A Comparative Analysis of Unbalanced Data Handling Techniques for Machine Learning Algorithms to Electricity Theft Detection

Jeanne Pereira and Filipe Saraiva
Computer Science Postgraduate Program
Institute of Exact and Natural Sciences
Federal University of Pará
Belém – Pará – Brazil
Email: jeanneop22@gmail.com, saraiva@ufpa.br

***Abstract*—Non-technical losses of electric energy are mainly caused by electricity theft, causing damage to power utilities, reducing profits, increasing the energy costs to other consumers, and more. The methods of machine learning have been applied to detect electricity consumption anomalies. However, the characteristic of unbalanced classes in this kind of data opens a possibility to explore unbalanced data handling techniques, that are not explored in most of the literature studies. In this paper, the authors conduce a comparative study between several strategies to balance data sets and applied several machine learning techniques in order to select which machine learning + data handling techniques obtain the better results for the simulations related to the electricity theft detection problem. In this paper, the authors utilized the machine learning methods Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN), and the strategies for data balancing Cost-Sensitive Learning (Weighting), Random Undersampling (RUS), Random Oversampling (ROS), K-medoids based undersampling (K-medoids), Synthetic Minority Oversampling Technique (SMOTE) and Cluster-based Oversampling (CBOS). The metrics utilized for the comparison were Area Under ROC Curve (AUC) and F1-score, more suitable for this kind of problem. The results show some combinations can reach significantly better values than others, comparing both the balancing techniques for a same machine learning method itself as well as comparing these combinations between themselves.**

***Index Terms*—Classification problem; Electricity theft; Unbalanced data; Machine Learning.**

I. Introduction

Electricity losses occur in generation, transmission, and distribution components of the power systems. One possible definition for electricity losses is the difference between the injected energy and the measured energy delivered to consumers [1].

Electricity losses can be classified in technical and non-technical losses (NTLs). Technical losses are related to physical properties inherent to electrical systems components and their physical properties, as Joule effect [1], [2]. Electricity theft is the main cause of NTLs. This theft can be caused by bypassing or hacking the electricity meter, tampering the meter reading, and more [3].

As mentioned in a study from 2015 conducted by Northeast Group, NTLs costs US$ 89.3 billion per year to the World, where India losses US$ 16.2 billion, Brazil losses US$ 10.5 billion, and Russia US$ 5.1 billion [4]. NTLs can cause damage in power utilities, reduce profits of power companies, increase the energy costs to other consumers, reduce future investments in power utilities, compromise the public investment in other areas, and more [5]. In order to address this problem, studies to detect electricity theft have been performed using several techniques like statistics and machine learning models. Classification methods have been very used, for example to reduce the costs of on-site inspections [1].

However, data sets containing electricity theft class are generally unbalanced, because most of the data are composed by normal consumers, where the anomalous (thieves) are a very small part of the data set [6]. The problem with unbalanced data set is known in the literature: if the classes are very unbalanced, the machine learning method will learn how to classify the most common classes, while the less common will not be learned [7]. The method, in fact, will just guess the less common classes. Additionally, as these classes are small, they will not be detected in the accuracy metric because most of the classification will be performed in the most common classes [7]. Therefore, the machine learning method will be unusable for the aimed problem.

In order to illustrate this point, let's get an example of credit card fraudulent transactions [7]. Suppose in the whole data set only 1% is composed by fraudulent transactions data. If all transactions were classified as legitimate, the classifier achieved 99% of accuracy but none of fraudulent transactions were correctly recognized. Situations like the described here can occur to any classification problem where the data set is unbalanced.

There are some more sophisticated metrics to measure the machine learning performance suitable for this kind of data set, like Area Under ROC Curve (AUC) and F1-score [7]. Some studies, like [8]–[15], use these metrics for this kind of problems.

In addition, in order to handle the unbalanced class problem,

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

there are some techniques in the literature aimed to equalize or reduce the difference between the size of the classes in the data sets. Some of them are the Cost-Sensitive Learning [16], Random Undersampling (RUS) [8], Random Oversampling (ROS) [8], K-medoids based undersampling [17], Synthetic Minority Oversampling Technique (SMOTE) [18] and Cluster-based Oversampling (CBOS) [19].

This paper conduces a comparative study between several machine learning techniques and unbalanced data sets handling approaches applied to the electricity theft detection problem. The objective is to verify how these approaches can improve the results obtained by several machine learning techniques to the problem using more suitable metrics for this kind of problem, like AUC and F1-score. In addition, the paper suggests both better balancing approaches for a same machine learning technique, and the combinations of machine learning technique and unbalanced data sets approaches such the results are better than other combinations.

