Soft Comput (2011) 15:1909–1936
DOI 10.1007/s00500-010-0625-8

FOCUS

Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling

Julián Luengo · Alberto Fernández · Salvador García · Francisco Herrera

Published online: 20 June 2010

Abstract In the classification framework there are problems in which the number of examples per class is not equitably distributed, formerly known as imbalanced data sets. This situation is a handicap when trying to identify the minority classes, as the learning algorithms are not usually adapted to such characteristics. An usual approach to deal with the problem of imbalanced data sets is the use of a preprocessing step. In this paper we analyze the usefulness of the data complexity measures in order to evaluate the behavior of undersampling and oversampling methods. Two classical learning methods, C4.5 and PART, are considered over a wide range of imbalanced data sets built from real data. Specifically, oversampling techniques and an evolutionary undersampling one have been selected for the study. We extract behavior patterns from the results in the data complexity space defined by the measures, coding them as intervals. Then, we derive rules from the intervals that describe both good or bad behaviors of C4.5 and PART for the different preprocessing approaches, thus obtaining a complete characterization of the data sets and the differences between the oversampling and undersampling results.

J. Luengo (✉) · F. Herrera
Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
e-mail: julianlm@decsai.ugr.es

F. Herrera
e-mail: herrera@decsai.ugr.es

A. Fernández · S. García
Department of Computer Science, University of Jaén, 23071 Jaén, Spain
e-mail: alberto.fernandez@ujaen.es

S. García
e-mail: sglopez@ujaen.es

Keywords Classification · Evolutionary algorithms · Data complexity · Imbalanced data sets · Oversampling · Undersampling · C4.5 · PART

1 Introduction

The problem of imbalanced classes is one of the problems that emerged when Machine Learning (ML) reached maturity, being a widely used technology in the world of business, industry, and scientific research. Its importance grew as researchers realized that the analyzed data sets contained many more instances or examples from a class or classes with respect to the remaining ones (Chawla et al. 2004), and the obtained classification models performed below the desired threshold in the minority classes. Currently it is considered as a challenge by the Data Mining Community (Yang and Wu 2006).

The main handicap of this type of problem is that standard learning algorithms minimize a global measure of error, and this supposes a bias towards the majority class (Sun et al. 2009). Hence, to tackle this issue, the use of preprocessing techniques is a good solution in order to balance the training set before the learning process (Batista et al. 2004; Estabrooks et al. 2004; He and Garcia 2009).

On the other hand, it is well known that the prediction capabilities of the classifiers are also dependent on the problem's characteristics as well. An emergent field, that uses a set of complexity measures applied to the problem to describe its difficulty, has recently arisen. These measures try to capture different aspects or sources of complexity which are considered complicated to the classification task (Basu and Ho 2006). Studies of data complexity metrics applied to particular classification's algorithms can be found (Basu and Ho 2006; Bernadó-Mansilla and Ho 2005;

Baumgartner and Somorjai 2006; Sánchez et al. 2007; García et al. 2009c).

Our objective is to show that the data complexity measures are adequate to analyze the effect of the preprocessing in imbalanced data for classification. We will consider two main preprocessing approaches: oversampling and undersampling of the data. We will identify the regions in the data complexity space in which the preprocessing works well, and the bad performance regions as well. In a related approach (García et al. 2008) the relationship between the Imbalance Ratio (IR) (Orriols-Puig and Bernadó-Mansilla 2008) and the overlapping of the class labels with respect to the performance of several learning methods was studied. However, no preprocessing approach was analyzed in this study.

In order to analyze the oversampling and undersampling by means of the data complexity measures, we will use the "Synthetic Minority Over-sampling Technique" (SMOTE) and its variant with the Wilson's Edited Nearest Neighbor Rule (ENN) as representative oversampling preprocesing methods. SMOTE is a classical oversampling method, whereas SMOTE-ENN was shown in Batista et al. (2004) to achieve a very good behavior with the C4.5 decision tree. The Evolutionary Undersampling with CHC (EUSCHC) method proposed by García and Herrera (2009a) will be included as a representative evolutionary undersampling approach. It has been proved to be very competitive with respect to SMOTE and SMOTE-ENN, and to be the best among other representative techniques from the family of undersampling as shown in their study.

The effect of these three preprocessing techniques will be analyzed with respect to two well-known learning methods. The first one is the C4.5 decision tree (Quinlan 1993), which has been used in many recent analyses of imbalanced data (Su and Hsiao 2007; García et al. 2009b; Drown et al. 2009). The second one is the PART algorithm (Frank and Witten 1998) also used by García et al. (2009b) in the imbalanced data framework.

Following the methodology proposed by Luengo and Herrera (2010), three of the data complexity measures proposed by Ho and Basu (2002) are informative in order to create intervals of their values over the data sets in which C4.5 and PART perform well or bad on average after applying each preprocessing technique. We will use a large collection of data sets with different degrees of imbalance from the UCI repository (Asuncion and Newman 2007) in order to sample the data complexity space. Then we will formulate rules for such intervals, comparing the support (number of data sets included in the interval) and average learning method's performance for the three preprocessing techniques. Therefore, we can evaluate the performance of C4.5 and PART when using the oversampling and undersampling approaches by means of observing differences in the covered data sets by the obtained rules. These differences will provide information about the behavior of the three considered preprocessing approaches for C4.5 and PART.

This paper is organized as follows: first, Sect. 2 presents the problem of imbalanced data sets, describing its features, the preprocessing methods used, and the metric we have employed in this context. Next, Sect. 3 introduces the data complexity metrics we have used along with recent studies in the topic. In Sect. 4 the background on the use of data complexity for imbalanced data and the experimental framework used in this study are presented. In Sect. 5 the analyses of the methodology used and the experimental results are performed. Section 6 summarizes and concludes the work. Appendix 1 contains the figures with the intervals extracted in the study. Appendix 2 depicts the tables with the average results obtained for each data set in the study.

2 Imbalanced data sets in classification

In this section, the problem of imbalanced data sets in Sect. 2.1 is introduced first. The SMOTE and SMOTE-ENN are described in Sect. 2.2. The EUSCHC method is described in Sect. 2.3. Finally, Sect. 2.4 presents the evaluation metrics for this kind of classification problems.

2.1 The problem of imbalanced data sets

In the classification problem field, the scenario of imbalanced data sets appears when the number of examples that represent the different classes are very different among them (Chawla et al. 2004). We focus on the binary-class imbalanced data sets, where there is only one positive (minority) and one negative (majority) class. In this work we consider the IR (Orriols-Puig and Bernadó-Mansilla 2008), defined as the number of negative class instances divided by the number of positive class instances. The IR can be used to organize the different data sets according to their degree of imbalance.

Most of the learning algorithms aim to obtain a model with a high prediction accuracy and a good generalization capability. However, this inductive bias towards such a model supposes a serious challenge with the classification of imbalanced data (Sun et al. 2009). First, if the search process is guided by the standard accuracy rate, it benefits the covering of the majority examples; second, classification rules that predict the positive class are often highly specialized and thus their coverage is very low; hence they are discarded in favor of more general rules, i.e., those that predict the negative class. Furthermore, it is not easy to distinguish between noise examples and minority class

examples and they can be completely ignored by the classifier.

In recent years regarding real world domains the importance of the imbalance learning problem grows since it is a recurring problem in many applications, such as remote-sensing (Williams et al. 2009), pollution detection (Lu and Wang 2008) and especially medical diagnosis (Kilic et al. 2007; Mazurowki et al. 2008; Celebi et al. 2007; Peng and King 2008).

For this reason, a large number of approaches have been previously proposed to deal with the class imbalance problem. These approaches can be categorized into two groups: the internal approaches that create new algorithms or modify existing ones to take the class imbalance problem into consideration (Barandela et al. 2003; Diamantini an Potena 2009) and external approaches that preprocess the data in order to diminish the effect cause by their class imbalance (Fernández et al. 2008; Drown et al. 2009; Tang et al. 2009). Furthermore, cost-sensitive learning solutions incorporating both the data and algorithmic level approaches assume higher misclassification costs with samples in the minority class and seek to minimize the high cost errors (Domingos 1999; Sun et al. 2007; Zhou and Liu 2006).

The great advantage of the external approaches is that they are more versatile, since their use is independent of the classifier selected. Furthermore, we may preprocess all data sets before-hand in order to use them to train different classifiers. In this manner, the computation time needed to prepare the data is only used once.

2.2 Oversampling approaches: the SMOTE and SMOTE-ENN algorithms

As mentioned before, applying a preprocessing step in order to balance the class distribution is a positive solution to the imbalance data set problem (Batista et al. 2004). Specifically, in this work we have chosen an over-sampling method which is a reference in this area: the SMOTE algorithm (Chawla et al. 2002), and a variant called SMOTE-ENN (Batista et al. 2004).

In SMOTE the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of oversampling required, neighbors from the k-nearest neighbors are randomly chosen. This process is illustrated in Fig. 1, where x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbors and r_1 to r_4 the synthetic data points created by the randomized interpolation. The implementation employed in this work uses the euclidean distance, and balances both classes to the 50% distribution.

Synthetic samples are generated in the following way: take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. An example is detailed in Fig. 2.

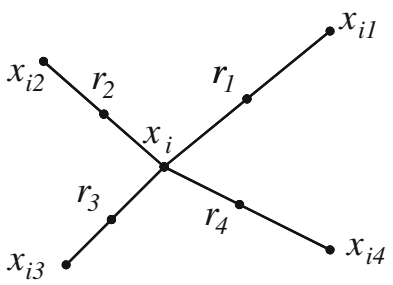

Fig. 1 An illustration on how to create the synthetic data points in the SMOTE algorithm

```
Consider a sample (6,4) and let (4,3) be its nearest neighbour.
(6,4) is the sample for which k-nearest neighbours are being
identified (4,3) is one of its k-nearest neighbours.
Let: f1_1 = 6 f2_1 = 4, f2_1 - f1_1 = -2
f1_2 = 4 f2_2 = 3, f2_2 - f1_2 = -1
The new samples will be generated as
f1',f2') = (6,4) + rand(0-1) * (-2,-1)
rand(0-1) generates a random number between 0 and 1.
```

Fig. 2 Example of the SMOTE application

In short, its main idea is to form new minority class examples by interpolating between several minority class examples that lie together. Thus, the overfitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

On the other hand, class clusters could not be well defined since some minority class examples might be invading the majority class space. This situation can occur when interpolating minority class examples can expand the minority class clusters, introducing artificial minority class examples too deeply in the majority class space. Inducing a classifier under such a situation can lead to overfitting. Batista et al. proposed to apply ENN to the over-sampled training set as a data cleaning method. ENN removes any example whose class label differs from the class of at least two of its three nearest neighbors. Thus, any example that is misclassified by its three nearest neighbors is removed from the training set. We refer to this technique as SMOTE-ENN.

