced. In other words, one of the classes, called minority or postive class, is strongly under-represented compared to the other class, called majority or negative class [12], [2]. The problem with imbalanced data sets is that standard classification learning algorithms assume a relatively uniform distribution of classes. They are often biased towards the majority class and, therefore, there is a higher rate of classification errors for minority class instances [7].

PLAIBDE Project. Programme Opérationnel FEDER-FSE de la région Nouvelle Aquitaine - France, 2017-2020

Unfortunately, the minority class is generally the most important from data analysis point of view. Thus, such a poor classification of minority instances often has serious consequences in real applications. For example, in credit card fraud detection systems, undetected fraudulent transactions are much more serious and costly than detecting normal behavior as fraud. Finding a good solution for imbalanced data with good accuracy has become an important area of research, known as learning from imbalanced data [2], [13], [1].

An important problem, which usually accompanies the imbalanced data classification, is the presence of noise. It is commonly referred as overlapping data and we named it as behavioral noise. The behavioral noise problem occurs when the data instances belonging to a class overlap the data instances of another class. Therefore, the boundaries of the classes may not be clearly defined. This problem plays an important role in the study of imbalanced data. Most classification learning algorithms could lead to classifying minority class instances into majority ones. Thus, classification performance depends on these two main problems: class imbalance and behavioral noise [20].

In this work, we are interested of credit card fraud detection problem. This is a good example of a very imbalanced and overlapping data classification problem [7]. We present a new approach called: One Side Behavioral Noise Reduction (OSBNR) to deal with class imbalance problem, with a particular emphasis on the presence of noise. OSBNR mainly contains two steps. First, a cluster analysis is applied to groups similar instances of the minority class dataset in multiple behavior clusters. Second, we select and eliminate instances of the majority class, considered as behavioral noise, which overlap the behavioral clusters of the minority class.

This article is organized as follows: In Section II, common approaches to address the data imbalance are presented. Section III presents in detail our proposed approach to improve the classification of imbalanced data. While section IV reports the experiments and shows the results obtained. The final section concludes the paper and provides our insights into future works.

II. Common approaches for addressing data imbalance

Different approaches have been proposed to deal with imbalanced data problem and improve the performance of prediction [23], [24], [21]. Often, these approaches are based on either the data level or the algorithm level approaches. Data-level based approaches are independent of the classifier and use sampling methods to produce a well balanced dataset from the imbalanced training dataset. With regard to sampling, we can distinguish two methods: undersampling and oversampling [16], [18], [28], [8]. Algorithm-level approach adapts existing classification learning algorithms to guide learning towards minority class. It requires a given specific knowledge of the corresponding classifier and of the application field[22], [5], [6], [9].

In reviewing the literature, related resampling methods is the main focus in this paper. A popular algorithm for undersampling, Condensed Nearest Neighbour Rule (CNN) was proposed in the paper [10]. CNN works by eliminating the majority class samples that are distant from the decision border since these samples can be considered as less relevant for learning.

Another popular algorithm for undersampling, Tomek's Link removal (TL) was introduced in [25], This algorithm works by detecting pair of data points, called Tomek's Link, that are each other's nearest neighbor but have different class labels. Undersampling can be done by either removing all Tomek links or by removing the majority class data belonging to the Tomek link.

In [27], Edited Nearest Neighbor Rule (ENN) was presented. It removes any instance whose class label is different from the class of at least two of its three nearest neighbors. The idea behind this technique is to remove the instances from the majority class near or around the borderline of different classes, in order to increase classification accuracy of minority instances rather than majority instances.

Another undersampling technique called Neighbourhood Cleaning Rule (NCR) was proposed by [15]. It uses Wilson's Edited Nearest Neighbour Rule (ENN) [27] to remove instances from the majority class when two out of three of the nearest neighbors of an instance contradict the class.

Two improvements to ENN are proposed in [26]: Repeated Edited Nearest Neighnor (RENN) and All-KNN (AKNN). Both methods make multiple passes over the training set repeating ENN. RENN just repeats the ENN algorithm until no further eliminations can be made from the edited set. AKNN repeats ENN for each sample using incrementation values of k each time and removing the sample if its label is not the predominant one at least for one value of k.

In [14] an undersampling approach called One-Side Selection (OSS) is proposed which is combination of Tomek's Link [25] followed by the application of CNN. Tomek's Link is used to remove noisy and borderline majority class examples. Then, CNN will remove example from the majority class that are distant from decision border. Then a consistent subset of the majority class is formed.

III. Proposed Approach

Most classification learning algorithms are often biased toward the majority class due to the data imbalance distribution, which leads to a higher misclassification rate for the minority class instances [7]. Unfortunately, the minority class is usually the most important from a data analysis perspective. So, such misclassification of minority instances often has serious consequences in real applications. Another critical factor in real world imbalanced data concerns presence of a relatively large number of noise instances from the majority class located inside the minority class. This problem is commonly referred as overlapping data and we named it as behavioral noise. The behavioral noise implies that some instances of a class have similar characteristics to those of a different class. The presence of noise has a severe impact in learning problems. The generated models can become more complex, showing less generalization abilities, lower precision, and higher computational cost [29]. So, the classification performance depends on these two main problems: class imbalance and behavioral noise.

In this work, we focus on credit card fraud detection as a very imbalanced and overlapped data classification problem, where non-fraudulent samples are much more numerous than fraud samples [7]. As well, it is known that some users share a behavior where the transactions are similar. While the transaction behavior of some users resembles no behavior and may even behave like transactions unlike their labels. For example, if a user's credit card information is stolen, fraudsters will make several large transactions in a short time to maximize the benefits, while some normal users may also make large transactions in a short time for certain reasons. The normal behavior of an individual user is therefore close to fraud. We define this type of transaction as behavioral noise. The existence of behavioral noises pushes the system to judge certain fraudulent transactions as authentic transactions. This leads to an erroneous classification which can be costly. Undetected fraudulent transactions (false positive) are much more serious and costly than detecting normal behavior as fraud (false negative). The cost of false positives is financial in nature, it varies according to the amount of the transaction. On the other hand, the cost of false negatives is measured in terms of customer dissatisfaction, and this latter can be resolved by strategies to compensate and retain customers.