The paper is divided into the following sections. Section II describes briefly some papers about how to handle unbalanced data in several applications, including NTL detection. Section III discusses the techniques to handle unbalanced data used in this paper. The Section IV details the proposed study with classifiers and the techniques to handle unbalanced data. The results are presented and analysed in Section V. Lastly, Section VI presents the conclusions and future works.

II. Related work

Several techniques to handle unbalanced data like Cost-Sensitive Learning [16], RUS [8], ROS [8], CBOS [19], and SMOTE (Synthetic Minority Oversampling Technique) [18] are discussed in [20]. K-medoids based undersampling is other technique described in [17]. The sampling techniques are RUS, ROS, CBOS, SMOTE and k-medoids based undersampling.

[8] uses several machine learning methods for classification with RUS, ROS, SMOTE and CBOS in different kinds of applications as software engineering measurements, mammography, and other data sets from UCI Machine Learning Repository (glass, german-credit, solar-flare and more). There is not a sampling technique that surpasses all others in relation to all classifiers and the "intelligent" techniques do not present a satisfactory performance. To evaluate the performance were used AUC, Kolmogorov-Smirnov statistic [21], geometric mean [22], F1-score, and True Positive Rate (TPR).

In [17] the classifiers RF and SVM were implemented using no balance, RUS, ROS, SMOTE and k-medoids based undersampling. That paper describes an application with an unbalanced dataset of neuro images for Alzheimer's Disease. The results show k-medoids based undersampling achieving the best AUC results in most of the test cases.

The papers [9]–[15], [23] describes machine learning methods applied to detect NTLs or electricity theft. Considering unbalanced data handling techniques in these papers, besides using of class weights, sampling techniques were used in [10]–[14].

In [9] a lot of classifiers were compared to each other for NTL detection problem, but without any application of an unbalanced data handling technique. The classifiers compared were Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Stochastic Gradient Descent, Decision Tree, K-Nearest Neighbors (KNN), Adaboost, CatBoost, LightGBM, and XGBoost. Except by LR, all other classifiers achieved precision, TPR and F1-score greater than 0.9. However, other features besides historical consumption are considered, like 'month-billing' (month of billing), and 'BilledAmount' (amount billed in current month).

In [16] one type of cost-sensitive learning is weighting and the idea is the minority class has a weight greater than majority class. A cost-sensitive approach using class weights (weighting) for the problem of NTL detection is suggested in [9] but it is not utilized in that paper.

In [10] SVM and ANN with a combination of ROS and RUS were applied to NTL detection, where different levels of oversampling were tested. Linear SVM did not perform well in comparison to Non-Linear SVM and ANN. The metrics used to evaluate were TPR, precision, F1-score, True Negative Rate (TNR), Fβ-score, AUC and Matthews Correlation Coefficient (MCC).

In [11] to detect NTLs, AdaBoost was combined with RUS to become RUSBoost. Maximal Overlap Discrete Wavelet-Packet Transform (MODWPT) was used to extract features in the preprocessing stage. RUSBoost with MODWPT outperforms other machine learning frameworks (Linear SVM, Non-Linear SVM and ANN) in almost all metrics such as the same of [10] except by F1-score.

[12] uses the techniques LR, SVM, RF and KNN to detect NTLs. The dataset was undersampled to generate different NTL proportions. The models used time series for all the features (time series, neighborhood features and selected master data). The models were evaluated using AUC. In the first analysis, the training and testing used the same NTL proportions varying in an interval. The best proportions found in the first analysis were used in the second analysis to train, and these were tested in all NTL proportions. For the first analysis LR, SVM and KNN achieved the best results for a balanced dataset of 50%, RF achieved the best results for 60% using only time series and 40% using all features. For the second analysis, KNN only achieved the best result in one proportion, despite of the RF achieved the maximum AUC.

In [13] is considered the maximization of economic return in the detection of NTLs. There are two datasets, one containing synthetics frauds, the other a real fraud data with 6% of fraudulent costumers and with other features besides historical energy consumption. The second dataset was oversampled and the techniques SVM, RF and ANN were applied. RF presented the maximum economic return for the metrics F1-score, precision, and TPR.

In [14] in an application for the detection of NTLs, the majority class of honest customers are undersampled in two

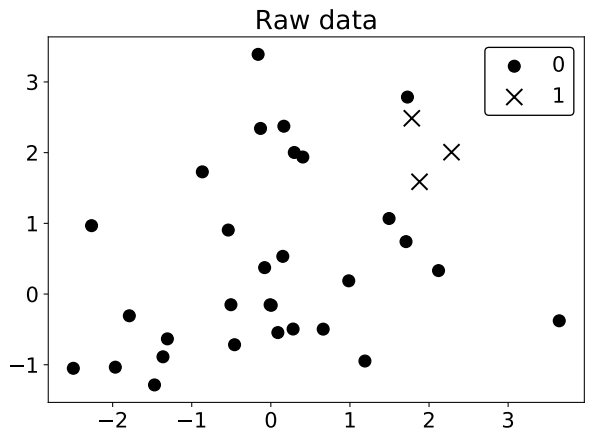


Fig. 1: Unbalanced data.

ways: first based in bad inspections and second randomly. The best machine learning technique applied without undersampling in comparison with SVM, LR and KNN was XGBoost. The metrics used for evaluation are AUC and precision-recall curve, recall is the same of TPR.