2.3 Undersampling approach: the EUSCHC algorithm

Instead of creating new examples of the minority class, the undersampling approach selects a subset of the examples which represents the initial problem better, and avoids the

Table 1 Confusion matrix for a two-class problem

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

bias to the minority class by removing redundant examples. This approach has also the advantage of creating a reduced set of examples to the induction process, making it less costly.

The search for the optimal subset of examples which affect the learning method's performance the less can be considered as a search problem in which evolutionary algorithms can be applied. In this work the EUSCHC technique (García and Herrera 2009a) is considered. EUSCHC is an evolutionary undersampling technique, which removes redundant noisy and redundant examples.

EUSCHC uses the well-known CHC evolutionary algorithm (Eshelman 1991) as a base for the selection of the subset of the examples, considering a binary codification for the subset membership. EUSCHC can also use any performance measure as fitness, weighting positively the correctly classified examples which are outside the selected subset.

2.4 Evaluation in imbalanced domains

The measures of the quality of classification are built from a confusion matrix (shown in Table 1) which records correctly and incorrectly recognized examples for each class. The most used empirical measure, accuracy (1), does not distinguish between the number of correct labels of different classes, which in the framework of imbalanced problems may lead to erroneous conclusions. As a classical example, if the ratio of imbalance presented in the data is 10:100, i.e., there is ten positive instance versus ninety negatives, then a classifier that obtains an accuracy rate of 90% is not truly accurate if it does not correctly cover the single minority class instance

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{1}$$

One appropriate metric that could be used to measure the performance of classification over imbalanced data sets is the Receiver Operating Characteristic (ROC) graphics (Bradley 1997). In these graphics the tradeoff between the benefits and costs can be visualized. They show that any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) (Huang and Ling 2005) corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary for the performance of learning algorithms.

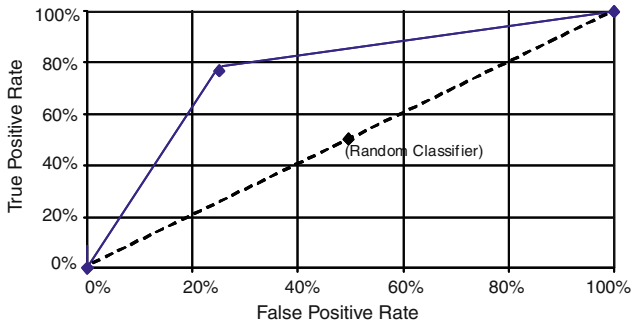

Fig. 3 Example of a ROC plot. Two classifiers are represented: the *solid line* is a good performing classifier whereas the *dashed line* represents a random classifier

The way to build the ROC space is to plot on a two-dimensional chart the true positive rate (Y axis) against the false positive rate (X axis) as shown in Fig. 3. The points (0, 0) and (1, 1) are trivial classifiers in which the output class is always predicted as negative and positive, respectively, while the point (0, 1) represents perfect classification. To compute the AUC we just need to obtain the area of the graphic as

$$\mathrm{AUC} = \frac{1 + \mathrm{TP}_{\mathrm{rate}} - \mathrm{FP}_{\mathrm{rate}}}{2}, \tag{2}$$

where $\mathrm{TP}_{\mathrm{rate}}$ and $\mathrm{FP}_{\mathrm{rate}}$ are the percentage of correctly and wrongly classified cases belonging to the positive class, respectively.

3 Data complexity

In this section we first present a short review on recent studies in data complexity in Sect. 3.1. The data complexity measures considered in this paper are described in Sect. 3.2.

3.1 Recent studies on data complexity measures

One direct approach to deal with data complexity is to obtain indicators about it by means of some measures. In particular, Ho and Basu (2002) proposed some complexity measures for binary classification problems, gathering metrics of three types: overlaps in feature values from different classes; separability of classes and measures of geometry, topology, and density of manifolds. Shortly after, Singh (2003) offered a review of data complexity measures of different nature [from Bayes error-based to nonparametric methods of Ho and Basu (2002)] and proposed two new ones.

Table 2 Data complexity measures names and acronyms proposed by Ho and Basu

Type	Id.	Description
Measures of overlaps in feature values from different classes	F1	Maximum Fisher's discriminant ratio
	F2	Error rate of linear classifier by linear programming
	F3	Maximum (individual) feature efficiency
Measures of separability of classes	L1	Minimized sum of error distance by linear programming
	L2	Error rate of linear classifier by linear programming
	N1	Fraction of points on class boundary
	N2	Ratio of average intra/inter class NN distance Ms Cercanos intra/inter-clases
	N3	Error rate of 1NN classifier
Measures of geometry, topology and density of manifolds	L3	Nonlinearity of linear classifier by linear programming
	N4	Non-linearity of 1NN classifier
	T1	Fraction of points with associated adherence subsets retained
	T2	Average number of points per dimension

These two studies, especially (Ho and Basu 2002), have been widely used afterwards. In the field of classification, we can find recent works using the measures of Ho and Basu. Bernadó-Mansilla and Ho (2005) investigated the domain of competence of XCS by means of a methodology that characterizes the complexity of a classification problem by a set of geometrical descriptors. Li et al. (2005) analyzed some omnivariate decision trees using the measure of complexity based in data density. Baumgartner and Somorjai (2006) defined specific measures for regularized linear classifiers. Sánchez et al. (2007) analyzed the effect of the data complexity in the nearest neighbors classifier, while García et al. (2009c) studied the relationship of the Fisher's discriminant ratio with respect to an evolutionary instance selection method in the classification task.

Focusing on how some data complexity measures affect the practical accuracy of these classification algorithms, they show which data complexity measures appear to better describe the behavior of the classifiers. More recently, Luengo and Herrera (2010) analyzed the domains of competence of a Fuzzy Rule-Based Classification System with respect to eight data complexity measures from Ho and Basu (2002). In the latter work, descriptive rules of good and bad behaviors of the method were defined based on the characteristics of the data sets characterized by the data complexity measures.

On the other hand, there exist proposals which do not use the measures of Ho and Basu with respect to a classification method directly, but taking into account a preprocessing technique. Dong and Kothari (2003) proposed a feature selection algorithm based on a complexity measure defined by Ho and Basu. Mollineda et al. (2005) extend some of Ho and Basu's measure definitions for problems with more than two classes. They analyzed these generalized measures in two classic Prototype Selection algorithms and remarked that Fisher's discriminant ratio is the most effective for Prototype Selection. Considering the opposite case, Kim and Oommen (2009) analyzed how to use prototype selection in order to decrease the computation time of several data complexity measures, without severely affecting the outcome with respect to the complete data set.

3.2 Data complexity measures

As we have mentioned, data complexity measures are a set of metrics that quantify characteristics which imply some difficulty to the classification task. In our analysis we will initially consider the 12 measures used in Ho and Basu (2002) for standard classification in the imbalanced framework used in this paper. The 12 measures are summarized in Table 2.

In our analysis, only F1, N4, and L3 measures of the 12 presented in Table 2 proved to be informative following the methodology described in Sect. 5.1 and observed in the experimental results in Sect. 5.2. The description of these three measures is included next.

F1: maximum Fisher's discriminant ratio. Fisher's discriminant ratio for one feature dimension is defined as

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

where μ_1, μ_2, σ_1^2, σ_2^2 are the means and variances of the two classes, respectively, in that feature dimension. We compute f for each feature and take the maximum as measure F1. For a multidimensional problem, not all features have to contribute to class discrimination. The problem is easy as long as there exists one discriminating feature. Therefore, we can just take the maximum f over all feature dimensions in discussing class separability.

L3: nonlinearity of linear classifier by LP. Hoekstra and Duin (1996) proposed a measure for the nonlinearity of a

classifier with respect to a given data set. Given a training set, the method first creates a test set by linear interpolation (with random coefficients) between randomly drawn pairs of points from the same class. Then the error rate of the classifier (trained by the given training set) on this test set is measured. Here, we use such a nonlinearity measure for the linear classifier defined for L1. In particular, we consider a Support Vector Machine with a linear Kernel, which acts as a linear discriminant in this case. This measure is sensitive to the smoothness of the classifier's decision boundary as well as the overlap of the convex hulls of the classes. For linear classifiers and linearly separable problems, it measures the alignment of the decision surface with the class boundary. It carries the effects of the training procedure in addition to those of the class separation.

N4: nonlinearity of 1NN classifier. This measure follows the same procedure described for L3. In the case of N4, error is calculated for a nearest neighbor classifier. This measure is for the alignment of the nearest-neighbor boundary with the shape of the gap or overlap between the convex hulls of the classes.

4 On the use of data complexity measures for imbalanced data

In this section we first present the imbalanced data considered in this study and the configuration used for the calculation of the data complexity measures in Sect. 4.1, then reasons which motivates their use are introduced in Sect. 4.2.

4.1 Data sets and configuration of the methods

In order to analyze the preprocessing of the SMOTE, SMOTE-ENN and EUSCHC methods, we have selected 44 data sets from UCI repository (Asuncion and Newman 2007). The data are summarized in Table 3, showing the number of examples (#Ex.), attributes (#Atts.), name of each class (minority and majority), class attribute distribution, IR and F1, N4, and L3 data complexity values associated.

