Engineering Applications of Artificial Intelligence 48 (2016) 134–139

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Fuzzy-rough imbalanced learning for the diagnosis of High Voltage Circuit Breaker maintenance: The SMOTE-FRST-2T algorithm

E. Ramentol [a], I. Gondres [b], S. Lajes [b], R. Bello [c], Y. Caballero [a], C. Cornelis [d,*], F. Herrera [d,e]

[a] Department of Computer Science, University of Camaguey, Cuba
[b] Department of Electrical Engineering, University of Camaguey, Cuba
[c] Department of Computer Science, Universidad Central de Las Villas, Cuba
[d] Department of Computer Science and AI, University of Granada, Spain
[e] Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 September 2013
Received in revised form
11 October 2015
Accepted 20 October 2015

Keywords:
High Voltage Circuit Breaker (HVCB)
Imbalanced learning
Fuzzy rough set theory
Resampling methods

ABSTRACT

For any electric power system, it is crucial to guarantee a reliable performance of its High Voltage Circuit Breaker (HCVB). Determining when the HCVB needs maintenance is an important and non-trivial problem, since these devices are used over extensive periods of time. In this paper, we propose the use of data mining techniques in order to predict the need of maintenance. In the corresponding data, one class (minority, or positive class) is significantly less represented than the other (majority, or negative class). For this reason, we introduce a new imbalanced learning preprocessing algorithm, called SMOTE-FRST-2T. It combines the well-known Synthetic Minority Oversampling Technique (SMOTE) with a strategy of instance selection based on fuzzy rough set theory (FRST), using two different thresholds for cleaning synthetic minority instances introduced by SMOTE, as well as real majority instances. Our experimental analysis shows that we obtain better results than a range of state-of-the-art algorithms.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

A High Voltage Circuit Breaker (HVCB) is a device designed to open or close an electric circuit. It should be able to open circuits that operate on a wide range of capacities, varying from capacitive currents of a few hundred Amperes to inductive currents of many kA. This is the main reason why it is crucial for any electric power system to ensure a reliable performance of its breakers (ANSI, 2000).

Maintaining the HVCB is a very important task to improve its operational reliability. This maintenance is performed in time intervals dictated by the manufacturer. Nevertheless, very often these intervals are not workable in practice (AREVA, 2005). This difference between the predicted and real times occurs because there are some important variables that are not taken into account in the prediction (Lindquist et al., 2008; Rudd et al., 2011; Fan and Xiaoguang, 2012; Fnineche and Aitken, 2012; Runde et al., 2012).

Therefore, we need a reliable tool to predict when maintenance is necessary for an HVCB, avoiding undesired electric system faults. This problem can be considered as a classification task since given some input variables, the system should decide on two possible outputs: “maintenance needed” (positive class) or “maintenance not needed” (negative class).

To obtain the data for this problem, several measurements over different HVCBs have been taken and for each of them a team of experts evaluates whether maintenance is necessary or not. In the majority of cases, the answer is “maintenance not needed”. In other words, the class distribution in the resulting dataset is imbalanced; this characteristic is well-known and can be solved by imbalanced classification techniques (He and García, 2009; López et al., 2013; Sun et al., 2009).

A successful strategy to tackle imbalanced classification uses resampling methods that preprocess the data prior to classification (García et al., 2009). Many state-of-the-art resampling methods are based on the Synthetic Minority Oversampling Technique (SMOTE, (Chawla et al., 2002), an oversampling method that creates artificial minority class examples by interpolating between real minority examples and their nearest neighbors. SMOTE is often used in conjunction with a data cleaning method that eliminates examples (artificial minority ones, or majority ones) that are considered harmful for classification. Prominent methods include SMOTE-Tomek links and SMOTE-ENN (Batista et al., 2004), Borderline-SMOTE1 and Borderline-SMOTE2 (Han et al., 2005), Safe-Level-SMOTE (Bunkhumpornpat et al., 2009),

* Corresponding author.
E-mail addresses: enislay@gmail.com (E. Ramentol), igondrest@gmail.com (I. Gondres), santiago.lajes@reduc.edu.cu (S. Lajes), rbellop@uclv.edu.cu (R. Bello), yailec@yahoo.com (Y. Caballero), chriscornelis@ugr.es (C. Cornelis), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.engappai.2015.10.009
0952-1976/© 2015 Elsevier Ltd. All rights reserved.