[23] uses the same dataset of [15] and they applied the classifiers CNN, Simple DNN (it will be referred as ANN because it has only four layers), SVM, RF and LR to detect electricity theft. In paper [23], the accuracy presented good values ($> 91\%$), however, other metrics like TPR, that represents detection rate of anomalies, and AUC were not presented. CNN achieved the highest accuracy in [15], [23], but they did not apply techniques to handle unbalanced data with the classifiers ANN, SVM, RF and LR.

According to the literature review performed, there is not a paper applied to NTL detection comparing machine learning classifiers combining the following unbalanced data handling techniques: Cost-Sensitive Learning, RUS, ROS, K-medoids, SMOTE, and CBOS. The contribution of this paper is to implement and evaluate some common machine learning classifiers without and with these balancing data set techniques and analysing them with metrics more suitable for the problems whose data set is unbalanced.

III. Techniques to handle unbalanced data

In this paper the following unbalanced data handling techniques were used to perform the comparison analysis: Cost-Sensitive Learning, RUS, ROS, K-medoids, SMOTE, and CBOS.

The respective techniques will be presented in following subsections. Figure 1 shows an unbalanced data used as a base for Figure 2, the elements of class 0 are represented with a filled circle and the elements of class 1 are represented with an 'x'.

A. Cost-Sensitive Learning (Weighting)

Class weights ('weighting') is a kind of cost-sensitive approach to handle unbalanced data sets. The weights are defined inversely proportional to the frequency of the classes [16]. This way, the class with less elements will have a high weight, balancing the classes of the data set. This approach has the same effect of sampling but the difference is that the number of samples do not increase or decrease [16]. In general the classifiers have support to the class weights, for example in an ANN the loss function is adapted to deal with class weights. Figure 2 - (a) shows the hyperplanes to separate classes for a SVM with class weights (dashed line) and without (solid line) in the unbalanced dataset presented in Figure 1. In Figure 2 - (a) SVM without class weights fails to classify elements of the class 1 different from SVM with class weights.

B. Random undersampling (RUS)

In RUS [8] majority training samples are randomly removed from the data set. This approach will put the same number of samples for all the classes. A disadvantage of this approach is the possible lose of important information because of the removed samples [17]. Figure 2 - (b) shows an application of RUS in the dataset presented in Figure 1.

C. Random oversampling (ROS)

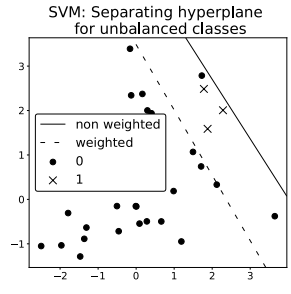
In ROS [8] minority training samples are randomly selected to be replicated. After the application of this technique, the quantity of the minority elements will be equal or almost equal to the majority elements. However, ROS can cause overfitting due to the repetition of several elements in the training dataset [17]. Figure 2 - (c) shows an application of ROS in the dataset presented in Figure 1, some points are duplicated, because of this they are overlapping each other.

D. K-medoids based undersampling

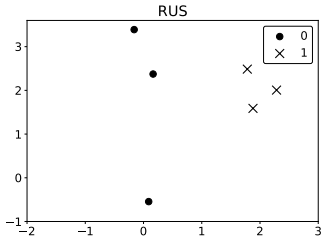
The majority class elements are clustered with k-medoids, where the number of clusters is equal to the number of minority training examples [17]. The training phase is performed using the medoids (centers of the clusters) and the minority class, because for this case they are balanced. Clustering makes it possible to extract general features related to elements of the majority class in a less amount of them. From this point of the paper, k-medoids based undersampling will be referred as just 'k-medoids'. Figure 2 - (d) shows an application of k-medoids in the dataset presented in Figure 1.