For every binary data set generated, we computed the 12 data complexity measures of Ho and Basu (2002) over the complete data set before preprocessing and splitting the data. Table 3 contains the F1, N4, and L3 measures' values for each original data set, as they proved to be the informative ones in our study. This will provide us information about the nature of the complete data set before preprocessing and applying the validation scheme.

The calculation of the data complexity measures supports some variants. The particular details of the computation of the measures that we have followed are detailed next.

- The instances with missing values are discarded previously to the measures calculation.
- The measures calculation over the data sets is performed with the original values, without any kind of normalization.
- The distance function used for continuous values is the normalized Euclidean distance function, i.e., the distance of each attribute is normalized by its range.
- The distance function used for nominal values is the overlap distance function. That is, if two nominal attributes are equal, the distance between them is 0. Otherwise the distance is 1.

It is essential to maintain this configuration for all the data sets, as any change in it has proven to produce changes in the estimated complexity obtained. Therefore, alterations in the distance functions used, for example, can disturb the analysis done and the conclusions obtained from it.

In order to carry out the different experiments we consider a *5-folder cross-validation model*, i.e., 5 random partitions of data with a 20%, and the combination of 4 of them (80%) as training and the remaining one as test. For each data set we consider the average results of the five partitions.

Then, in order to reduce the effect of imbalance, we will employ the SMOTE and SMOTE-ENN preprocessing method for all our experiments balancing both classes to the 50% distribution in the training partition (Batista et al. 2004). EUSCHC aims at reducing the data in the training partition as much as possible while the performance in AUC is not decreased.

The C4.5 and PART algorithms were run using KEEL[1] software (Alcalá-Fdez et al. 2009) following the recommended parameter values given in this platform, which must also correspond to the ones given by the authors in the original papers:

- For C4.5 the minimum number of item-sets per leaf was set to 2, and a pruning step is applied for the final tree with a confidence level of 0.25.
- For PART the minimum number of item-sets per leaf was also set to 2, and a pruning step is applied for the final tree with a confidence level of 0.25 as well.

In Table 4 we have summarized the global average for training and test AUC and the corresponding standard deviation obtained by C4.5 with SMOTE, SMOTE-ENN and EUSCHC preprocessing. These two tables are mean to be used for further reference in the analysis of the behavior of the preprocessing techniques in the following sections. As a general comment, SMOTE and SMOTE-ENN produce C4.5 and PART have a better training adjustment, while EUSCHC allows C4.5 and PART to generalize

[1] http://keel.es.

Table 3 Summary description for imbalanced data sets

Data set	#Ex.	#Atts.	Class (min., maj.)	%Class (min.; maj.)	IR	F1	N4	L3
Glass1	214	9	(build-win-non oat-proc; remainder)	(35,51; 64,49)	1.82	0.1897	0.3084	0.5000
Ecoli0vs1	220	7	(im; cp)	(35,00; 65,00)	1.86	9.7520	0.0136	0.1182
Wisconsin	683	9	(malignant; benign)	(35,00; 65,00)	1.86	3.5680	0.0432	0.0066
Pima	768	8	(tested-positive; tested-negative)	(34,84; 66,16)	1.90	0.5760	0.2754	0.5000
Iris0	150	4	(Iris-Setosa; remainder)	(33,33; 66,67)	2.00	16.8200	0.0000	0.0000
Glass0	214	9	(build-win-oat-proc; remainder)	(32,71; 67,29)	2.06	0.6492	0.2009	0.5000
Yeast1	1484	8	(nuc; remainder)	(28,91; 71,09)	2.46	0.2422	0.3201	0.5000
Vehicle1	846	18	(Saab; remainder)	(28,37; 71,63)	2.52	0.3805	0.1761	0.2311
Vehicle2	846	18	(Bus; remainder)	(28,37; 71,63)	2.52	0.1691	0.3304	0.3682
Vehicle3	846	18	(Opel; remainder)	(28,37; 71,63)	2.52	0.1855	0.3747	0.3511
Haberman	306	3	(Die; Survive)	(27,42; 73,58)	2.68	0.1850	0.3431	0.4967
Glass0123vs456	214	9	(non-window glass; remainder)	(23,83; 76,17)	3.19	3.3240	0.0561	0.3294
Vehicle0	846	18	(Van; remainder)	(23,64; 76,36)	3.23	1.1240	0.1734	0.1219
Ecoli1	336	7	(im; remainder)	(22,92; 77,08)	3.36	2.6500	0.1265	0.5000
New-thyroid2	215	5	(hypo; remainder)	(16,89; 83,11)	4.92	3.5790	0.0233	0.2791
New-thyroid1	215	5	(hyper; remainder)	(16,28; 83,72)	5.14	3.5790	0.0209	0.2721
Ecoli2	336	7	(pp; remainder)	(15,48; 84,52)	5.46	1.8260	0.0685	0.5000
Segment0	2308	19	(brickface; remainder)	(14,26; 85,74)	6.01	1.7980	0.0358	0.5000
Glass6	214	9	(headlamps; remainder)	(13,55; 86,45)	6.38	2.3910	0.0537	0.5000
Yeast3	1484	8	(me3; remainder)	(10,98; 89,02)	8.11	2.7510	0.1122	0.5000
Ecoli3	336	7	(imU; remainder)	(10,88; 89,12)	8.19	1.5790	0.1652	0.5000
Page-blocks0	5472	10	(remainder; text)	(10,23; 89,77)	8.77	0.5087	0.2069	0.3332
Yeast2vs4	514	8	(cyt; me2)	(9,92; 90,08)	9.08	1.5790	0.1333	0.5000
Yeast05679vs4	528	8	(me2; mit, me3, exc, vac, erl)	(9,66; 90,34)	9.35	1.0510	0.2509	0.5000
Vowel0	988	13	(hid; remainder)	(9,01; 90,99)	10.10	2.4580	0.2034	0.5000
Glass016vs2	192	9	(ve-win-oat-proc; build-win-oat-proc, build-win-non oat-proc,headlamps)	(8,89; 91,11)	10.29	0.2692	0.2891	0.5000
Glass2	214	9	(Ve-win-oat-proc; remainder)	(8,78; 91,22)	10.39	0.3952	0.3364	0.5000
Ecoli4	336	7	(om; remainder)	(6,74; 93,26)	13.84	3.2470	0.0506	0.5000
Yeast1vs7	459	8	(nuc; vac)	(6,72; 93,28)	13.87	12.9700	0.0016	0.0019
Shuttle0vs4	1829	9	(Rad Flow; Bypass)	(6,72; 93,28)	13.87	0.3534	0.3137	0.5000
Glass4	214	9	(containers; remainder)	(6,07; 93,93)	15.47	1.4690	0.1285	0.5000
Page-blocks13vs2	472	10	(graphic; horiz.line, picture)	(5,93; 94,07)	15.85	1.5470	0.0540	0.0678
Abalone9vs18	731	8	(18; 9)	(5,65; 94,25)	16.68	0.6320	0.3324	0.5000
Glass016vs5	184	9	(tableware; build-win-oat-proc, build-win-non oat-proc, headlamps)	(4,89; 95,11)	19.44	1.8510	0.0788	0.5000
Shuttle2vs4	129	9	(Fpv Open; Bypass)	(4,65; 95,35)	20.50	12.1300	0.0155	0.0000
Yeast1458vs7	693	8	(vac; nuc, me2, me3, pox)	(4,33; 95,67)	22.10	0.1757	0.3752	0.5000
Glass5	214	9	(tableware; remainder)	(4,20; 95,80)	22.81	1.0190	0.0724	0.5000
Yeast2vs8	482	8	(pox; cyt)	(4,15; 95,85)	23.10	1.1420	0.2261	0.5000
Yeast4	1484	8	(me2; remainder)	(3,43; 96,57)	28.41	0.7412	0.2342	0.5000
Yeast1289vs7	947	8	(vac; nuc, cyt, pox, erl)	(3,17; 96,83)	30.56	0.3660	0.3627	0.5000
Yeast5	1484	8	(me1; remainder)	(2,96; 97,04)	32.78	4.1980	0.1216	0.5000
Ecoli0137vs26	281	7	(pp, imL; cp, im, imU, imS)	(2,49; 97,51)	39.15	1.9670	0.1701	0.5000
Yeast6	1484	8	(exc; remainder)	(2,49; 97,51)	39.15	2.3020	0.1157	0.5000
Abalone19	4174	8	(19; remainder)	(0,77; 99,23)	128.87	0.5295	0.4534	0.5000

Table 4 Global average Training and Test AUC for C4.5

	Global % AUC Training	Global % AUC Test
C4.5 with SMOTE preprocessing	0.9546 ± 0.0551	0.8217 ± 0.1375
C4.5 with SMOTE-ENN preprocessing	0.9438 ± 0.0635	0.8362 ± 0.1309
C4.5 with EUSCHC preprocessing	0.9241 ± 0.0859	0.8914 ± 0.1035

Table 5 Global average Training and Test AUC for PART

	Global % AUC Training	Global % AUC Test
PART with SMOTE preprocessing	0.9440 ± 0.0727	0.8298 ± 0.1368
PART with SMOTE-ENN preprocessing	0.9353 ± 0.0727	0.8372 ± 0.1313
PART with EUSCHC preprocessing	0.9172 ± 0.0796	0.8900 ± 0.0899

better. In Table 5 we present the same AUC averages for PART. The complete tables of results are shown in Appendix 2.

4.2 Motivation of the use of complexity measures

In Sect. 2, the necessity of the use of instance preprocessing in the framework of imbalanced data has been shown. The IR measure has been also used in order to classify the different imbalanced problems based on their imbalanced degree. However, we have observed empirically that this measure has not a clear relationship with the performance obtained with the preprocessing techniques. In this sense, Fig. 4 depicts the results for C4.5 in the case of preprocessing the 44 data sets with SMOTE and EUSCHC, sorting the data sets by their IR value. We can observe that the good and bad results of both learning methods with respect to the preprocessing are not related with the IR value, nor the improvements achieved with such preprocessing step.

Therefore, the use of the IR as a unique measure to identify the improvement of the preprocessing appears to be insufficient, and we need to consider other measures to characterize the good or bad behavior of the preprocessing, like the data complexity measures presented in Sect. 3.2.