The main objective of our approach, called: One Side Behavioral Noise Reduction (OSBNR), is to handle behavioral noise to improve the classification of the minority class instances. OSBNR consists of separating normal transactions (majority class instances) from fraudulent ones (minority class instances). Then, a cluster analysis is applied to group similar instances of the minority class containing the fraudulent transactions in several subsets, that form several behavior groups. The second step eliminates normal transactions behaviors, considered as behavioral noise, which overlap with the fraudulent transactions behaviors.

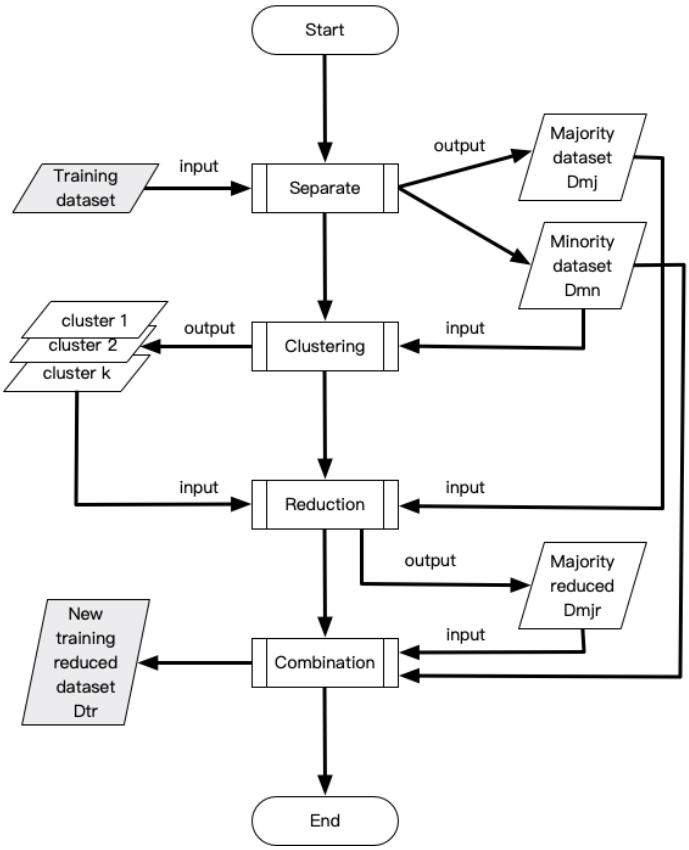

Fig. 1: Flowchart of the OSBNR approach

Figure 1 shows the main steps of OSBNR approach:

1) Separate: separates the majority Dmj and minority Dmn from the original training dataset.
2) Clustering: using k-means clustering algorithm [3] to form samples of similar Dmn instances into a number of behavior clusters. Each cluster seems to have distinct characteristics in the high-dimensional features space.
3) Reduction: carries out the behavioral noise reduction of the majority instances with those of the minority class. The Euclidean distance is used to measure the level of similarity between the different clusters centers of minority class and the majority class instances. So first, we calculate the furthest distance $dmax_i, 1 < i < k$, from minority instances to the cluster center $Cmin_i$ for each minority cluster according. Second, the distances $dmaj_{ij}$ between majority instances and different clusters centres of minority class C_i are obtained. As results, all majority instances in the minority clusters area are identified if $dmax_i \geq dmaj_{ij}$. So, they are considered as noisy instances and then eliminated.
4) Combination: we combine the reduced majority instances set $Dmjr$ with the minority instances set Dmn to have a new training dataset Dtr.

Accurate identification and elimination of these instances maximize the visibility of the minority class instances and at the same time minimize excessive elimination of data.

IV. Experiments and Results

A. Dataset Description

For this work, we use the Kaggle credit card fraud detection dataset [11]. It contains transactions made by credit card during two days of September 2013 by European card holders. Table I provides statistics for the dataset and shows that the minority class (fraud) accounts for 0.172% of all transactions. Therefore, this dataset is highly imbalanced [7]. It contains 31 numerical features. Since some of the input features contains financial information, the PCA transformation of 28 digital input features (named $V_1, \ldots, V_{28}$) were performed due to confidentiality issues. Three of the given features weren't transformed. Time feature shows the time between first transaction and every other transaction in the dataset. Amount feature is the amount's value spent in a single transaction made by credit card. Class feature represents the label, and takes only 2 values: value 1 in case of fraud transaction and 0 otherwise.

TABLE I: Kaggle credit card fraud dataset details

Transactions	Majority class	Minority class	Columns
284 807	284 315	492	31

B. Feature selection

Feature selection is a fundamental technique that selects the most relevant features from the given dataset. Choosing the right features wisely and removing the less important ones can reduce over-learning, improve accuracy, and reduce training time. Visualization techniques can be helpful in this process. Formally, we select a subset of features or attributes from the set of features and eliminate redundant features that do not contribute to performance. Thus, a feature is important when its data distribution of the two classes are divergent. Therefore, this feature can potentially separate the two classes and improve prediction performance.

Figure 2 shows the class distribution for some features of our dataset. We can see for V_9, V_{10}, V_{11}, V_{12} and V_{14} a significant divergence of class distribution. They are therefore features with a strong predictive power. So, we can keep them during the models construction. Similarly, we can see for feature V_{13} that the distribution of normal transactions (majority class) corresponds to the distribution of fraudulent transactions (minority class). This feature cannot effectively contribute to the separation between the two classes. We carried out this process for all 28 features. As a result, 11 relevant features were selected for our experiments: V_3, V_4, V_9, V_{10}, V_{11}, V_{12}, V_{14}, V_{16}, V_{17}, V_{18} and V_{19}.