SPIDER2 (Napierala et al., 2010), SMOTE-RSB* (Ramentol et al., 2012a) and SMOTE-FRST (Ramentol et al., 2012b).

In this paper, we propose a modification of the SMOTE-FRST algorithm in order to improve it for solving the HVCB maintenance problem. SMOTE-FRST uses fuzzy rough set theory (FRST, Dubois and Prade, 1990) in order to remove data points (instances) that do not sufficiently belong to the fuzzy rough positive region. An important drawback of the method is that to use a high threshold for instance removal usually ends up in eliminating too many original majority examples, causing a reduction in classification performance. On the other hand, choosing the threshold too low undermines the method's data cleaning purpose. In this paper, we deal with this dilemma by cleaning/reducing the training data using a double threshold for eliminating original majority data on the one hand, and synthetic minority data on the other hand. The resulting method is called SMOTE-FRST-2T.

We set up an experimental study to compare our proposal with SMOTE-FRST, as well as with the eight previously mentioned resampling algorithms. As we will see, SMOTE-FRST-2T outperforms all selected methods, demonstrating its competitiveness, and in particular is able to strike a balance between a low number of false negatives (that is, a failure to predict maintenance in the HVCB when it is actually necessary) and false positives (predicting an unnecessary maintenance step).

The remainder of the paper is structured as follows. In Section 2, we recall the HVCB maintenance problem and introduce the dataset used in our research. In Section 3, we present the details of our new proposal, while in Section 4 it is evaluated experimentally. Finally, in Section 5 we conclude.

2. Diagnosis of High Voltage Circuit Breaker maintenance

In this section, we describe in detail the problem of HCVB maintenance. HVCBs are mechanical switching devices that carry and disrupt electrical current in a circuit. Circuit breakers must function in normal and abnormal conditions, and must accommodate short circuits and outages. Circuit breakers are used with switching generators, power stations, cable feeders, transformers, and overhead lines in power distribution systems (Garzon, 2002).

2.1. The HVCB problem

The primary functions of a High Voltage Circuit Breaker include carrying rated current at rated voltage and power frequency when in closed position; interrupting rated currents at rated voltage and power frequency on command; and maintaining rated dielectric (power frequency and impulse) withstand levels when in open position. Sometimes the breaker may not open or close on command, allowing the fault to exist for a longer duration than the system can sustain while functioning normally (Garzon, 2002). Unless a breaker failure initiate action is taken, faults of breakers can lead to undesired changes in system functioning that may result in the system going into an abnormal state, potentially causing major system-wide power outage. This is the reason why it is so important to timely predict when maintenance is necessary for an HVCB.

To decide whether an HVCB needs maintenance is not obvious in many cases for they are used, open or close, for extensive periods of time. The need for a correct prediction of their performance grows in time with the expansion of the transmission systems since they transport more energy in wider regions. Hence preventive or time based maintenance is used. This means that maintenance to the HVCB is scheduled regularly on preset timeslots independent of its state. Nevertheless with the development of the technologies, new approaches for this scheduling have also been derived (ANSI, 2000). Predictive maintenance bases the decision on the inspection of the equipment on regular time-slots. It includes the objective (with the adequate tools) and subjective (with human senses) inspection as well as the reparation of the problem (potential fault). The goal is to accurately predict the condition of the breaker without opening it for inspection, augmenting its efficiency and significantly dropping its maintenance cost. This technique is normally performed by tests, statistical analysis or condition monitoring.

The effectiveness of the predictive maintenance depends on the accuracy of the analysis of the visual revision, tests and statistics to determine the level of the damage. In practice, there are several variables to be included in such an analysis which is often affected by the expertise of the specialist. In this paper, we use data mining techniques to perform such a prediction and the task is divided into the following steps:

1. Making-up the data-set, i.e.,
   (a) determining the variables;
   (b) measurements of each variable;
   (c) labeling each observation as maintenance needed or not by a human expert.
2. Determining the data-set characteristics given the amount of instances on each class.
3. Choosing the classifier to be used in the application.
4. Choosing the preprocessing techniques to be used in the application.
5. Evaluating the performance of the system in an experimental study.