E. Synthetic Minority Oversampling Technique (SMOTE)

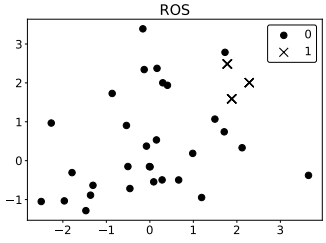
Proposed by [18], artificial minority samples are generated from the interpolation using the 'feature space' of existing minority samples and their k nearest neighbors. This procedure increases the number of minority samples to reach the number of majority samples, consequently the classes now have the same number of elements. Different from ROS, SMOTE reduces the number of duplicated instances by using interpolation. Figure 2 - (e) shows an application of SMOTE in the dataset presented in Figure 1. New instances were created interpolating each minority sample with its 2 nearest neighbors, one neighbor at a time. According to SMOTE, random values were used to create different synthetic samples for the same pair of a sample and its neighbor.



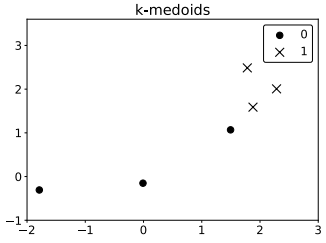
(a) An example of weighting.



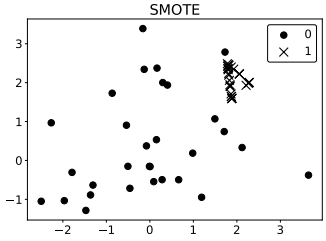
(b) An example of RUS.



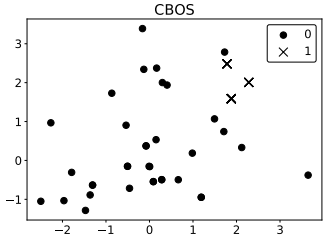
(c) An example of ROS.



(d) An example of k-medoids.



(e) An example of SMOTE.



(f) An example of CBOS.

Fig. 2: Unbalanced data handling techniques.

F. Cluster-based oversampling (CBOS)

Proposed by [19], each class is clustered separately by k-means to determine the best number of clusters. After that, random oversampling starts. In the majority class, all clusters except the largest one are randomly oversampled to contain the same numbers of elements of the largest one.

Let $maxclasssize$ be the total number of largest class elements. Each cluster in the minority class is randomly oversampled until the size of each cluster is equal to $maxclasssize/Nsmallclass$, where $Nsmallclass$ is the number of subclusters in the small class [19]. Finally majority class and minority class have the number of elements equal to $maxclasssize$.

CBOS takes into account *between-class imbalance* (imbalance between two different classes) and *within-class imbalance* (imbalance inside the subcluster of each class) [19]. Figure 2 - (f) shows an application of CBOS in the dataset presented in Figure 1, CBOS makes use of random oversampling, some points are duplicated and they overlapping each other.

TABLE I: Number of elements for unbalanced data handling techniques

Technique	Number of elements
Raw data (no balance)	class 0 = 30, class 1 = 3
Weighting	class 0 = 30, class 1 = 3
RUS	class 0 = 3, class 1 = 3
ROS	class 0 = 30, class 1 = 30
k-medoids	class 0 = 3, class 1 = 3
SMOTE	class 0 = 30, class 1 = 30
CBOS	class 0 = 36, class 1 = 36

Table I shows the number of elements in each class for raw data in Figure 1 and for the unbalanced data handling techniques showed in Figure 2. The unbalanced ratio of 'class 0' : 'class 1' is 10 : 1.

IV. Proposed work

This section describes the main characteristics and the steps of the study performed.

A. Data set description

The data set was available by State Grid Corporation of China (SGCC) [15] and contains electricity consumption data from 42372 consumers, where 38757 (91.47%) are normal consumers and 3615 (8.53%) are electricity thieves. The time interval for the data collection starts in 01/01/2014 and ending in 10/31/2016 (1035 days). One more day was added in order to reach 148 weeks ($1036/7 = 148$).

The samples were chosen randomly to compose training and validation sets, where 80% was used for training and 20% for validation. Each classifier with each unbalanced data handling technique was executed 10 times in order to obtain the results.

B. Preprocessing

Missing values, occurring when an instance does not present a value for an attribute [24], were filled according to the interpolation equation developed in [23] and presented in Equation (1). For that equation, x_i is the value of electricity consumption in a day, NaN means 'Not a Number' to represent missing values. The dataset contains approximately 25% of missing values referent to the values of electricity consumption, and all of them were added.

$$f(x_i) = \begin{cases} 0, & x_i \in NaN, i = 1; \\ x_{i-1}, & x_i \in NaN, i > 1; \\ x_i, & x_i \notin NaN; \end{cases} \quad (1)$$

C. Classifiers

The following classifiers were utilized in the paper in order to compare the unbalanced data handling techniques in the context of the problem aimed.