C. Classifiers and resampling techniques

In this work, we applied various resampling techniques such as the Condensed Nearest Neighbour Rule (CNN) [10], Tomek's links (TL) [25], One-Side Selection (OSS) [14], Edited Nearest Neighbour Rule (ENN) [27], Repeated Edited Nearest Neighnor (RENN) [26], All-KNN (AKNN) [26] and

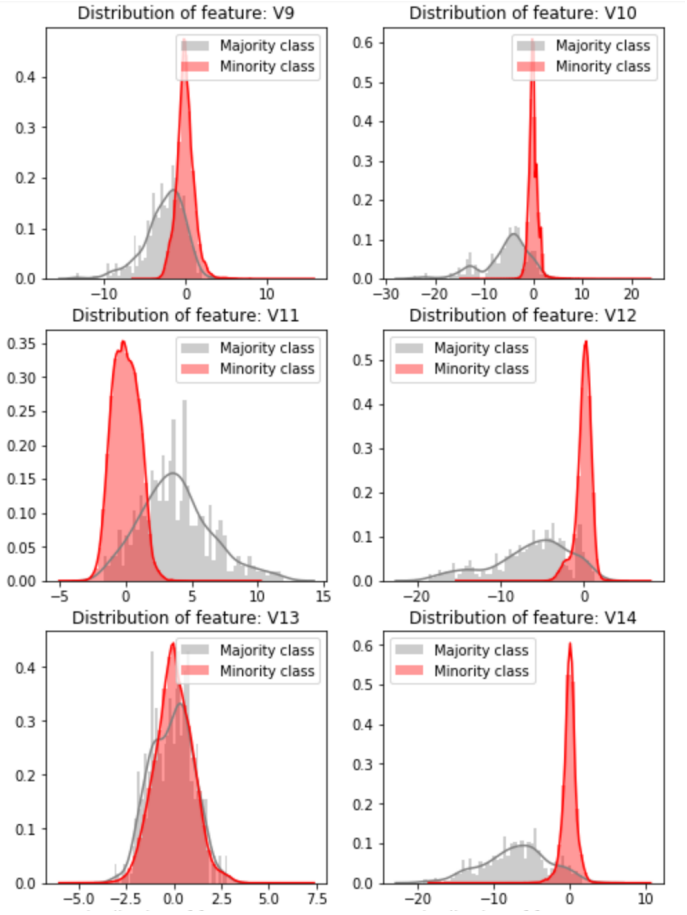

Fig. 2: Class distribution histogram on some features

Neighbor Cleaning Rule (NCR) [15]. We evaluated their performance with the proposed approach OSBNR using the best and widely used classifiers: Random Forest (RF) and Multilayer Perceptron (MLP) [19]:

- Random forest (RF) is an algorithm that consist of many decision trees. This algorithm works best when there are more trees in the forest. Each decision tree in the forest gives results. These results are merged in order to obtain a more precise and stable prediction [4].
- Multilayer perceptron (MLP) is an artificial neural network with direct action which is made up of at least 3 layers of nodes: entry layer, hidden layer and exit layer. Each node uses an activation function. The activation function calculates the weighted sum of its inputs and adds a bias. This allows us to decide which neuron should be removed and not taken into account in the external connections.

RF and MLP models parameters were determined from various preliminary tests carried out on the training data, as shown in Table II.

D. Evaluation Metrics

Evaluation metrics play an important role to assess and guide learning algorithms [23]. The common metric used is

TABLE II: RF and MLP parameters used

Classifiers	Parameter
Random forest (RF)	Number of trees = 20 Depth of each tree = 8 Impurity = Gini
Multilayer perceptron (MLP)	Number of iterations = 100 Tolerance parameter = 1e-6

accuracy. However, accuracy is not a good indicator of the actual classification performance when the class distribution is not uniform, especially for the positive (minority) class. Indeed, because it has less effect on accuracy compared to the negative (majority) class. As in [17], we consider other metrics summarized as follows, where:

$$\begin{cases} FP & \text{false positive} \\ FN & \text{false negative} \\ TP & \text{true positive} \\ TN & \text{true negative} \end{cases}$$

- Precision or Positive Predictive Value (1): represents the proportion of positive samples that were correctly classified to the total number of positive predicted samples.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- True Positive Rate (2): called Sensitivity or Recall, is the number of actual positives which are predicted positives.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- F-measure or F1-score (3): represents the harmonic mean of precision and recall. The value ranges from 0 to 1, if the value is high then F-measure indicates high classification performance.

$$F1\text{-score} = 2\frac{Precision * Recall}{Precision + Recall} \tag{3}$$

- AUC (4): represents the ability to distinguish classes, which considers both the true positive rate TPR (2) and the false positive rate FPR (5). AUC is based on the consideration that the higher the true positive rate TPR, and the lower false positive rate FPR, classification performance is better.

$$AUC = \frac{1 + TPR - FPR}{2} \tag{4}$$

Where False Positive Rate (FPR (5)) represents the proportion of legitimate samples that were wrongly predicted as fraud.

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

V. Results Analysis

A. Training and test datasets used for the experiments

We present different experiments to compare the performance of our proposed OSBNR approach and the state-of-art resampling methods (CNN, ENN, AKNN, RENN, TL, OSS,

and NCR). As there is no rule-of-thumb for how to divide a dataset into training and test sets, we have noticed that in the case of the $70/30$ rule, the percentages of the minority class of the training and test sets are: 80%, 20% respectively of the total fraud. In order to demonstrate the effectiveness of the proposed OSBNR method to deal with the imbalance and overlapping between classes problem, we studied 3 different divisions of the dataset by resampling the minority class based on the ratios: $80/20, 70/30, 60/40$, and the majority class based on the $70/30$ ratio. Table III presents training and test datasets used as input of our OSBNR approach (Fig. 1).

TABLE III: Training and test datasets used for the experiments

		Total	Majority class	Minority class
100%	**Dataset**	284 807	284315 - 100%	492 - 100%
Rule 80/20	**Training**	199 413	199020 - 70%	393 - **80%**
	Test	85 394	85295 - 30%	99 - **20%**
Rule 70/30	**Training**	199 364	199020 - 70%	344 - **70%**
	Test	85 443	85295 - 30%	148 - **30%**
Rule 60/40	**Training**	199 315	199020 - 70%	295 - **60%**
	Test	85 492	85295 - 30%	197 - **40%**

B. Performance study of the OSBNR: case of all features

In this section, we analyze the impact of the proposed OSBNR approach on the performance of each classifier by comparing it with existing resampling methods taking into account all the features and according to 3 different divisions of the dataset. The results are calculated for four metrics: AUC, Precision, Recall and F1-score.