2.2. Construction of the dataset

The variables used in our data were previously determined from international studies, rules and procedures according to approaches of several specialists of the electric company. The Delphi method was used to consult several experts to validate the feasibility and relevance of the variables used in the training set. This consult was applied to 35 renowned specialists, including 25 locals and 10 foreigners. For creating the dataset, the variables chosen were those that had more effect on the class. This procedure resulted in 17 variables that demonstrated to have a high impact on the decision associated to maintenance. These variables are shown in Table 1.

After deciding on the variables, we proceed with the measurements. In this process, experts provided important information related to the state of the breakers. All measurements were

Table 1
Descriptive variables.

Variable names	Type
Insulation chamber	Real
Insulation support	Real
Total insulation	Real
Contact resistance	Real
Gas pressure SF6	Real
Resistance coils	Real
Number of operations	Integer
Principal terminals	Nominal {1–10}
Porcelain	Nominal {1–10}
Temperature-compensated	Nominal {1–10}
Cabinets	Nominal {1–10}
Grounding	Nominal {1–10}
Connections	Nominal {1–10}
General control	Nominal {1–10}
Operational criticality	Nominal {high, normal, low}
Electrical wear	Nominal {high, normal, low}
Breaker wear	Nominal {high, normal, low}

labeled with their corresponding class (positive/ negative) assigned by the experts. The resulting dataset is composed of 369 examples; 120 belong to the positive or minority class (maintenance needed), while the 249 remaining examples represent the negative or majority class (maintenance not needed). In other words, there are more than two negative examples for each positive one.

3. Fuzzy-rough imbalanced learning with double threshold for the HCVB problem

In this section, we introduce our method SMOTE-FRST-2T designed to solve the HCVB problem. The method is a variation on SMOTE-FRST (Ramentol et al., 2012b), a preprocessing method that evaluates each synthetic and majority instance using a measure based on fuzzy rough set theory (Dubois and Prade, 1990) (FRST), and deletes those instances for which the value does not exceed a given threshold. As we will see, this proposal results unsuitably in the context of diagnosis of HVCB maintenance, because it eliminates many important original examples. Our proposal is described in detail in Section 3.2. In the next subsection, we first recall the necessary concepts from fuzzy rough set theory.

3.1. Fuzzy Rough Set Theory (FRST)

Rough Set Theory (RST) was introduced by Pawlak (1982), and has evolved into a popular methodology for dealing with uncertainty produced by inconsistencies in data (Bello, 2008). The theory revolves around the notion of (in)discernibility: the ability to distinguish between instances, based on their attribute values. Rough sets are often hybridized with fuzzy sets (Zadeh, 1965) which model gradual transitions in the satisfaction of a concept or relation. In the resulting FRST (Dubois and Prade, 1990), indiscernibility is typically modeled by means of a fuzzy relation R that expresses how similar two instances are on a scale from 0 (totally dissimilar) to 1 (completely indiscernible).

In this paper, we assume the following definitions, as proposed also in Ramentol et al. (2012b). Let x be the set of data instances and $\mathcal{A}$ the set of attributes. Given a real attribute a in $\mathcal{A}$, and two instances x and y in X, such that $a(x)$ and $a(y)$ represent the values of x and y for a, respectively, we define

$$R_a(a,b) = \max\left(1 - \frac{|a(y)-a(x)|}{range(a)}, 0\right) \quad (1)$$

where $R_a(x,y)$ is a value between 0 and 1 which evaluates the indiscernibility between x and y: the higher $R_a(x,y)$ the closer (more similar) x and y are. On the other hand, if a is a nominal attribute, we define

$$R_a(x,y) = \begin{cases} 1 & \text{if } a(x) = a(y) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In this case, the indiscernibility with respect to a is two-valued; if two instances have different values for a, they can be discerned $R_a(x,y) = 0$, while in the opposite case they cannot $R_a(x,y) = 1$. In order to compare x and y w.r.t. the entire set of attributes $\mathcal{A}$, we calculate the following aggregated value:

$$R(x,y) = T_L(\underbrace{R_a(x,y)}_{a \in \mathcal{A}}) \quad (3)$$

where T_L represents the so-called *Lukasiewicz t-norm* given by $T_L(v_1,\ldots,v_n) = \max(0,\ v_1+\ldots+v_n-n+1)$ for $v_1,\ldots,v_n \in [0,1]$, where $n = |A|$. R is called the fuzzy-rough indiscernibility relation.