- Logistic Regression (LR): It is a basic model used for binary classification. It is similar to a neural network with only one layer and a sigmoid activation function [15], [25].
- Random Forest (RF): It is an ensemble method composed by decision trees. Random Forest can outperform a single

TABLE II: Algorithm arguments

Algorithm	Arguments and Values
LR	penalty: l2
RF	max depth: 7
Linear SVC	kernel: linear function
ANN	100 epochs

decision tree and avoids overfitting [24]. In a previous study [26] it was used to identify power quality disturbances.

- Support Vector Machine (SVM): It uses the concept that classes can be separated by a hyperplane, where support vectors are used to find these hyperplanes [24]. Linear SVC (Support Vector Classifier) is a SVM for classification with linear kernel, which is fast for large data sets [27]. This SVM architecture was utilized in this study.
- Artificial Neural Network (ANN): Also called multilayer feed-forward neural network or multilayer perceptron, has an architecture with one input layer, one or more hidden layers, and one output layer [24]. Despite of [23], [25] consider an ANN with two hidden layers a Deep Neural Network (DNN), this paper will consider this architecture an ANN, based on [28]. The ANN in this paper contains one input layer, two hidden layers and one output layer.

The classifiers LR, RF, SVM, and ANN were used with weighting, ROS, RUS, k-medoids, SMOTE and CBOS. The classifiers used the same parameters utilized in [23], as presented in Table II. ANN is implemented in according to [23] and the Table III shows the ANN architecture. The loss function, the same utilized in [23], was categorical cross-entropy with $softmax$ as activation function in the last layer - as the problem has two classes, this loss has the same effect of binary cross-entropy [29]. In ANN the other layers use rectified linear unit, the optimizer used was SGD and the batch size was 128.

In SMOTE, the parameter k (number of neighbors) used for testing were 2, 3, 5, 7, 10, 20, 30, 40, 50, 60, 70, and 100, and the selected value was associated with the best value of AUC.

CBOS technique was tested with the number of clusters for normal and anomalous classes from 2 to 11, where the best number of clusters for each class was determined by the elbow method and the silhouette coefficient [24].

In weighting the classes have weights inversely proportional to their frequencies. In RUS and ROS the normal and anomalous classes have the same number of elements. In k-medoids the number of clusters is equal to quantity of minority class elements.

The programming language used was Python that allowed use of libraries of machine learning algorithms, unbalanced data handling techniques, and also algorithms related to the database preprocessing. For the ANN implementation was

TABLE III: Model architecture

ANN
Input (1036)
layer1 (128)
layer2 (32)
layer3 (2)

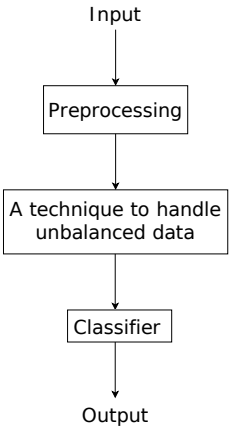


Fig. 3: Steps of execution.

used Keras[1] with Tensorflow[2] as backend; Scikit-learn[3] was used to implement LR, RF, SVM, k-means in CBOS, and other preprocessing and evaluation steps; for k-medoids implementation was used Pyclustering[4]. For RUS, ROS and SMOTE was used Imbalanced-learn[5]. Numpy[6] was used for preprocessing tasks and to deal with arrays and matrices. Pandas[7] was used in the preprocessing being useful to deal with csv files. Figure 3 shows the program steps. The program ran on a Windows 10 with Intel Core I7 2.9 GHz CPU and 8 GB of RAM.

D. Evaluation metrics

The results were evaluated using accuracy, AUC (Area Under ROC Curve), and F1-score.

Accuracy is the number of instances correctly classified divided by the total of instances. It is calculated as presented in equation (2), where, TP is True Positives, TN is True Negatives, FN is False Negatives, and FP is False Positives [24].

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \qquad (2)$$

For the problem of electricity theft detection, the normal consumers class in data set is much bigger than the theft class - normal consumers correspond to 91.47% of the data set utilized in this study. In this case, if the value of TN, related to normal consumers, is very high, the value for accuracy itself

[1]https://keras.io/
[2]https://www.tensorflow.org/
[3]https://scikit-learn.org/
[4]https://pypi.org/project/pyclustering/
[5]https://imbalanced-learn.readthedocs.io/en/stable/index.html
[6]https://numpy.org/
[7]https://pandas.pydata.org/

will also be high, independent of the value for TP. For this example, in terms of learning, the classifier does not care about how to correctly classify the uncommon class because it does not have any impact in accuracy metric. In the case, accuracy is not the metric more suitable for this kind of problem.

There are metrics more suitable for classification problems in unbalanced data sets, as AUC and F1-score.

AUC manages the trade-off between the rates of TP and FP. For these metric, how higher is the TP rate and lower is the FP rate, better is the performance of the classifier because it is correctly classifying the classes of the problem.