Figures 3 and 4 show the results using the AUC metric for the RF and MLP classifiers respectively. The most interesting observation is that the proposed OSBNR offers significantly better performance compared to the other methods for the two classifiers from the AUC point of view for all the distributions of training and test sets. For the RF classifier, the best score is obtained by RF_OSBNR with an AUC value of ($AUC = 0.9341$), RF_CNN takes second place with a score of ($AUC = 0.9079$), when the training and test sets are set to $80/20$ rule. For the MLP classifier, the best AUC score rule is obtained by MLP_OSBNR with a value of ($AUC = 0.9487$), followed by MLP_OSS with a score of ($AUC = 0.9079$) when the training and test sets are set to $80/20$ rule.

Similarly, Figures 5 and 6 present the results in terms of precision. The results illustrated in Figure 5 show that OSBNR outperforms the other resampling methods for all the distributions of the training and test sets. The best precision score for the RF classifier is obtained by the OSBNR approach when the training and test sets are set at 80% and 20% fraud with a score of ($Precision = 0.8686$), while RF_CNN takes second place with a score of ($Precision = 0.8163$). Similar in MLP, as illustrated in Figure 6, it is clear that the best precision score is obtained by MLP_OSBNR with a score of ($Precision = 0.8979$), followed by MLP_OSS with a score of ($Precision = 0.8511$). Based on these results, we conclude that the proposed OSBNR can significantly improve the recognition rate of minority samples.

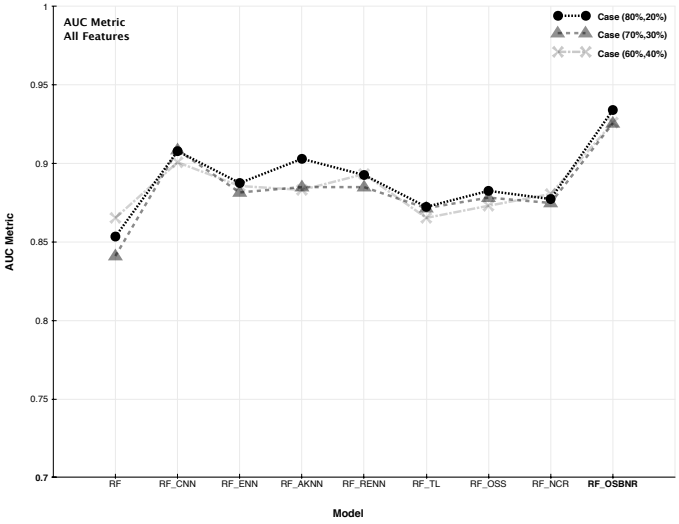

Fig. 3: AUC metric for OSBNR and the reference resampling approaches: RF as base classifier case of all features

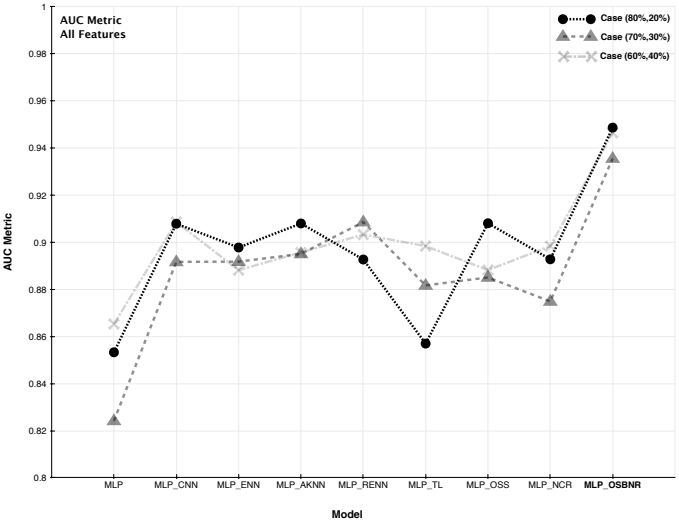

Fig. 4: AUC metric for OSBNR and the reference resampling approaches: MLP as base classifier case of all features

Figures 7 and 8 show the results obtained using Recall measure. The most interesting observation is that the RF and MLP classifiers without preprocessing clearly offer the best performance in terms of recall metric. Such a result was expected in a way, because the resampling methods were introduced to manage class imbalance and class overlap problems. They eliminate the instances of the majority class considered as noise to improve the prediction of instances of the minority class, and this can slightly increase the rate of false negatives. For the RF classifier, the second best recall score is obtained by RF_OSS with a value of ($Recall = 0.96$) when the training and test sets are set at 60% and 40% fraud.

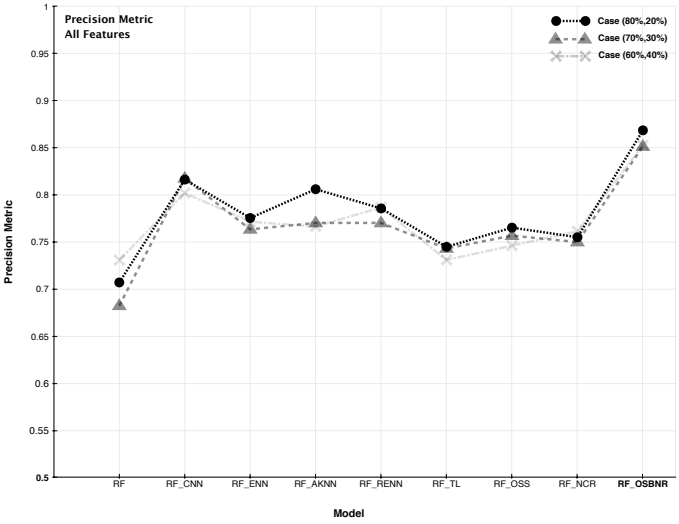

Fig. 5: Precision metric for OSBNR and the reference resampling approaches: RF as base classifier case of all features