Using R, we can evaluate, for each data instance x, its membership to the positive region *POS*, by looking at the most similar instance y which has a different class label:

$$POS(x) = \min_{class(y) \neq class(x)} 1 - R(x,y)) \quad (4)$$

The idea of the positive region is that instances x on the border of a class (i.e., for which there exists a similar instance in another class) will have a small $POS(x)$ value compared to instances in the center of a class. This makes the positive region suitable to measure the quality of an instance as a typical representative of its class.

3.2. Using two thresholds: SMOTE-FRST-2T algorithm

In Ramentol et al. (2012b), we introduced a hybrid method for preprocessing imbalanced data called SMOTE-FRST. This method consists of two stages: first, it generates synthetic minority examples and then it evaluates the dataset using a cleaning method based on FRST. These stages are repeated until the training data is balanced, or until a maximum number of iterations T is reached.

The hypothesis of SMOTE-FRST is that some of the introduced synthetic minority class instances may not be suitable for use in the learning phase, and hence should be eliminated. Similarly, majority class instances in the original training data that do not sufficiently belong to the positive region are removed. Note that we do not apply this procedure to the original minority class instances, since they are relatively sparse and are better left untouched. The algorithm needs two parameters: the number of iterations and the positive region threshold. In Ramentol et al. (2012b), $T=10$ and $\gamma = 0.8$ were proposed.

In this paper, we modify this algorithm as the use of a single threshold proves inadequate. For instance, when using $\gamma = 0.8$ in the HCVB problem, all majority examples are eliminated, which is clearly undesirable. On the other hand, if we lower the threshold to retain more majority examples, the quality of the synthetic examples also gets poorer, which has a negative effect because synthetic minority instances should represent a real occurrence of an uncommon event.

For these reasons, we propose a new method called SMOTE-FRST-2T that uses two different thresholds to evaluate the instances using FRST. Algorithm 1 shows the detailed steps of our method. In particular, SMOTE-FRST-2T consists of three stages:

1. Apply SMOTE (Chawla et al., 2002) to introduce new synthetic minority class instances to the training set (line 6).
2. Insert synthetic instances (x_s) for which the membership degree $POS(x_s)$ to the positive region of the training set is higher than a given threshold γ_S (lines 7–13).
3. Insert majority class instances (x_m) for which the membership degree $POS(x_m)$ to the positive region of the training set is higher than a given threshold γ_M (lines 14–20).

Steps 2 and 3 are repeated until a predetermined number T of executions are reached, or the resulting dataset is balanced, i.e., contains an equal number of examples of each class (line 5).

The membership degree POS is computed in each step of the algorithm with respect to all instances in the training set (original + synthetic). In every step of the algorithm the original majority instances are the resulting instance of the pervious step.

Algorithm 1. SMOTE-FRST-2T algorithm.

Require: threshold for synthetic examples γ_S
threshold for majority class γ_M
maximum number of iterations T
array of majority examples *maj*[ ]
array of minority examples *min*[ ]

Ensure: *resultSet*
1: *resultSet* = *min*[ ]
2: *executionNumber* = 0
3: *nsynt* = *nmaj* = 0
4: *isbalance* = *false*
5: **while** (*executionNumber* < = *T*)& !(*isbalance*) **do**
6: Apply SMOTE to create an array *syntInst*[ ] of synthetic minority examples
7: **for** *i* ← 1 **to** (*syntInst*[ ].*length*) **do**
8: Compute *POS*(*syntInst*[*i*])
9: **if** *POS*(*syntInst*[*i*]) > = γ_S **then**
10: *resultSet* = *resultSet* ∪ *syntInst*[*i*]
11: *nsynt* + +
12: **end if**
13: **end for**
14: **for** *j* ← 1 **to** *maj*[ ].*lenght* **do**
15: Compute *POS*(*maj*[*j*])
16: **if** *POS*(*maj*[*j*]) > = γ_M **then**
17: *resultSet* = *resultSet* ∪ *maj*[*j*]
18: *nmaj* + +
19: **end if**
20: **end for**
21: *balance* = (*nmaj* = = (*nsynt* + *nmin*[ ].*lenght*))
22: *executionNumber* + +
23: **end while**