The AUC can be visualized as the ROC (Receiver Operating Characteristic) curve, that shows exactly the trade-off between TP and FP rates. The area under ROC curve is represented by AUC and its value varies between 0 and 1. If AUC is equal to 1, the classifier is perfectly classifying the classes. In other hands, if AUC is equal to 0.5, it indicates the classifier is performing a random guessing [7].

Precision in (4) is the ratio of correctly classified the positive class (TP) when compared to the total of positive classes (TP + FP). A high value for precision is related to low FP rate.

Recall in (3), also called sensitivity or True Positive Rate (TPR), is the rate of correctly positive class classified (TP) compared to all observations in the actual class (TP + FN). This metric analysis, in the universe of the positive classes, how many were correctly classified.

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{4}$$

F1-score, also called F-measure, is presented in (5). This metric is the weighted average of Precision and Recall [7] and its values varies from 0 (the worst) to 1 (the best) [30]. F1-score is usually more useful than accuracy, especially if the problem has unbalanced classes distribution. If the classes are very unbalanced, it is better to look at both Precision and Recall - and F1-score combine both in order to provide a more suitable metric for this kind of data set [7].

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

Next section shows the results using the metrics cited to evaluate machine learning classifiers LR, RF, SVM and ANN without and with unbalanced data handling techniques and it discusses the results obtained.

V. Results and discussion

Each combination of machine learning method and unbalanced data handling technique was executed 10 times in order to collect the results and perform the analysis. Tables IV to VII shows the mean results of AUC, F1-score and accuracy for each classifier.

The best results were highlighted in each table and metric by using letters in bold style. For methods without balancing techniques, it is expected high values in accuracy. However, for metrics more suited to data sets with unbalancing classes, as AUC and F1-score, these executions have poor results - it confirms the literature about the topic of unbalanced data sets [7]. For instance, despite RF without balancing has 91.63% in accuracy, this execution has only 0.52 in AUC - it implies the method is performing a guessing in the task of classification.

In other hands, the AUC value for RF + CBOS is 0.67, 15% better than the technique without data set balancing.

TABLE IV: LR results.

Method	AUC	F1-score	Accuracy
LR + No balance	0.5624	0.2036	**88.21%**
LR + Weighting	0.6395	0.2284	61.14%
LR + RUS	0.6315	0.2208	58.81%
LR + ROS	0.6415	0.2302	61.7%
LR + K-medoids	0.61	0.2026	49.68%
LR + SMOTE	**0.6501**	**0.2399**	64.54%
LR + CBOS	0.6421	0.2303	61.44%

TABLE V: RF results.

Method	AUC	F1-score	Accuracy
RF + No balance	0.5229	0.009	**91.63%**
RF + Weighting	0.6652	**0.3124**	82.01%
RF + RUS	0.6759	0.2822	73.95%
RF + ROS	0.6771	0.3108	79.91%
RF + K-medoids	0.6777	0.2832	73.88%
RF + SMOTE	0.6617	0.2758	75.3%
RF + CBOS	**0.6781**	0.3116	79.91%

TABLE VI: SVM results.

Method	AUC	F1-score	Accuracy
SVM + No balance	0.5614	0.1842	**85.52%**
SVM + Weighting	0.5568	0.1676	82.03%
SVM + RUS	0.5931	0.2036	65.46%
SVM + ROS	0.5969	0.2056	63.91%
SVM + K-medoids	0.5943	0.2002	63.07%
SVM + SMOTE	**0.6342**	**0.2281**	62.39%
SVM + CBOS	0.5953	0.2124	76.37%

Table IV shows the results for LR. For this classifier, SMOTE reach a better value in both AUC and F1-score.

Table V shows the results for RF. For this method, CBOS reach a better value in AUC while Weighting reach a better value in F1-score. However, F1-score has a close value for both techniques (0.312 for Weighting and 0.311 for CBOS) while AUC is a bit better in CBOS (0.678) than in Weighting (0.665). So, for RF, CBOS is the best method for balancing data set classes.

Table VI shows the results for SVM. Only SMOTE presents an AUC value greater than 0.6. F1-score of Weighting was the only with less than 0.2 value. For SVM technique, the results present SMOTE as a better balancing method in both AUC and F1-score.

Table VII shows the results for ANN. For this method, SMOTE reach best values in both AUC (0.679) and F1-score

TABLE VII: ANN results.

Method	AUC	F1-score	Accuracy
ANN + No balance	0.5691	0.2201	**91.46%**
ANN + Weighting	0.6089	0.2516	49.42%
ANN + RUS	0.6575	0.2808	68.19%
ANN + ROS	0.6426	0.3103	79.51%
ANN + K-medoids	0.6465	0.2732	79.39%
ANN + SMOTE	**0.6792**	**0.3169**	80.36%
ANN + CBOS	0.6088	0.2707	68.72%

TABLE VIII: Best results.