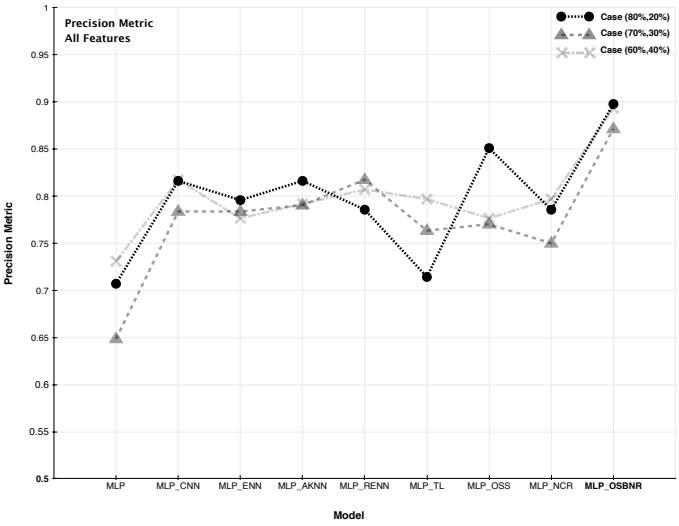

Fig. 6: Precision metric for OSBNR and the reference resampling approaches: MLP as base classifier case of features

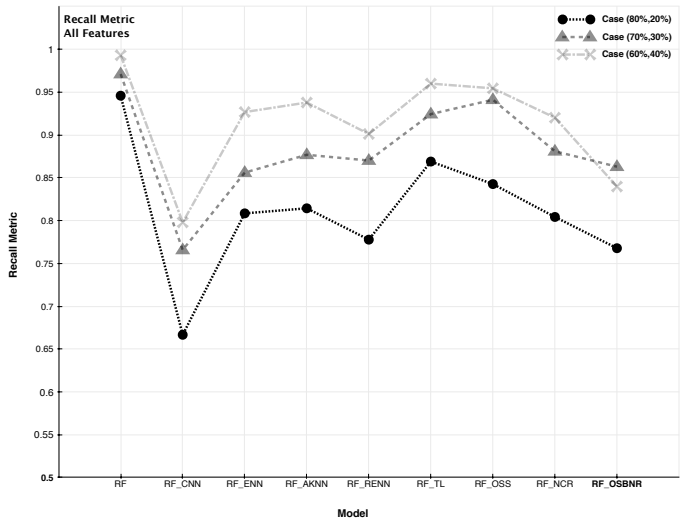

Fig. 7: Recall metric for OSBNR and the reference resampling approaches: RF as base classifier case of all features

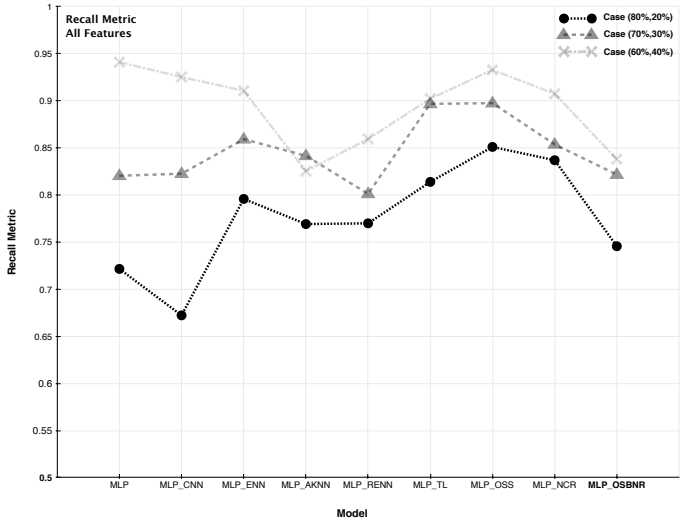

Fig. 8: Recall metric for OSBNR and the reference resampling approaches: MLP as base classifier case of all features

Similarly, for MLP the second place is occupied by MLP_OSS with a score of ($Recall = 0.9329$) with the $60/40$ rule.

As for measure F1-score, Figures 9 and 10 present the results of all the resampling methods by applying the two learning classifiers. We see that the OSBNR approach outperforms the other resampling methods for all the distributions of the training and test sets for the two classifiers. For RF, the best F1-score is obtained by RF_OSBNR with a score of ($F1 - score = 0.8572$) according to the $70/30$ rule, followed by RF_AKNN with a score of ($F1 - score = 0.8436$) when applying rule $60/40$ to divide the training and test sets. Regarding MLP, the best score is obtained by MLP_CNN with a F1-score of 0.8679 according to the $60/40$ rule, MLP_OSBNR takes second place with a value of 0.8648.

C. Performance study of the OSBNR: case of relevant features

In this section, we analyze the impact of the proposed OSBNR on the performance of each classifier by comparing it with existing resampling methods taking into account relevant features and also according to the 3 different divisions of the dataset. The results are calculated for four metrics: AUC, Precision, Recall and F1-score.

In order to better situate the results, we start by reporting on AUC metric from Figures 11 and 12. We can see that the best AUC scores are obtained by RF and MLP combined with OSBNR for all the distributions of the sets of training and testing. For the RF classifier, the best score is obtained by RF_OSBNR with an AUC value of ($AUC = 0.9442$), RF_CNN takes second place with a score of ($AUC = 0.9129$) according

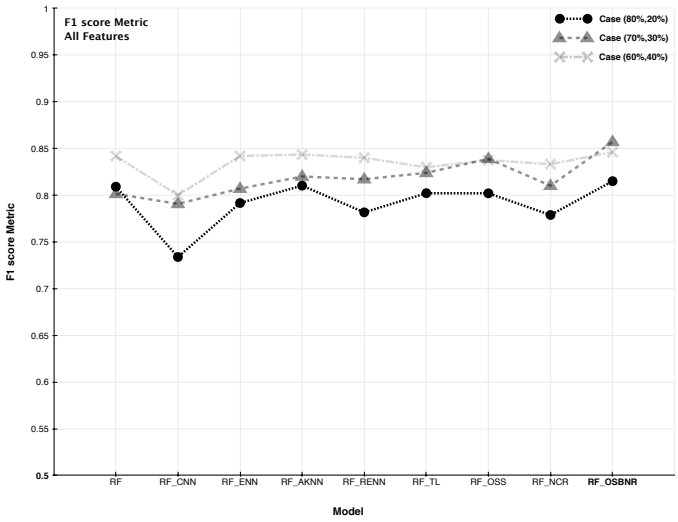

Fig. 9: F1-score metric for OSBNR and the reference resampling approaches: RF as base classifier case of all features