The underlying idea of SMOTE-FRST-2T is, on the one hand, to use a very low threshold γ_M for the original majority examples in order to eliminate only a few of them, namely those with a very weak membership to the positive region. Indeed, in a real-world application like the HVCB dataset, every original instance represents one real occurrence, so we need to be very careful with every instance that we decide to eliminate. On the other hand, we propose to use a very high value for the synthetic examples; this helps us to insert only synthetic examples with a very high membership to the positive region to the training set. In the next section, we will validate this hypothesis experimentally on the HCVB dataset.

4. Experimental study

In this section, we experimentally evaluate the proposed SMOTE-FRST-2T algorithm on the HCVB dataset described in Section 2, comparing our proposal to the original SMOTE-FRST method, as well as to the state-of-the-art resampling techniques listed in Section 1.

In Section 4.1, we describe the setup of our experiments and the selection of the parameters associated with our method, while in Section 4.2, we give the results and analyze them.

4.1. Setup and parameter selection

The general outline of our experiments[1] is as follows: the HCVB dataset is divided into five parts in order to perform a 5-fold cross-validation (5FCV) procedure. Each fold is first preprocessed using one of the resampling methods listed below, and then it is transferred to the base classifier (learning algorithm), which in our case is C4.5 (Quinlan, 1993). Classification performance is then evaluated using the Area Under the Curve (AUC) metric (Huang and Ling, 2005). Moreover, the 5FCV procedure is repeated 5 times using different partitions of the data, in order to increase the reliability of the obtained results and the conclusions derived from them.

Our choice of C4.5 as learning algorithm is motivated by the fact that it has been identified as one of the 10 top algorithms in data mining (Wu et al., 2008), and has been widely used in imbalanced problems (Batista et al., 2004). We used the parameters recommended by its author: we set a confidence level of 0.25, the minimum number of itemsets per leaf was set to 2 and the application of pruning was used to obtain the final decision tree.

In order to evaluate SMOTE-FRST-2T, we compare it to nine existing resampling algorithms: SMOTE (Chawla et al., 2002), SMOTE-TL (Batista et al., 2004), SMOTE-ENN (Batista et al., 2004), Borderline-SMOTE1 (Han et al., 2005), Borderline- SMOTE2 (Han et al., 2005), Safe-Level-SMOTE (Bunkhumpornpat et al., 2009), SPIDER2 (Napierala et al., 2010), SMOTERSB* (Ramentol et al., 2012a) and SMOTE-FRST (Ramentol et al., 2012b). The parameter values used for these methods are those recommended by their authors. Only the algorithm SMOTE-FRST was modified slightly in order to be able to apply it meaningfully here. As indicated in Section 3, the original version of SMOTE-FRST eliminates all majority examples for the HCVB dataset. For this reason, we decided to leave the majority instances unchanged when using SMOTE-FRST, and to eliminate only synthetic minority instances according to the threshold γ_s. We denote this modified version as SMOTE-FRST- S, pointing out its action on the synthetic instances (examples). Both $\gamma_s = 0.8$ and $\gamma_s = 1$ were tried.

As mentioned in Section 3.2, SMOTE-FRST-2T needs three parameters: the number of iterations *T*, the threshold γ_M to evaluate original majority instances using the positive region *POS* and the threshold γ_s to evaluate synthetic instances, also using *POS*.