Metrics	Value	Technique
Best AUC	0.6792	ANN + SMOTE
Best F1-score	0.3169	ANN + SMOTE

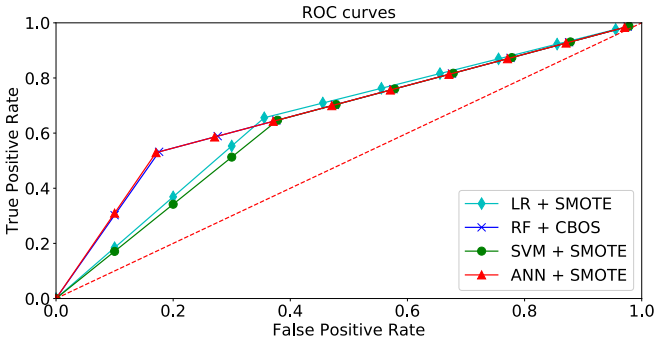


Fig. 4: ROC curves.

(0.316). Similar to LR and SVM, ANN + SMOTE achieved results significantly better than the others unbalanced data handling techniques.

Table VIII summarizes all results. In general the best AUC is associated with the best F1-score. SMOTE achieved the best AUC and F1-score in 3 of 4 classifiers (LR, SVM, and ANN). Comparing all the classifiers methods utilized, ANN was the best one in the main metrics utilized in this study, despite RF + CBOS has reached a very close value (AUC: ANN + SMOTE = 0.679, RF + CBOS = 0.678; F1-score: ANN + SMOTE = 0.316, RF + CBOS = 0.311).

Figure 4 shows the ROC curves for each classifier using the best unbalanced data handling technique as measured by AUC metric. As discussed previously and as presented in Tables V and VII, the ROC curves of ANN with SMOTE and RF with CBOS present a closer area. The dashed line represents a random guessing and good curves must to be above it.

The Tables IX and X show the average computational time for each method in all executions. In training and inference time for each execution, SMOTE considers the time to find the best number of neighbors. Only in training time, CBOS take into account the time to find the best number of normal clusters and anomalous clusters, while ANN the time for all epochs. RUS is faster than no balance because the number of instances is randomly removed and the number of instances is less than when the data is unbalanced.

TABLE IX: Training time.

Technique	LR training time	RF training time	SVM training time	ANN training time
No balance	9.49s	6.83s	59.81s	11min 3.63s
Weighting	10.74s	6.19s	56.36s	14min 22.84s
RUS	2.18s	1.29s	9.55s	3min 29.35s
ROS	24.97s	13.99s	1min 59.02s	20min 6.65s
K-medoids	13min and 45.33s	13min and 8.20s	13min and 18.76s	13min and 18.69s
SMOTE	6min and 25.33s	5min and 4.44s	36min and 59.3s	1h and 7.56min
CBOS	15min and 24.56s	14min and 11.70s	28min and 39.93s	1h and 52.43 min

TABLE X: Inference time.

Technique	LR inference time	RF inference time	SVM inference time	ANN inference time
No balance	1.80s	1.33s	1.83s	6.60s
Weighting	1.77s	1.61s	1.55s	5.04s
RUS	1.88s	1.37s	1.49s	5.67s
ROS	2.02s	1.51s	1.56s	4.26s
K-medoids	1.50s	1.58s	1.53s	4.35s
SMOTE	17.07s	18.73s	18.80s	1min 13.98s
CBOS	4.05s	3.69s	7.53s	14.86s

VI. Conclusion

In this paper, a comparative analysis between several unbalanced data handling techniques applied to several machine learning methods was performed for the context of electricity theft detection problem. The analysis conducted in the paper is very important because the data set for this problem has very unbalanced classes and, despite this characteristic, most of papers related to this problem does not use balancing techniques before the application of the classifier.

For the study, the balancing techniques compared were Weighting, RUS, ROS, K-medoids, SMOTE, and CBOS. The machine learning methods implemented were LR, RF, SVM, and ANN.

The results obtained show ANN combined with SMOTE reaches the best values for both AUC and F1-score, the metrics more suitable to problems with unbalanced classes than accuracy. However, RF combined to CBOS also reaches good values in those metrics when compared to ANN + SMOTE.

In short, the results obtained and discussed here can be used to point interesting combinations of data handling techniques and machine learning methods to be applied in the problem of electricity theft detection, but the results can be better and after an improvement this will become an interesting feature to be deployed by power distribution companies in the context of smart grids.