To the best of our knowledge there is no analysis on the relationship of the data complexity and the application of preprocessing techniques for imbalanced data. García et al. (2008) built a bunch on synthetic data sets with a wide range of overlapping present in the two classes. Using this framework, the response of local and global learning methods (the *k*-NN classifier and several others, C4.5 among them) is studied when varying the IR and the overlapping between the class labels. Albeit they do not explicitly used the data complexity measures of Ho and Basu (2002), the class overlapping and IR can be considered as related to them. Results showed that the more represented class in overlapped regions tends to be better classified by methods based on global learning, while the less class represented in such regions tends to be better classified by local methods.

However, in García et al.'s (2008) study no preprocessing was performed. In this paper we will analyze the mentioned the undersampling and oversampling preprocessing approaches by means of the data complexity measures.

We emphasize that this study does not attempt to establish the best preprocessing method for a given

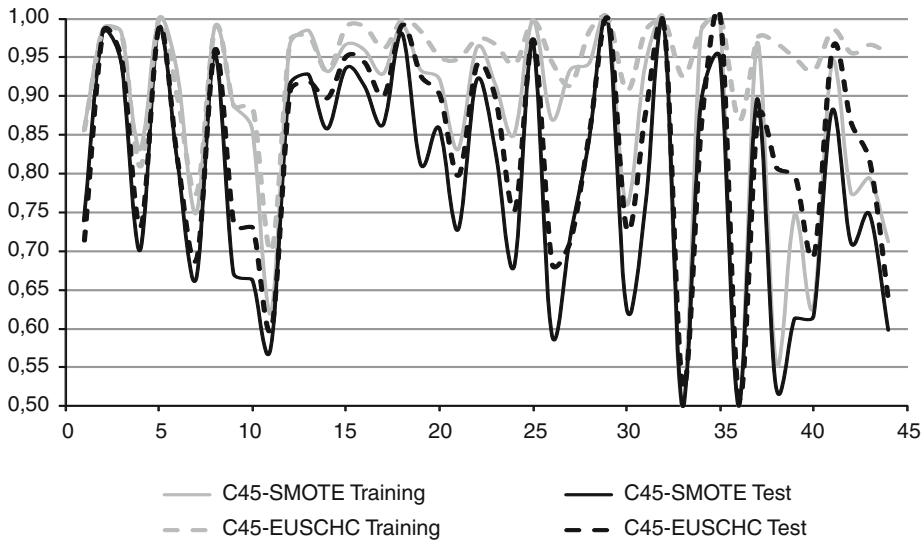

Fig. 4 C4.5 AUC in Training/Test sorted by IR

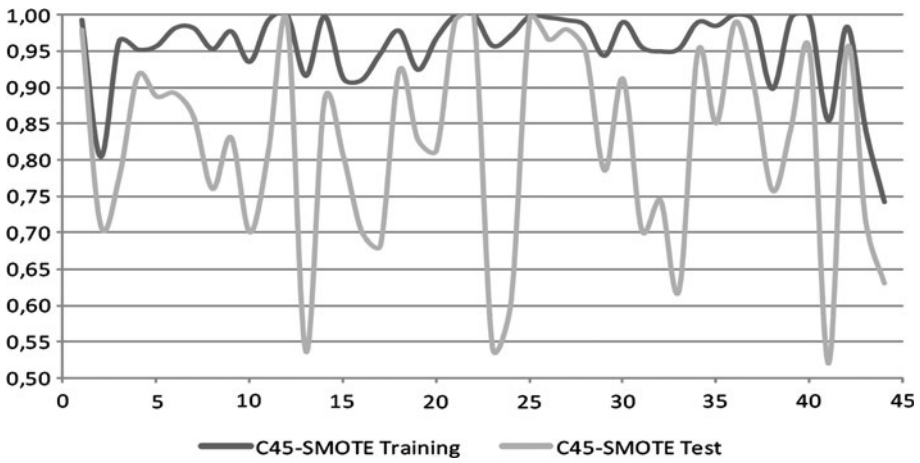

Fig. 5 C4.5 AUC with SMOTE in Training/Test sorted by F2

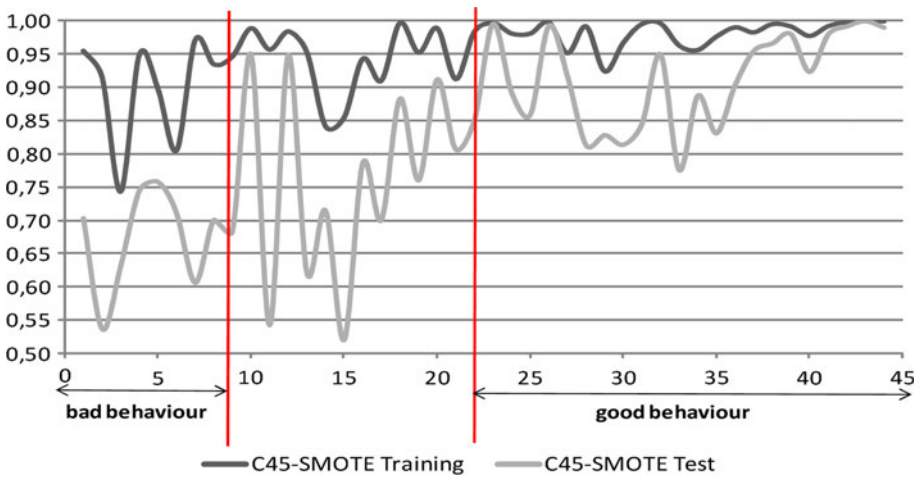

Fig. 6 C4.5 AUC with SMOTE in Training/Test sorted by F1

problem. This estimation problem has been already formalized as a new learning problem in the Meta-Learning approach (MetaL) (Brazdil et al. 2009). The MetaL approach faces two important drawbacks:

- How to represent an ML problem instance was tackled using diverse descriptors, e.g., number of examples, number of attributes, percentage of missing values, and landmarkers (Pfahringer et al. 2000). The difficulty is due to the fact the descriptors must take into account the example distribution, which is not easily achieved in most cases.
- A second difficulty concerns the selection of the ML problem instances. Kalousis (2002) indicates that the representativity of the problems and the perturbation induce strong biases in the Metal classifier.

Due to the several difficulties already studied in the specialized literature, the attempt to indicate the best preprocessing method is out of the scope of this paper.

Table 6 Significant intervals for C4.5 and PART

With SMOTE and SMOTE-ENN Interval	With EUSCHC Interval	Behavior
C4.5		
F1 $\geq$ 1.469	F1 $\geq$ 0.6492	Good
N4 $\leq$ 0.2069	N4 $\leq$ 0.2509	Good
L3 $\leq$ 0.3332	L3 $\leq$ 0.3332	Good
F1 $\leq$ 0.366	F1 $\leq$ 0.3534	Bad
N4 $\geq$ 0.2261	N4 $\geq$ 0.2754	Bad
PART		
F1 $\geq$ 1.469	F1 $\geq$ 0.632	Good
N4 $\leq$ 0.2069	N4 $\leq$ 0.2509	Good
L3 $\leq$ 0.3332	L3 $\leq$ 0.3332	Good
F1 $\leq$ 0.366	F1 $\leq$ 0.3534	Bad
N4 $\geq$ 0.2261	N4 $\geq$ 0.2754	Bad

5 Analysis of the influence of preprocessing in imbalanced data

In this study, our aim is to analyze the suitability of the use of data complexity measures to evaluate the behavior of SMOTE, SMOTE-ENN, and EUSCHC using C4.5 and PART in the scenario of imbalanced data sets.

Table 7 Rules with one metric obtained from the intervals for C4.5

Id.	Rule	Preprocess	Support	Avg. AUC Train	Train diff.	Avg. AUC Test	Test diff.
R1+	If F1 $\geq$ 1.469 then good behavior	SMOTE	52.27	0.9826	0.028	0.9103	0.0886
		SMOTE-ENN		0.9785	0.0347	0.9234	0.0872
	If F1 $\geq$ 0.6482 then good behavior	EUSCHC	65.91	0.9687	0.0515	0.9450	0.1632
R2+	If L3 $\leq$ 0.3332 then good behavior	SMOTE	27.27	0.9929	0.0383	0.9641	0.1424
		SMOTE-ENN		0.9876	0.0438	0.9610	0.1248
		EUSCHC		0.9877	0.0636	0.9688	0.0774
R3+	If N4 $\leq$ 0.2069 then good behavior	SMOTE	63.63	0.9823	0.0277	0.9077	0.086
		SMOTE-ENN		0.9756	0.0318	0.9196	0.0834
	If N4 $\leq$ 0.2509 then good behavior	EUSCHC	70.45	0.9692	0.0451	0.9460	0.0546
R1–	If F1 $\leq$ 0.366 then bad behavior	SMOTE	20.45	0.9021	–0.0525	0.6748	–0.1469
		SMOTE-ENN		0.8613	–0.0825	0.6762	–0.1600
	If F1 $\leq$ 0.3534 then bad behavior	EUSCHC	18.18	0.8186	–0.1055	0.7519	–0.1395
R2–	If N4 $\geq$ 0.2261 then bad behavior	SMOTE	36.36	0.9062	–0.0484	0.6712	–0.1505
		SMOTE-ENN		0.8881	–0.0557	0.6903	–0.1459
	If N4 $\geq$ 0.2754 then bad behavior	EUSCHC	29.55	0.8166	–0.1075	0.7613	–0.1301

Table 8 Rules with one metric obtained from the intervals for PART

Id.	Rule	Preprocess	Support	Avg. AUC Train	Train diff.	Avg. AUC Test	Test diff.
R1+	If F1 $\geq$ 1.469 then good behavior	SMOTE	52.27	0.9817	0.0377	0.9194	0.0896
		SMOTE-ENN		0.9764	0.0411	0.9259	0.0887
	If F1 $\geq$ 0.632 then good behavior	EUSCHC	68.18	0.9538	0.0366	0.9322	0.0422
R2+	If L3 $\leq$ 0.3332 then good behavior	SMOTE	27.27	0.9932	0.0492	0.9646	0.1348
		SMOTE-ENN		0.9857	0.0504	0.9687	0.1315
		EUSCHC		0.9716	0.0544	0.9514	0.0614
R3+	If N4 $\leq$ 0.2069 then good behavior	SMOTE	63.63	0.9805	0.0365	0.9162	0.0864
		SMOTE-ENN		0.9736	0.0383	0.9203	0.0831
	If N4 $\leq$ 0.2509 then good behavior	EUSCHC	70.45	0.9564	0.0392	0.9349	0.0449
R1–	If F1 $\leq$ 0.366 then bad behavior	SMOTE	20.45	0.8637	–0.0803	0.6687	–0.1611
		SMOTE-ENN		0.8364	–0.0989	0.6618	–0.1754
	If F1 $\leq$ 0.3534 then bad behavior	EUSCHC	18.18	0.7914	–0.1258	0.7459	–0.1441
R2–	If N4 $\geq$ 0.2261 then bad behavior	SMOTE	36.36	0.8801	–0.0639	0.6788	–0.1510
		SMOTE-ENN		0.8684	–0.0669	0.6917	–0.1455
	If N4 $\geq$ 0.2754 then bad behavior	EUSCHC	29.55	0.8236	–0.0936	0.7831	–0.1069

In the remaining of this section, we will first show the methodology followed in this study in Sect. 5.1. Next the empirical study for both C4.5 and PART in imbalanced data sets with data complexity measures is shown in Sect. 5.2. Finally, in Sect. 5.3 the collective evaluation of the set of rules for both learning models is carried out.