In Ramentol et al. (2012b), we proposed the use of 10 iterations, while in this research we used 5; γ_s for the synthetic examples is fixed as 1 and γ_M for the original majority examples is fixed as 0.03. These parameters were selected using the following assumptions:

- We limit ourselves to 5 iterations, because given the number of examples to generate and the quality of the synthetic examples that are demanded with our evaluation method, 5 iterations should be sufficient to reach convergence.
- An instance that belongs to degree 1 to the positive region is absolutely certain to be a good representative of its class. Since we work with a real world application in which every synthetic example represents a need for maintenance state of the HVCB, such synthetic examples need to be carefully introduced, since they have not been observed in a real observation. This is the reason why only very good representative synthetic examples are introduced, meaning that the threshold for the membership degree to the class of such examples has to be the maximum possible (1).
- Each original example in the dataset represents a real observation of the engineers on the HVCB, hence removing any of these examples has to be considered with great care as it may lead to losing important knowledge. A membership degree very close to zero represents an example that is very close to the opposite class: these examples greatly affect the separability of the classes in a classification process, or can even be considered as noise. In this work, it was decided to eliminate all those majority instances with a positive region membership degree below 0.03.

4.2. Results

In Table 2, we report the average AUC values and standard deviations obtained over 5 independent runs of the 5FCV

[1] Our experiments were executed using the KEEL software tool (Alcalá et al., 2010); all the methods referenced below are implemented in it.

Table 2
Comparison of the AUC results for training and test data, averaged over 5 independent runs of 5-fold cross validation, along with their standard deviation.

Nr	Resampling method	tra	tst
1	Original	0.9715 ± 0.0073	0.9340 ± 0.0123
2	SMOTE	0.9779 ± 0.0051	0.9341 ± 0.0100
3	SMOTE-TL	0.9522 ± 0.0063	0.9140 ± 0.0065
4	SMOTE-ENN	0.9586 ± 0.0042	0.9202 ± 0.0055
5	Borderline-SMOTE1	0.9692 ± 0.0055	0.9214 ± 0.0178
6	Borderline-SMOTE2	0.9657 ± 0.0083	0.9234 ± 0.0138
7	Safe-Level-SMOTE	0.9581 ± 0.0054	0.9309 ± 0.0114
8	SPIDER2	0.9531 ± 0.0064	0.9264 ± 0.0087
9	SMOTE-RSB*	0.9739 ± 0.0059	0.9427 ± 0.0058
10	SMOTE-FRST-S ($\gamma_S = 0.8$)	0.9773 ± 0.0039	0.9400 ± 0.0146
11	SMOTE-FRST-S ($\gamma_S = 1$)	0.9773 ± 0.0044	0.9394 ± 0.0098
12	SMOTE-FRST-2T	0.9725 ± 0.0044	0.9520 ± 0.0109

Table 3
Mean of created and deleted instances by the methods SMOTE-FRST-S and SMOTE-FRST-2T over 5 independent runs of 5-fold cross validation.

Nr	**SMOTE-FRST-S**	**SMOTE-FRST-2T**
Fold 1		
#may deleted	0	5.4
#synt created	103	97.6
Fold 2		
#may deleted	0	6.6
#synt created	103	96.8
Fold 3		
#may deleted	0	7
#synt created	103	96.6
Fold 4		
#may deleted	0	8.8
#synt created	103	94.4
Fold 5		
#may deleted	0	7.4
#synt created	104	96.6

procedure with each strategy. For reference, we also include the performance of C4.5 without introducing any preprocessing to the data (original data). We also distinguish between training (tra) and test (tst) AUC: the former shows how well the classifier works on the data that was used for training it, while the latter reveals how good it is at making predictions for unseen data.

It can be seen that among the compared algorithms, SMOTE-FRST-2T obtains the best test AUC, outperforming the second best method (SMOTE-RSB*) by almost one percent.

Note also that several of the other considered preprocessing methods do not manage to improve the AUC obtained using the unreduced data, showing that they are unsuitable for this problem.

It can also be seen that the two considered variants of SMOTE-FRST-S obtain a higher training AUC than SMOTE-FRST-2T, but do not perform as well as on the test data. This fact supports the benefits of removing the original majority examples with a very low membership to the majority class. For a more detailed comparison between our proposal and its predecessor, Table 3 shows the mean of data remaining in each fold after executing SMOTE-FRST-S ($\gamma_s = 0.8$) and SMOTE-FRST-2T over 5 independent runs of 5-fold cross validation. Each row labeled as "#maj deleted" shows the number of original majority examples deleted. Each row labeled as "#synt created" shows the number of synthetic examples generated. As mentioned, SMOTE-FRST-S does not eliminate any majority sample; however, by using a low threshold when evaluating the majority samples, SMOTE-FRST-2T eliminates a few in each fold, as the low positive region membership indicates that they are probably noisy. On the other hand, using a higher value to evaluate the synthetic samples leads to SMOTE-FRST-2T deleting more synthetic instances than SMOTE-FRST-S. From the results, it can be inferred that these modifications lead to an overall better classification.