For future works, the authors would like to test different approaches for balancing classes in data set as the recently developed Generative Artificial Networks (GANs) and others, and also test others machine learning methods like Convolutional Neural Networks (CNNs). It is also in our planning

to test different values for parameters in machine learning methods in order to improve the results obtained for AUC and F1-score.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), a Brazilian research agency, for funding this study.

REFERENCES

[1] R. Bhat, R. Trevizan, R. Sengputa, X. Li, and A. Bretas, "Identifying Nontechnical Power Loss via Spatial and Temporal Deep Learning," in *15th IEEE International Conference on Machine Learning and Applications, ICMLA*, 2016, pp. 272–279.

[2] E. Sankari and R. Rajesh, "Detection of Non-Technical Loss in Power Utilities using Data Mining Techniques," *International Journal for Innovative Research in Science & Technology*, vol. 1, no. 9, pp. 97–101, 2015.

[3] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A Multi-Sensor Energy Theft Detection Framework for Advanced Metering Infrastructures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1319–1330, 2013.

[4] Northeast Group - LLC, "World Loses $89.3 Billion to Electricity Theft Annually, $58.7 Billion in Emerging Markets," https://www.prnewswire.com/news-releases/world-loses-893-billion-to-electricity-theft-annually-587-billion-in-emerging-markets-300006515.html, Access in 11/02/2019.

[5] Northeast Group - LLC, "Electricity Theft and Non-Technical Losses: Global Markets, Solutions, and Vendors," 2017.

[6] A. Maamar and K. Benahmed, "Machine Learning Techniques for Energy Theft Detection in AMI," in *Proceedings of the 2018 International Conference on Software Engineering and Information Management*, ser. ICSIM2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 57–62.

[7] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., 2005.

[8] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental Perspectives on Learning from Imbalanced Data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. ACM, 2007, pp. 935–942.

[9] K. M. Ghori, R. A. Abbasi, M. Awais, A. Ullah, and L. Szathmary, "Performance Analysis of Different Types of Machine Learning Classifiers for Non-Technical Loss Detection," *IEEE Access*, pp. 1–1, 2019.

[10] G. Figueroa, Y. Chen, N. Avila, and C. Chu, "Improved practices in machine learning algorithms for NTL detection with imbalanced data," in *2017 IEEE Power Energy Society General Meeting*, 2017, pp. 1–5.

[11] N. F. Avila, G. Figueroa, and C. Chu, "NTL Detection in Electric Distribution Systems Using the Maximal Overlap Discrete Wavelet-Packet Transform and Random Undersampling Boosting," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7171–7180, 2018.

[12] P. Glauner, J. A. Meira, L. Dolberg, R. State, F. Bettinger, and Y. Rangoni, "Neighborhood Features Help Detecting Non-Technical Losses in Big Data Sets," in *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, ser. BDCAT '16. Association for Computing Machinery, 2016, p. 253–261.

[13] P. Massaferro, J. M. D. Martino, and A. Fernández, "Fraud Detection in Electric Power Distribution: An Approach That Maximizes the Economic Return," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 703–710, 2020.

[14] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of Non-Technical Losses Using Smart Meter Data and Supervised Learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, 2019.

[15] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[16] C. X Ling and V. Sheng, "Cost-Sensitive Learning and the Class Imbalance Problem," *Encyclopedia of Machine Learning*, 2010.

[17] R. Dubey, J. Zhou, Y. Wang, P. Thompson, and J. Ye, "Analysis of sampling techniques for imbalanced data: An n = 648 ADNI study," *NeuroImage*, vol. 87, pp. 220—-241, 2014.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[19] D. T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *SIGKDD Explorations*, vol. 6, pp. 40–49, 2004.

[20] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, 2019.

[21] D. J. Hand, "Good practice in retail credit scorecard assessment," *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1109–1117, 2005.

[22] M. Kubat, R. C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," *Machine Learning*, vol. 30, no. 2, pp. 195–215, 1998.

[23] D. Yao, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, "Energy Theft Detection with Energy Privacy Preservation in the Smart Grid," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1., 2019.

[24] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers Inc., 2012.

[25] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[26] J. Diplock and D. Plecas, "The Increasing Problem of Electrical Consumption In Indoor Marihuana Grow Operations in British ColumbIa," 08 2019.

[27] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[28] G. Cybenko, *Continuous valued neural networks with two hidden layers are sufficient*. Department of Computer Science, Tufts University, 1988, Technical Report.

[29] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*, 1st ed. O'Reilly Media, Inc., 2017.

[30] S. Bhattacharyya, S. Bhattacharyya, H. Bhaumik, S. De, and G. Klepac, *Intelligent Analysis of Multimedia Information*, 1st ed. IGI Global, 2016.