5.1 Methodology

In order to characterize the results of C4.5 and PART when using undersampling and oversampling preprocessing in imbalanced data, we will follow the methodology proposed in (Luengo and Herrera 2010) for characterizing the performance of a learning method in standard classification, which is briefly described next.

We consider intervals of data complexity measures' values in which C4.5 and PART perform good or bad, calculated for every data set of Sect. 4.1.

- We understand for *good behavior* an average high test AUC in the interval (at least 0.8 approximately), as well as the absence of over-fitting (less than a 0.1 difference in Training and Test AUC approximately).
- By *bad behavior* we refer to the presence of over-fitting and/or average low test AUC in the interval.

Table 9 Data sets sorted by F1 covered by the R1+ and R1– rules

Data set	F1	C4.5, PART with SMOTE, SMOTE-ENN	C4.5 with EUSCHC	PART with EUSCHC
Vehicle2	0.1691	R1- bad behavior	R1- bad behavior	R1- bad behavior
Yeast1458vs7	0.1757			
Haberman	0.1850			
Vehicle3	0.1855			
Glass1	0.1897			
Yeast1	0.2422			
Glass016vs2	0.2692			
Shuttle0vs4	0.3534			
Yeast1289vs7	0.3660			
Vehicle1	0.3805			
Glass2	0.3952			
Page-blocks0	0.5087			
Abalone19	0.5295			
Pima	0.5760			
Abalone9vs18	0.6320			
Glass0	0.6492			
Yeast4	0.7412			
Glass5	1.0190			
Yeast05679vs4	1.0510			
Vehicle0	1.1240			
Yeast2vs8	1.1420			
Glass4	1.4690	R1+ good behavior	R1+ good behavior	R1+ good behavior
Page-blocks13vs2	1.5470			
Ecoli3	1.5790			
Yeast2vs4	1.5790			
Segment0	1.7980			
Ecoli2	1.8260			
Glass016vs5	1.8510			
Ecoli0137vs26	1.9670			
Yeast6	2.3020			
Glass6	2.3910			
Vowel0	2.4580			
Ecoli1	2.6500			
Yeast3	2.7510			
Ecoli4	3.2470			
Glass0123vs456	3.3240			
Wisconsin	3.5680			
New-thyroid2	3.5790			
New-thyroid1	3.5790			
Yeast5	4.1980			
Ecoli0vs1	9.7520			
Shuttle2vs4	12.1300			
Yeast1vs7	12.9700			
Iris0	16.8200			

Table 10 Data sets sorted by N4 covered by the R3+ and R2– rules

Data set	N4	C4.5, PART with SMOTE, SMOTE-ENN	C4.5 with EUSCHC	PART with EUSCHC
Iris0	0.0000	R3+ good behavior	R3+ good behavior	R3+ good behavior
Yeast1vs7	0.0016			
Ecoli0vs1	0.0136			
Shuttle2vs4	0.0155			
New-thyroid1	0.0209			
New-thyroid2	0.0233			
Segment0	0.0358			
Wisconsin	0.0432			
Ecoli4	0.0506			
Glass6	0.0537			
Page-blocks13vs2	0.0540			
Glass0123vs456	0.0561			
Ecoli2	0.0685			
Glass5	0.0724			
Glass016vs5	0.0788			
Yeast3	0.1122			
Yeast6	0.1157			
Yeast5	0.1216			
Ecoli1	0.1265			
Glass4	0.1285			
Yeast2vs4	0.1333			
Ecoli3	0.1652			
Ecoli0137vs26	0.1701			
Vehicle0	0.1734			
Vehicle1	0.1761			
Glass0	0.2009			
Vowel0	0.2034			
Page-blocks0	0.2069			
Yeast2vs8	0.2261	R2- bad behavior	R2- bad behavior	R2- bad behavior
Yeast4	0.2342			
Yeast05679vs4	0.2509			
Pima	0.2754			
Glass016vs2	0.2891			
Glass1	0.3084			
Shuttle0vs4	0.3137			
Yeast1	0.3201			
Vehicle2	0.3304			
Abalone9vs18	0.3324			
Glass2	0.3364			
Haberman	0.3431			
Yeast1289vs7	0.3627			
Vehicle3	0.3747			
Yeast1458vs7	0.3752			
Abalone19	0.4534			

The intervals are extracted by means of a particular graphic representation of the AUC results for C4.5 or PART considering the three preprocessing methods. The AUC results for the preprocessed data sets are arranged equidistantly in an ordered series, sorting them by one of the data complexity measures computed over the original data sets. Therefore, the *X* axis contains the data sets with a constant separation (and not related with the values of the considered data complexity), and the *Y* axis depicts the AUC obtained both in Training and Test for the particular data set. The reason to use this constant separation is to give each data set the same space in the graphic representation.

For those measures where we can find different *ad-hoc* intervals which present *good* or *bad behavior* of C4.5 or PART, we use a vertical line to delimit the interval of the region of interest.

Following the process described, not every data complexity measure can be used in order to select an interval of good or bad behavior of the methods. Figure 5 depicts an example in which no interval could be extracted for the F2 measure. Figure 6 shows an example in which both good and bad behavior intervals could be extracted, indicated by vertical lines for the F1 measure using the same preprocessing technique.

As mentioned in Sect. 3.2, only F1, N4 and L3 data complexity measures can be used to extract significative intervals with enough support following this methodology.

Table 11 Data sets sorted by L3 covered by the R2+ rule

Data set	L3	C4.5, PART with SMOTE, SMOTE-ENN	C4.5 with EUSCHC	PART with EUSCHC
Iris0	0.0000	R2+ good behavior	R2+ good behavior	R2+ good behavior
Yeast1vs7	0.0019			
Wisconsin	0.0066			
Page-blocks13vs2	0.0678			
Ecoli0vs1	0.1182			
Vehicle0	0.1219			
Vehicle1	0.2311			
New-thyroid1	0.2721			
New-thyroid2	0.2791			
Glass0123vs456	0.3294			
Page-blocks0	0.3332			
Vehicle3	0.3511			
Vehicle2	0.3682			
Haberman	0.4967			
Yeast1458vs7	0.5000			
Glass1	0.5000			
Yeast1	0.5000			
Glass016vs2	0.5000			
Shuttle0vs4	0.5000			
Yeast1289vs7	0.5000			
Glass2	0.5000			
Abalone19	0.5000			
Pima	0.5000			
Abalone9vs18	0.5000			
Glass0	0.5000			
Yeast4	0.5000			
Glass5	0.5000			
Yeast05679vs4	0.5000			
Yeast2vs8	0.5000			
Glass4	0.5000			
Ecoli3	0.5000			
Yeast2vs4	0.5000			
Segment0	0.5000			
Ecoli2	0.5000			
Glass016vs5	0.5000			
Ecoli0137vs26	0.5000			
Yeast6	0.5000			
Glass6	0.5000			
Vowel0	0.5000			
Ecoli1	0.5000			
Yeast3	0.5000			
Ecoli4	0.5000			
Yeast5	0.5000			

Table 12 Disjunction Rules from all simple rules for C4.5

Id.	Rule	Preprocess	Support (%)	Avg. AUC Train	Train diff.	Avg. AUC Test	Test diff.
PRD-S	If R1+ or R2+ R3+ then good behavior	SMOTE	63.63	0.9823	0.0277	0.9077	0.0861
PRD-S-ENN		SMOTE-ENN		0.9756	0.0318	0.9196	0.0834
PRD-EUS		EUSCHC	70.45	0.9692	0.0451	0.9460	0.0546
NRD-S	If R1– or R2– then bad behavior	SMOTE	36.36	0.9062	–0.0484	0.6712	–0.1504
NRD-S-ENN		SMOTE-ENN		0.8881	–0.0557	0.6903	–0.1459
NRD∧¬PRD-EUS		EUSCHC	29.55	0.8166	–0.1075	0.7613	–0.1301

Table 13 Disjunction Rules from all simple rules for PART

Id.	Rule	Preprocess	Support (%)	Avg. AUC Train	Train diff.	Avg. AUC Test	Test diff.
PRD-S	If R1+ or R2+ R3+ then good behavior	SMOTE	63.63	0.9805	0.0365	0.9162	0.0864
PRD-S-ENN		SMOTE-ENN		0.9736	0.0383	0.9203	0.0831
PRD-EUS		EUSCHC	72.73	0.9549	0.0377	0.9337	0.0437
NRD-S	If R1– or R2– then bad behavior	SMOTE	36.36	0.8801	–0.0639	0.6788	–0.1510
NRD-S-ENN		SMOTE-ENN		0.8684	–0.0669	0.6917	–0.1455
NRD∧¬PRD-EUS		EUSCHC	27.27	0.8167	–0.1005	0.7736	–0.1164