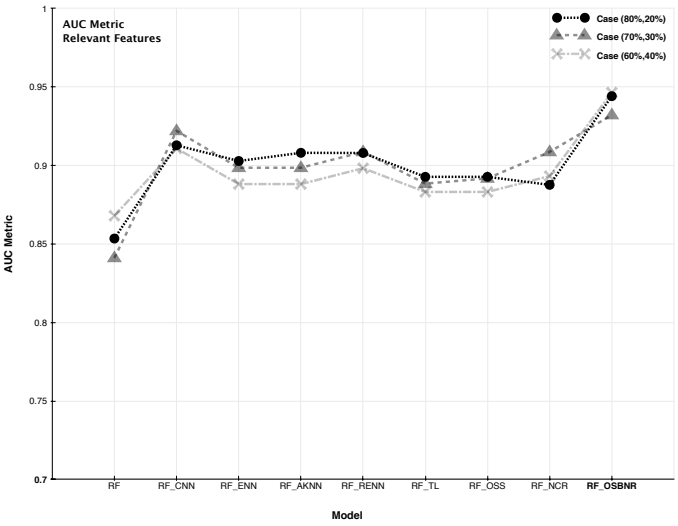

Fig. 11: AUC metric for OSBNR and the reference resampling approaches: RF as base classifier case of relevant features

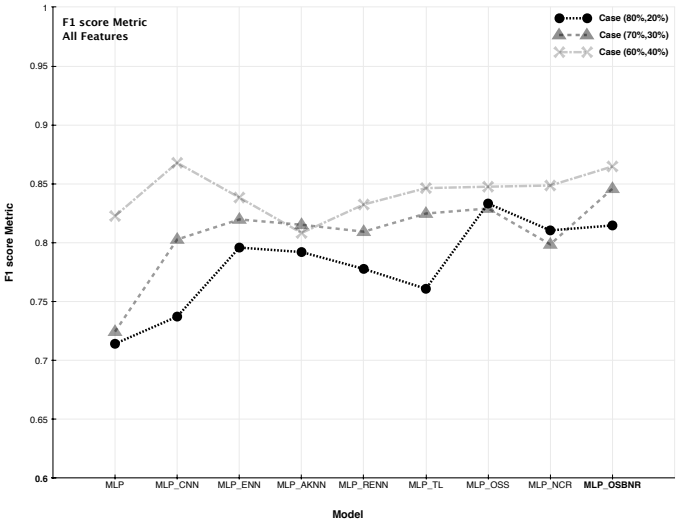

Fig. 10: F1-score metric for OSBNR and the reference resampling approaches: MLP as base classifier case of all features

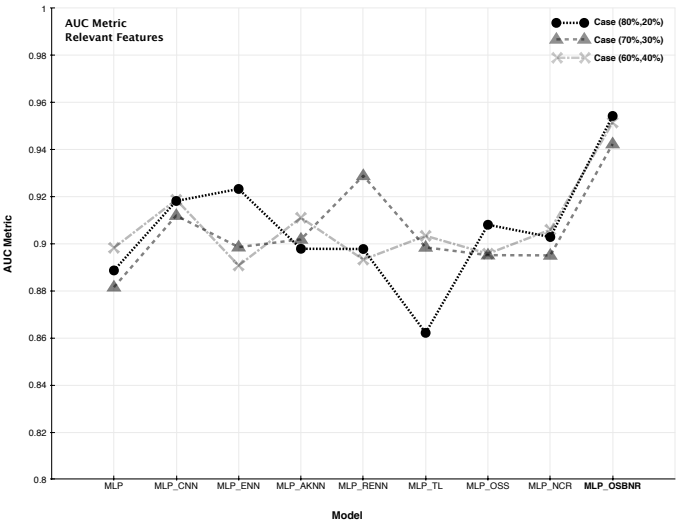

Fig. 12: AUC metric for OSBNR and the reference resampling approaches: MLP as base classifier case of relevant features

to the $80/20$ rule. Regarding the MLP classifier, the best AUC score is obtained by MLP combined with OSBNR when applying the $80/20$ rule with a value of ($AUC = 0.9543$), followed by MLP_RENN with a score of ($AUC = 0.9289$), while MLP_OSS maintained its score with the same value of ($AUC = 0.9079$). From these results, we can conclude that after eliminating redundant features that do not contribute to performance, we get continuity or even improvement in predictive performance.

Similarly, the figures 13 and 14 present the results in terms of precision. The results illustrated in Figure 13 show that the best precision score for the RF classifier is obtained by the OSBNR approach for all the distributions of training and test sets with a best value score of ($Precision = 0.8934$) according to $80/20$ rule, which means that this model offers a better prediction of minority instances. Similar in MLP, As illustrated in Figure 14, it is clear that the best precision score is obtained by MLP_OSBNR with a best score of ($Precision = 0.909$).

With regard to the Recall measure and according to the $70/30$ rule, Figures 15 and 16 show the results obtained. we also notice that the RF and MLP classifiers without preprocessing clearly offer the best performance in terms of recall metric. For the RF classifier, the second best recall score is obtained by RF_OSS with a value of ($Recall = 0.928$). Similarly, for MLP the second place is occupied by MLP_AKNN with a score of ($Recall = 0.9474$).