Table 4
Mean of the false negatives and false positives obtained by each method over 5 independent runs of 5-fold cross validation.

Methods	False negative	False positive
Original	11.4	7.8
SMOTE	8.2	15.8
SMOTE-TL	5.4	31.6
SMOTE-ENN	12.4	14
Borderline-SMOTE1	9.4	19.6
Borderline-SMOTE2	7	23.6
Safe-Level-SMOTE	8	17.8
SPIDER2	6.4	23.4
SMOTE-RSB*	7	14
SMOTE-FRST-S ($\gamma_S = 0.8$)	11.4	5.8
SMOTE-FRST-S ($\gamma_S = 1$)	10.4	9
SMOTE-FRST-2T	7	9.4

Table 5
Maximum false negatives obtained by each method over 5 independent runs of 5-fold cross validation.

Methods	Maximum false negatives
Original	15
SMOTE	10
SMOTE-TL	8
SMOTE-ENN	14
Borderline-SMOTE1	17
Borderline-SMOTE2	9
Safelevel-SMOTE	13
SPIDER2	9
SMOTE-RSB*	9
SMOTE-FRST-S ($\gamma_s = 0.8$)	15
SMOTE-FRST-S ($\gamma_s = 1$)	13
SMOTE-FRST-2T	8

Finally, in Tables 4 and 5 we investigate the number of false positives and false negatives for the compared algorithms; the numbers in Table 4 are the mean over 5 independent runs of 5-fold cross validation. The table shows that SMOTE-FRST-2T yields on average 7 false negatives, which is the same result as that obtained using Borderline-SMOTE2 and SMOTE-RSB*. This mean is improved to 6.4 by SPIDER" and to 5.4 when using SMOTE-TL. However, it can be seen that all these competing methods reach a high false positives rate compared to our proposal. On the other hand, in Table 5 we can observe that our proposal and SMOTE-TL obtained the same maximum number value of false negatives over all 5 runs of cross-validation, while for Borderline-SMOTE2, SPIDER2 and SMOTE-RSB* is strictly higher.

The above observations have an important consequence from an electrical engineering point of view: indeed, the main objective is to reduce the false negative rate while also reducing the false positive rate, this means finding an equilibrium between failures in both classes.

If the HVCB does not need maintenance and the system predicts "yes", the associated cost relates to testing the equipment, perhaps the maintenance itself to the involved parts, or the associated cost, to the out-of-work time of the HVCB while it is opened to check if it really needs maintenance. This could generate faults on the electrical system. If the system predicts "no" when the maintenance is actually needed, the problem is far worse since this can cause damage to the system and eventual failure. These damages might produce the loss of synchronism of the generators and lead the system (or part of it) to shut down. The associated cost in terms of replacements may rise, as well as the economic losses given the electric faults.

Concluding, while an improvement of only 1% in AUC may appear comparatively small, the reduction in associated cost due to failure in both classes is significant, and may save the enterprise operating the HCVB a considerable expense.

5. Conclusion

In this paper, we have proposed the algorithm SMOTE-FRST-2T as a preprocessing step in order to predict the necessity of maintenance of a HVCB. Our main contribution from the machine learning point of view can be summarized as follows:

- The use of two thresholds allows treating original majority instances in a different way than synthetic instances.
- Majority instances with a very low membership to the positive region are deleted and only synthetic examples with the highest value of membership to the positive region are inserted into the final dataset.
- Our proposal obtained better results than nine well-known algorithms of the state-of-the-art.

Summing up, our method is able to reduce in a significant way the number of misclassified examples. From an electrical engineering point of view, this translates to avoiding power system faults, HVCB ruptures and replacements, and unnecessary opening of the equipment, as well as to saving significant resources.

Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Technology under the Project TIN2014-57251-P and the Andalusian Research Plans P10-TIC-6858, P11-TIC-7765 and P12-TIC-2958, and by Project PYR-2014-8 of the Genil Program of CEI BioTic GRANADA.

References

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2010. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. (17), 255–287.

AREVA, 2005. Manual de instrucciones. Interruptores de SF6 GL314 con mandos de resortes FK3–1. T&D High VoltageProducts, France.

Batista, G.E.A.P.A., Prati, R.C., Monard, M.C., 2004. A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explor. 6 (1), 20–29.

Bello, R., Falcon, R., Pedrycz, W., Kacprzyk, J., 2008. Granular Computing: At The Junction of Rough Sets and Fuzzy Sets. Springer, Berlin-Heidelberg.

Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, vol. 3644, pp. 475–482.

ANSI C37.06, 2000. AC high-voltage circuit breakers rated on a symmetrical current basis-preferred ratings and related required capabilities. American National Standards Institute, Inc, USA.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Dubois, D., Prade, H., 1990. Rough fuzzy sets and fuzzy rough sets. Int. J. Gen. Syst. 17, 191–209.

Fan, Y., Xiaoguang, H.H., 2012. Research on the mechanical state parameter extraction method of high voltage circuit breakers. In: The 10th IEEE International Conference on Industrial Informatics (INDIN 2012), pp. 1062–1066.

Fnineche, C., Aitken, O., 2012. Investigations on some parameters influencing the current commutation in the circuit breakers. In: The 26th International Conference on Electrical Contacts (ICEC 2012), pp. 497–501.

García, S., Fernández, A., Luengo, J., Herrera, F., 2009. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput. 13 (10), 959–977.

Garzon, R.D., 2002. High Voltage Circuit Breakers. Design and Applications. Marcel Dekker, Inc., USA.

Han, H., Wang, W.Y., Mao, B.H., 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, International Conference on Intelligent Computing, ICIC. Springer-Verlag, pp. 878–887.

He, H., García, E.A., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21 (9), 1263–1284.

Huang, J., Ling, C.X., 2005. Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. Knowl. Data Eng. 17 (3), 299–310.

Lindquist, T., Bertling, L., Eriksson, R., 2008. Circuit breaker failure data and reliability modelling. IET Gener. Transm. Distrib., 813–820.

López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. 250, 113–141.

Napierala, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples. In: Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science, vol. 6086, 2010, pp. 158–167.

Pawlak, Z., 1982. Rough sets. Int. J. Comput. Inf. Sci. 11, 145–172.

Quinlan, J.R., 1993. C4.5 Programs for Machine Learning. Morgan Kaufmann, CA.

Ramentol, E., Caballero, Y., Bello, R., Herrera, F., 2012a. SMOTE-RSB$_*$: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. Int. J. Knowl. Inf. Syst. 33, 245–265.

Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Chris Cornelis, Herrera, F., 2012b. SMOTE-FRST: a new resampling method using fuzzy rough set theory. In: Proceedings of FLINS2012, pp. 800–805.

Rudd, S., Catterson, V.M., McArthur, S.D.J., Johnstone, C., 2011. Circuit breaker prognostics using sf6 data. In: IEEE Power and Energy Society General Meeting, 2011, pp. 1–6.

Runde, M., Sölver, C.E., Carvalho, A., Cormenzana, M.L., Furuta, H., Grieshaber, W., Hyrczak, A., Kopejtkova, D., Krone, J.G., Kudoke, M., Makareinis, D., Martins, J.F., 2012. Tb no. 509 - wg a3.06 final report of the 2004–2007. In: International Enquiry on Reliability of High Voltage. ELECTRA 264.

Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification of imbalanced data: a review. Int. J. Pattern Recognit. Artif. Intell. 23 (4), 687–719.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A.F.M., Liu, B., Yu, P.S., Zhou, Z., Steinbach, M., Hand, D.J., Steinberg, D., 2008. Top 10 algorithms in data mining. Knowl. Inf. Syst. 14 (1), 1–37.

Zadeh, L., 1965. Fuzzy sets. Inf. Control 8, 338–353.