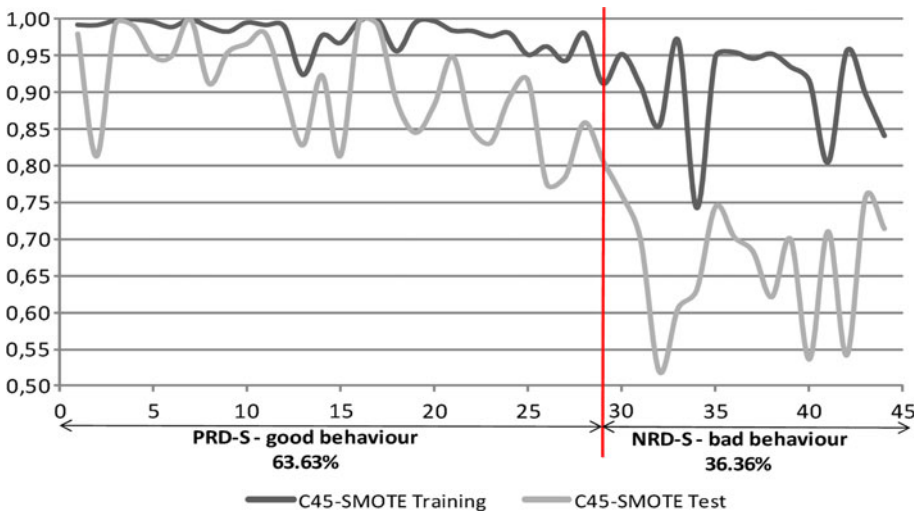

Fig. 7 Three blocks representation for C4.5 with SMOTE

Our objective is to analyze the data sets covered by a good or bad behavior interval considering the different preprocessing methods. These data sets will be characterized as those which the preprocessing technique used allows C4.5 and PART to obtain good or bad results.

5.2 Single intervals extraction

As we have previously indicated, only the F1, N4 and L3 measures offered significant intervals following the process described in Sect. 5.1. In Appendix 1, Figs. 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,

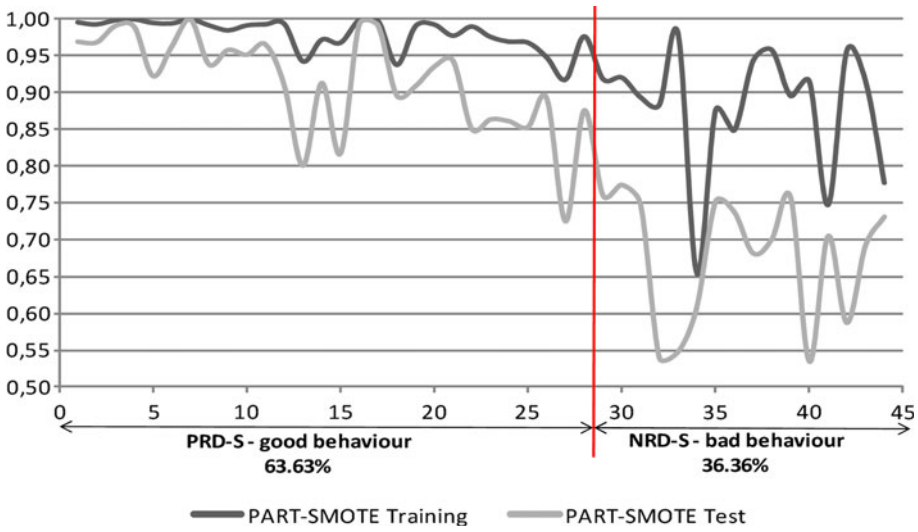

Fig. 8 Three blocks representation for PART with SMOTE

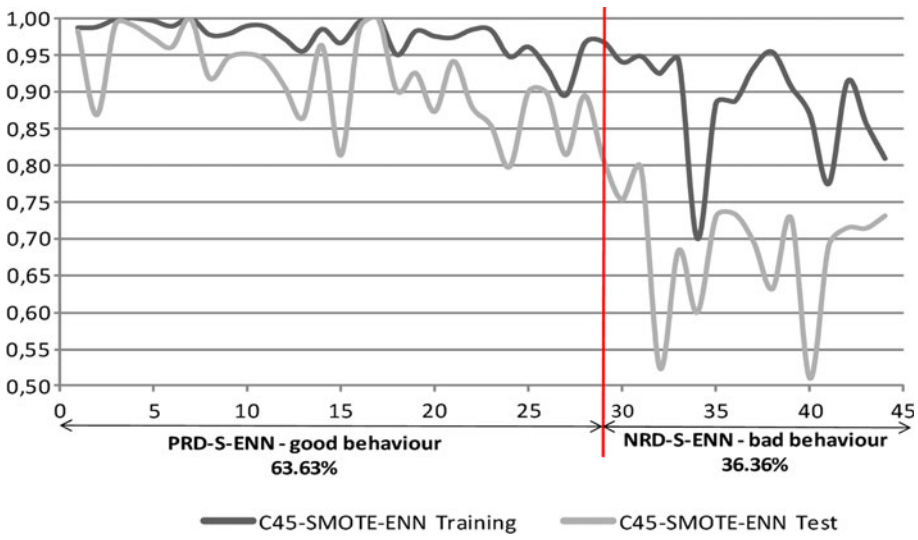

Fig. 9 Three blocks representation for C4.5 with SMOTE-ENN

26, 27, 28, 29, 30 depict the results for C4.5 and PART considering these three data complexity measures.

In Table 6 we have summarized the intervals found ad-hoc from the aforementioned figures for C4.5 and PART. The intervals are the same for C4.5 and PART. The intervals obtained for SMOTE and SMOTE-ENN are also always the same, therefore we will not find differences between the two versions of SMOTE considered, except in the average AUC, from now on.

All the extracted intervals can be translated into rules, using them as the antecedents of the rules. In Table 7 we have summarized the rules derived from the individual intervals for C4.5 and in Table 8 we show the equivalent rules for PART. Both tables are organized with the following columns:

- The first column corresponds to the identifier of the rule for further references.
- The "Rule" column presents the rule itself.
- The third column shows the type of preprocessing carried out.
- The fourth column "Support" presents the percentage of data sets which verifies the antecedent of the rule.
- The column "% Training" shows the average AUC in training of all the data sets covered by the rule.
- The column "Training diff." contains the difference between the training AUC of the rule and the global training AUC across all 44 data sets showed in Tables 4 and 5 for the preprocessing case of the row (SMOTE, SMOTE-ENN or EUSCHC).
- The column "% Test" shows the average AUC in test of all the data sets covered by the rule.