As for measure F1-score, Figures 15 and 16 show the results

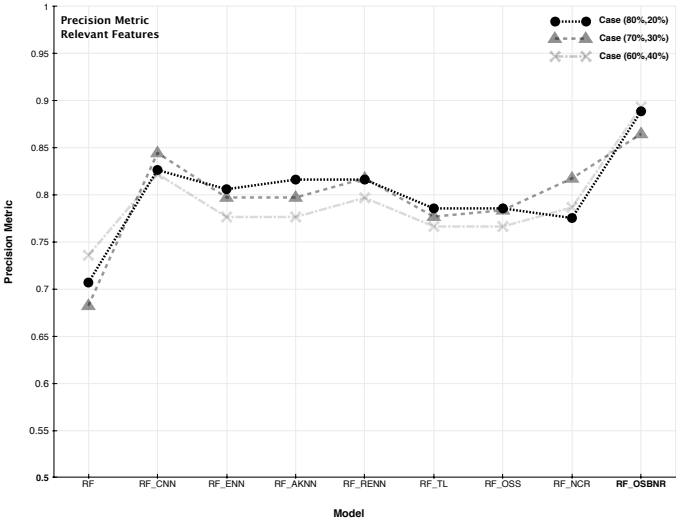

Fig. 13: Precision metric for OSBNR and the reference resampling approaches: RF as base classifier case of relevant features

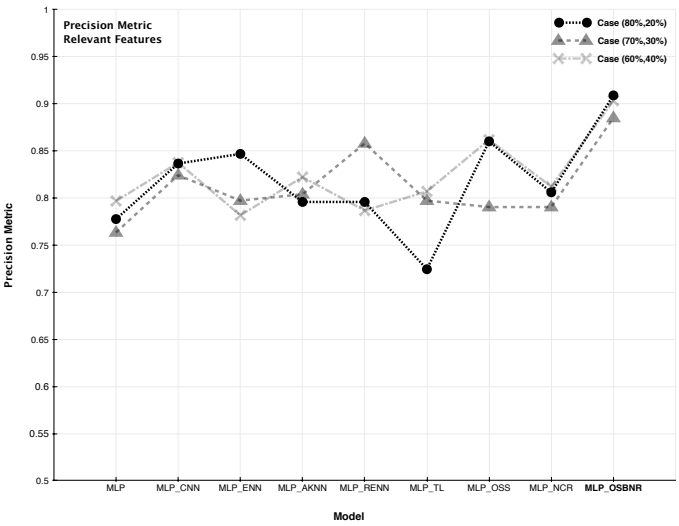

Fig. 14: Precision metric for OSBNR and the reference resampling approaches: MLP as base classifier case of relevant features

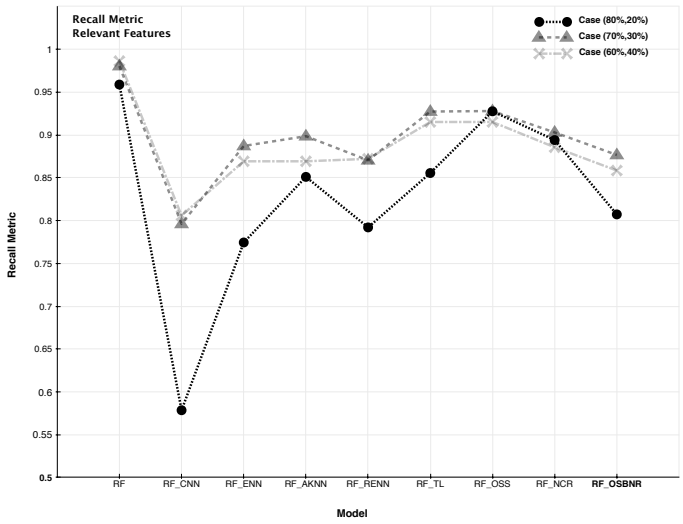

Fig. 15: Recall metric for OSBNR and the reference resampling approaches: RF as base classifier case of relevant features

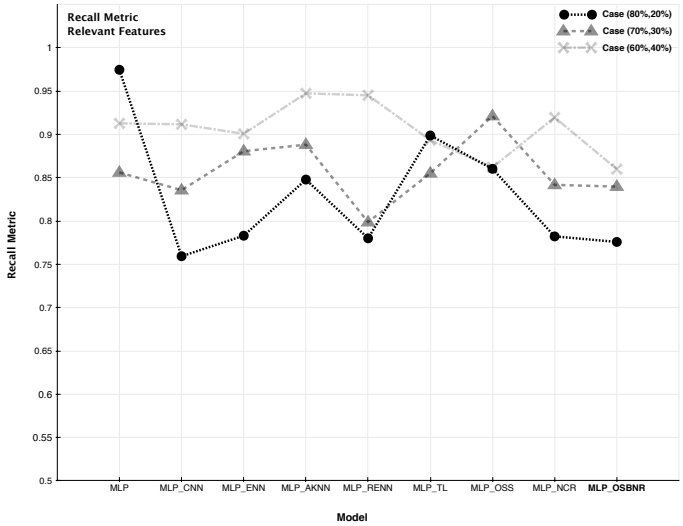

Fig. 16: Recall metric for OSBNR and the reference resampling approaches: MLP as base classifier case of relevant features

of all the resampling methods by applying the two learning classifiers. For RF, we see that the OSBNR outperforms the other resampling methods for all the distributions of the training and test sets. Likewise for MLP, the best score is obtained by MLP combined with OSBNR for all the distributions of training and test sets.

VI. Conclusion and Future Work

In this paper, we provide a survey of the existing methods for solving the two-class imbalanced classification problem. Then we present a new approach called: One Side Behavioral Noise Reduction (OSBNR). Our approach combines clustering analysis and a behavioral noisy data reduction process.

To study the effectiveness of the proposed method, we compare it to several state-of-the-art resampling methods. Two learning classifier, namely Random Forest and MultiLayer Perceptron, have been tested over these resampling methods. Experimental results measured using four metrics (AUC, Precision, Recall, F1-score) indicate that OSBNR achieves much better classification performance than the other compared methods to deal with noise data problem with a significant difference.