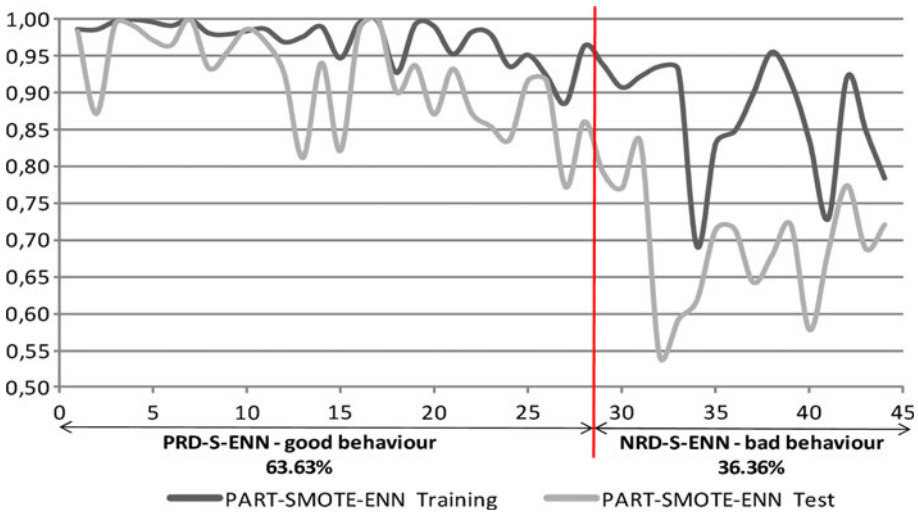

Fig. 10 Three blocks representation for PART with SMOTE-ENN

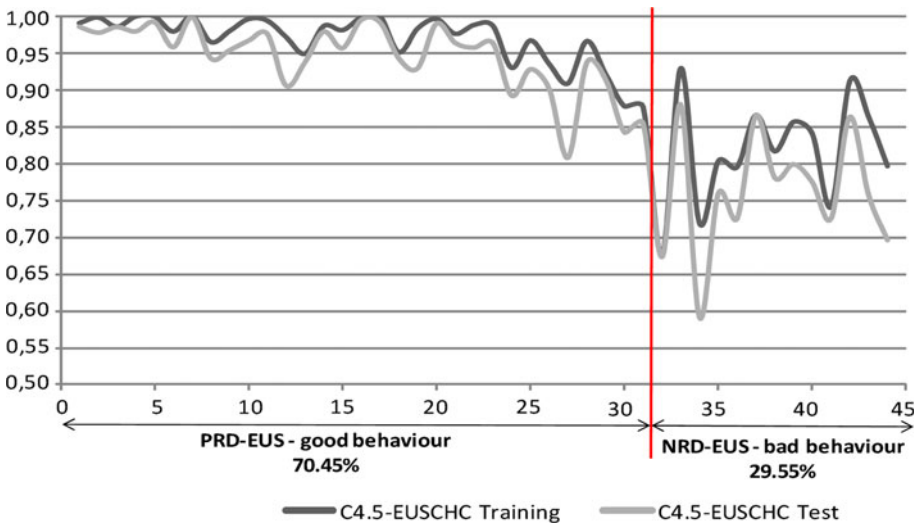

Fig. 11 Three blocks representation for C4.5 with EUSCHC

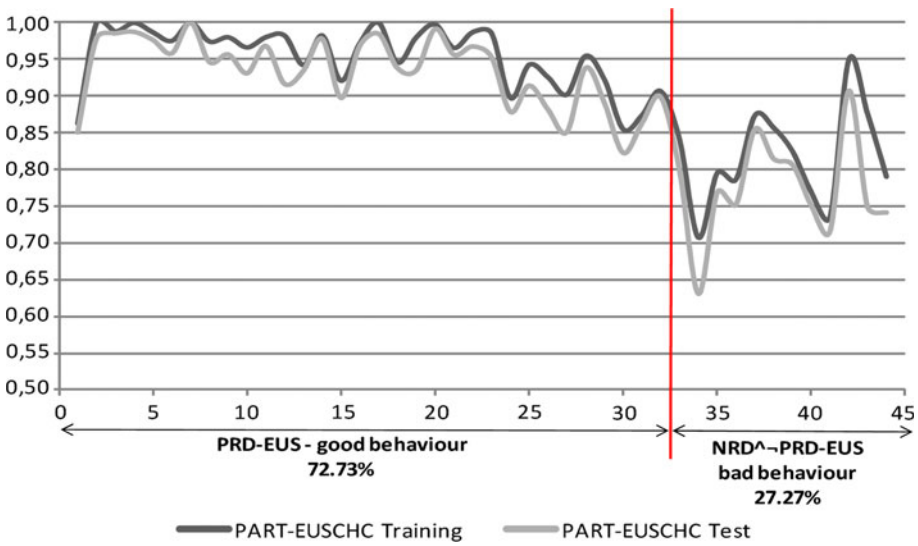

Fig. 12 Three blocks representation for PART with EUSCHC

Table 14 Data sets covered by the PRD and NRD rules

Data set	C4.5, PART with SMOTE, SMOTE-ENN	C4.5 with EUSCHC	PART with EUSCHC
Ecoli0vs1	PRD good behavior	PRD good behavior	PRD good behavior
Glass016vs5			
Segment0			
Iris0			
Vowel0			
Vehicle2			
shuttle0vs4			
Vehicle0			
Wisconsin			
New-Thyroid2			
New-Thyroid1			
Glass0123vs456			
Yeast6			
Yeast5			
Ecoli0137vs26			
Page-Blocks13vs4			
shuttle2vs4			
Yeast3			
Glass6			
Glass5			
Page-Blocks0			
Glass4			
Ecoli4			
Ecoli3			
Ecoli2			
Ecoli1			
Glass0			
Yeast2vs4			
Yeast2vs8	NRD bad behavior	NRD bad behavior	NRD∧¬PRD bad behavior
Yeast05679vs4			
Yeast4			
Abalone19			
Glass016vs2			
Haberman			
Vehicle3			
Vehicle1			
Yeast1289vs7			
Abalone9-18			
yeastB1vs7			
Yeast1458vs7			
Yeast2			
Glass2			
Glass1			
Pima			

- The column "Test diff." contains the difference between the test AUC of the rule and the global test AUC across all 44 data sets showed in Tables 4 and 5 for the preprocessing case of the row (SMOTE, SMOTE-ENN or EUSCHC).

The positive rules (denoted with a "+" symbol in their identifier) always show a positive difference with the global average, both in training and test AUC. The negative ones (with a "–" symbol in their identifier) verify the opposite case. The support of the rules shows us that we can characterize a wide range of data sets and obtain significant differences in AUC both inf with and without preprocessing cases.

In Tables 9, 10 and 11 the specific data sets arranged by the F1, N4, and L3 measures, respectively, are depicted. The data sets covered by each rule is indicated by means of the adjacent columns, considering the three different support cases obtained from the rules: SMOTE and SMOTE-ENN for both C4.5 and PART; EUSCHC for C4.5, and EUSCHC for PART.

If we compare the equivalent rules and the covered data sets in the different cases of SMOTE, SMOTE-ENN and EUSCHC preprocessing we can observe the following:

- With the use of EUSCHC the support of R1+ and R3+ rules with this preprocessing method is wider than the SMOTE–based approaches.
- The support of the R1– and R2– is also smaller for EUSCHC, as less data sets can be identified as bad for C4.5 or PART. These differences with respect to SMOTE and SMOTE-ENN has been properly characterized by the rules.
- The R2+ rules is invariant with respect to the preprocessing. It represents the good data sets for

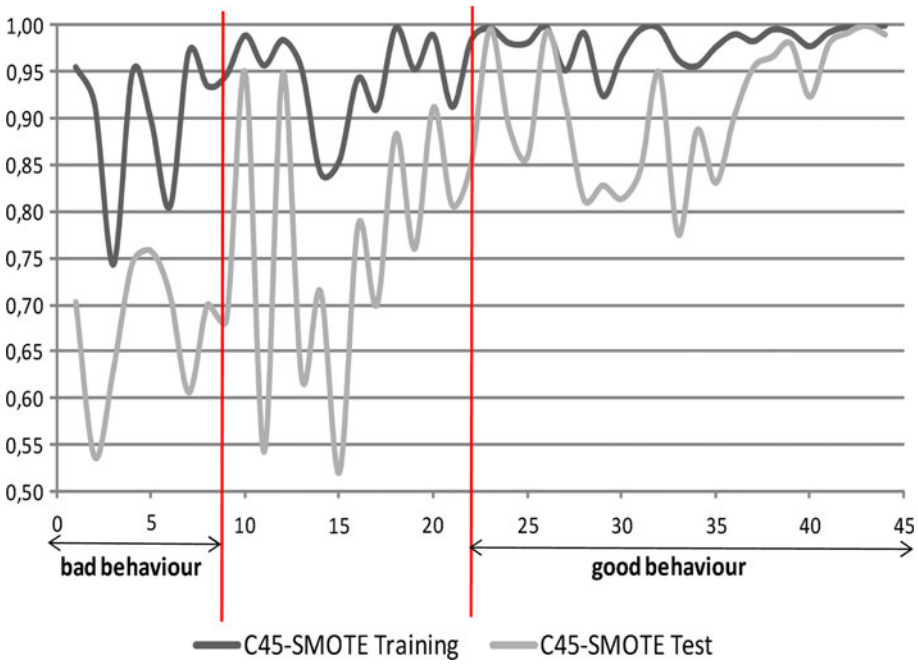

Fig. 13 C4.5 with SMOTE AUC in Training/Test sorted by F1

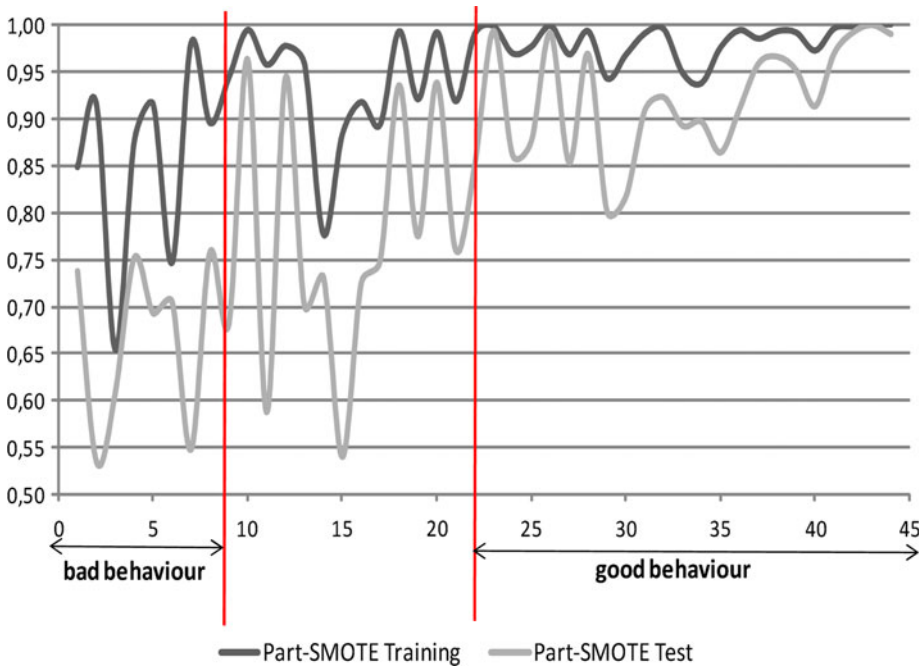

Fig. 14 PART with SMOTE AUC in Training/Test sorted by F1

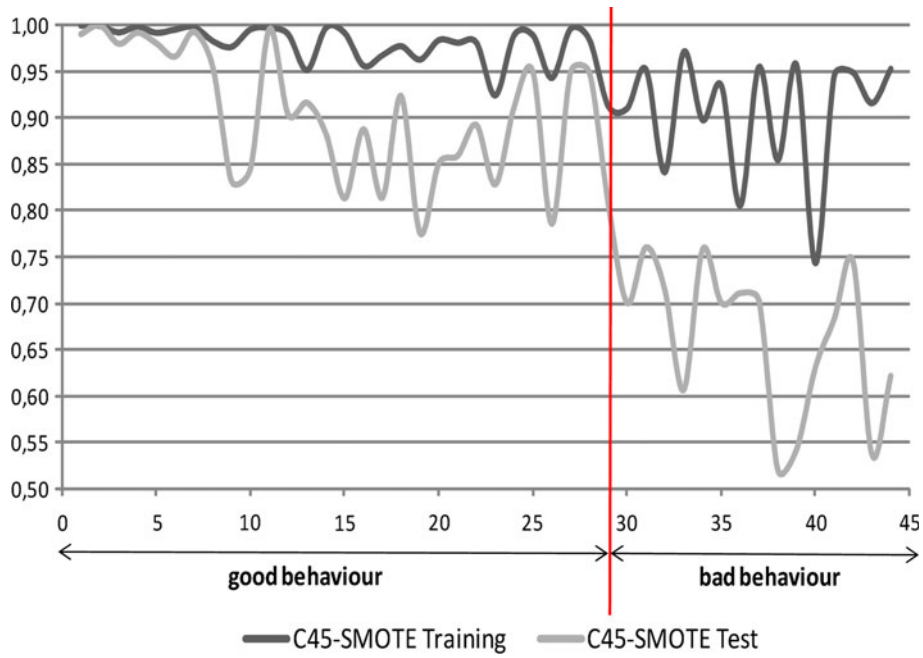

Fig. 15 C4.5 with SMOTE AUC in Training/Test sorted by N4

C4.5 and PART considering any preprocessing technique.