This work constitute an important part of the framework in development. Thus, we wish to study the behavior of the scaling of our approach in the context of a real application. This will raise two fundamental questions:

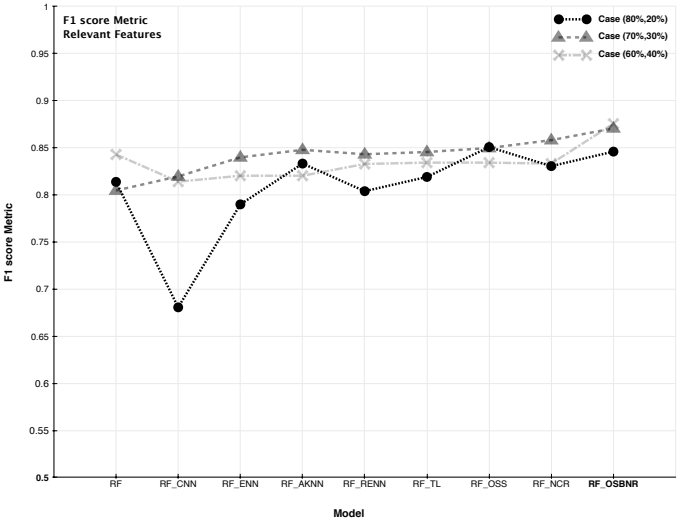

Fig. 17: F1-score metric for OSBNR and the reference re-sampling approaches: RF as base classifier case of relevant features

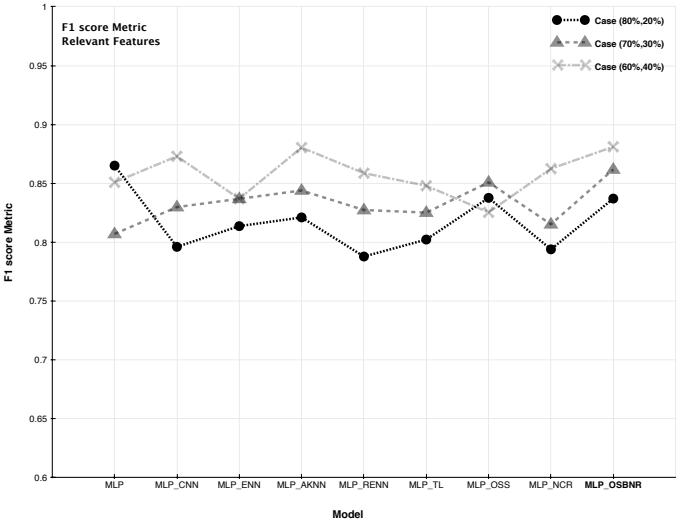

Fig. 18: F1-score metric for OSBNR and the reference re-sampling approaches: MLP as base classifier case of relevant features

- Confidence and predictability of predictions for decision making. The main objective of our explanatory approach to machine learning is to propose methods to understand and explain how the system produces its decisions in case of real domain application.
- Notion of uncertainty in machine learning which is of major importance and constitutes a key element of modern machine learning methodology. It has gained in importance due to the increasing relevance of machine learning in real applications.

ACKNOWLEDGMENT

This work is carried out thanks to the support of the European Union through the PLAIBDE project of the FEDER-FSE operational program for the Nouvelle-Aquitaine region, France. The project is supported by aYaline company, with partners: LIAS-ENSMA laboratory in Poitiers, and the L3i laboratory at La Rochelle University.

REFERENCES

[1] Hilario Alberto Fernández, López Salvador García, Galar Mikel, C. Prati Ronaldo, Krawczyk Bartosz, and Herrera Francisco. *Learning from imbalanced data sets*. Springer International Publishing, 2018.

[2] Aida Ali, Siti Mariyam Shamsuddin, and AncaL Ralescu. Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl*, 7(3):176–204, 2015.

[3] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. An efficient k-means clustering algorithm. 1997.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Peng Cao, Dazhe Zhao, and Osmar Zaiane. An optimized cost-sensitive svm for imbalanced data learning. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 280–292. Springer, 2013.

[6] Cristiano L Castro and Antônio P Braga. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems*, 24(6):888–899, 2013.

[7] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

[8] Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20, 2018.

[9] Mikel Galar, Alberto Fernández, Edurne Barrenechea, and Francisco Herrera. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern recognition*, 46(12):3460–3471, 2013.

[10] Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516, 1968.

[11] Kaggle Inc. Credit card fraud detection: Anonymized credit card transactions labeled as fraudulent or genuine, 2013.

[12] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[13] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[14] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Nashville, USA, 1997.

[15] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 63–66. Springer, 2001.

[16] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26, 2017.

[17] C.Victoria Priscilla and D.Padma Prabha. Credit card fraud detection: A systematic review. In *International Conference on Information, Communication and Computing Technology*, pages 290–303. Springer, 2019.

[18] Mostafizur Rahman and Darryl N. Davis. Cluster based under-sampling for unbalanced cardiovascular data. In *Proceedings of the World Congress on Engineering*, volume 3, pages 3–5, 2013.

[19] El Hajjami Salma, Malki Jamal, Berrada Mohammed, and Fourka Bouziane. Machine learning for anomaly detection. performance study considering anomaly distribution in an imbalanced dataset. In *2020 5th International Conference on Cloud Computing and Artificial Intelligence Technologies and Applications (Cloudtech'20)*. IEEE, 2020.

[20] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Andres Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571–595, 2014.

[21] Yu Sui, Mengze Yu, Haifeng Hong, and Xianxian Pan. Learning from imbalanced data: A comparative study. In *International Symposium on Security and Privacy in Social Networks and Big Data*, pages 264–274. Springer, 2019.

[22] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[23] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

[24] Mirza Amaad Ul Haq Tahir, Sohail Asghar, Awais Manzoor, and Muhammad Asim Noor. A classification model for class imbalance dataset using genetic programming. *IEEE Access*, 7:71013–71037, 2019.

[25] I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(2):679–772, 1976.

[26] Ivan Tomek. An experiment with the edited nearest-neighbor rule. 1976.

[27] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[28] Jianjun Zhang, Ting Wang, Wing WY Ng, Shuai Zhang, and Chris D Nugent. Undersampling near decision boundary for imbalance problems. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–8. IEEE, 2019.

[29] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210, 2004.