- SMOTE and SMOTE-ENN behave equal for C4.5 and PART. The differences with respect to the global training and test AUC are similar, and the support is equal in every case.
- EUSCHC behaves equally for C4.5 and PART when considering the N4 and L3 data complexity measures. However, for the F1 measure, EUSCHC is slightly better for PART, as the support of R1+ for this learning method is a 3% higher.

5.3 Combination of the single rules

The objective of this section is to analyze the effect of combining the rules of good and behavior independently. By means of merging the individual rules we can arrive at a

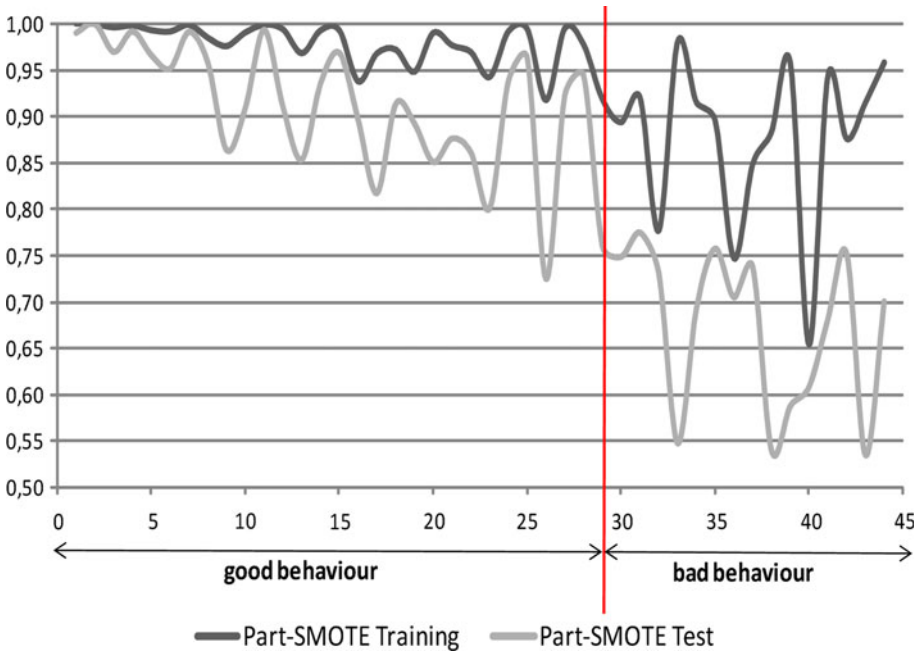

Fig. 16 PART with SMOTE AUC in Training/Test sorted by N4

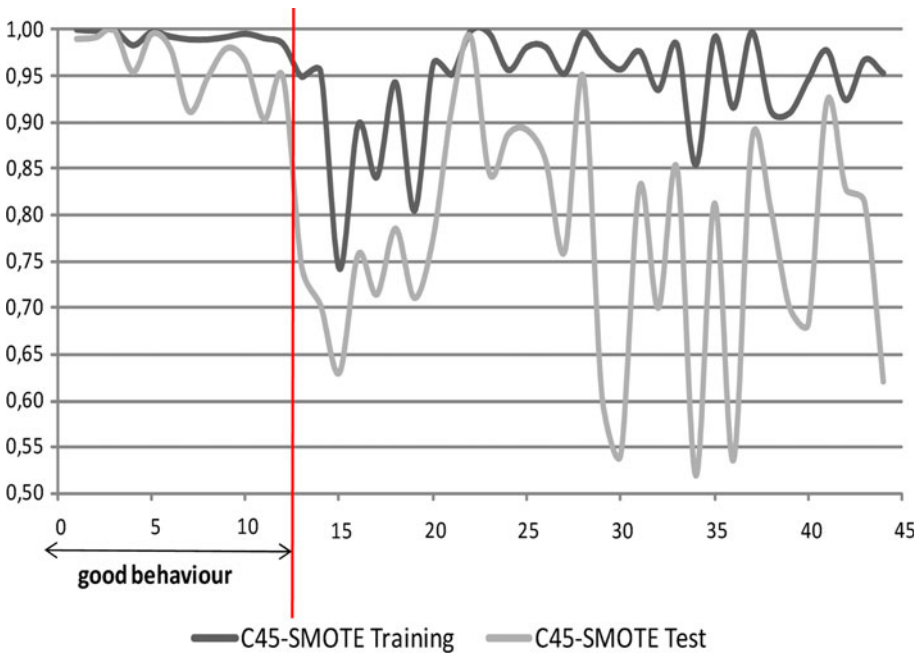

Fig. 17 C4.5 with SMOTE AUC in Training/Test sorted by L3

more general description, with a wider support, of the preprocessing methods' effect.

First, we have considered the disjunctive combination of all the positive rules to obtain a single rule (Positive Rule Disjunction -PRD-), that is, we use the *or* operator to combine the individual positive rules. The same procedure is done with all the negative ones so we obtain another rule (Negative Rule Disjunction -NRD-). The new disjunctive rules will be activated if any of the component rules' antecedents are verified.

In the case of PART with EUSCHC preprocessing, overlapping between the data sets covered by PRD and NRD appears. Following the same methodology presented by Luengo and Herrera (2010) the PRD rule is kept as representative of the good behavior, and the set difference between the NRD and PRD rules is

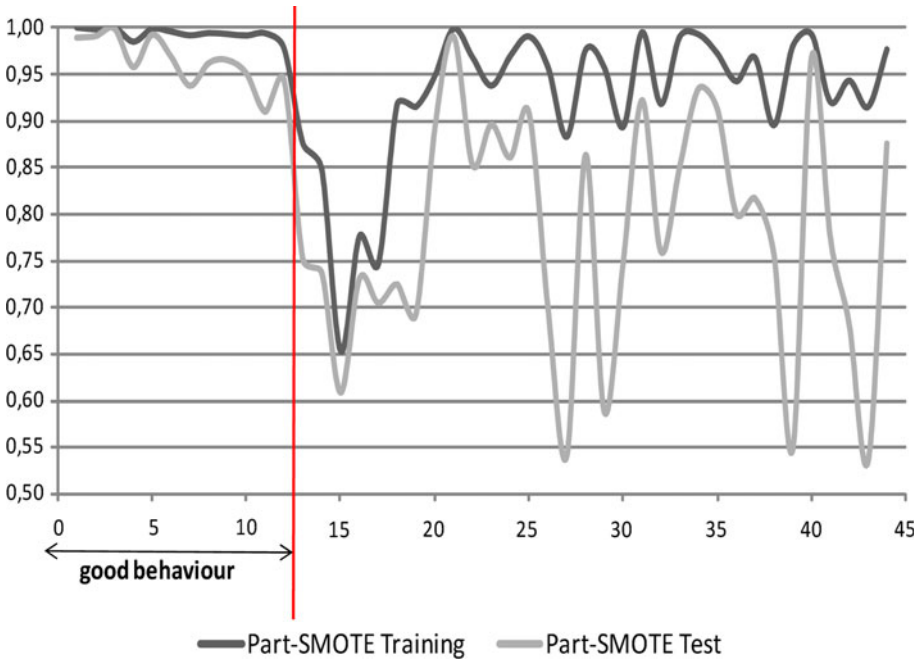

Fig. 18 PART with SMOTE AUC in Training/Test sorted by L3

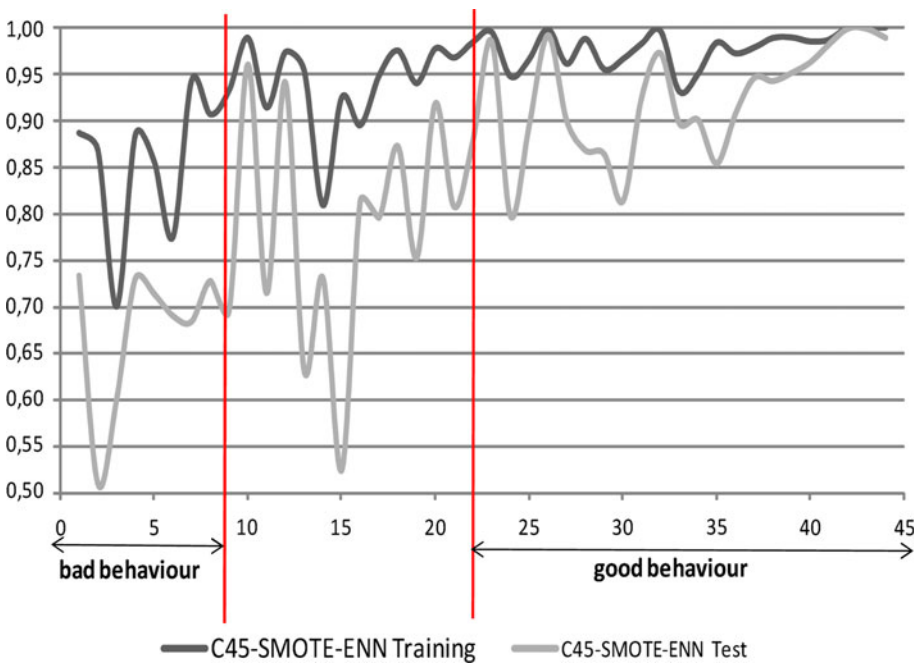

Fig. 19 C4.5 with SMOTE-ENN AUC in Training/Test sorted by F1

considered as representative of the bad behavior. This difference will remove the data sets for which C4.5 and PART presents good behavior from the NRD rule, naming this new rule as NRD$\wedge\neg$PRD.

In Table 12 we summarize the new rules obtained for C4.5. In Table 13 the equivalent rules for PART are depicted. In these two tables, we have added the following suffixes to the rule identifier in order to distinguish them:

- In the case of SMOTE preprocessing, we add the "-S" suffix.
- In the case of SMOTE-ENN preprocessing, we add the "-S-ENN" suffix.

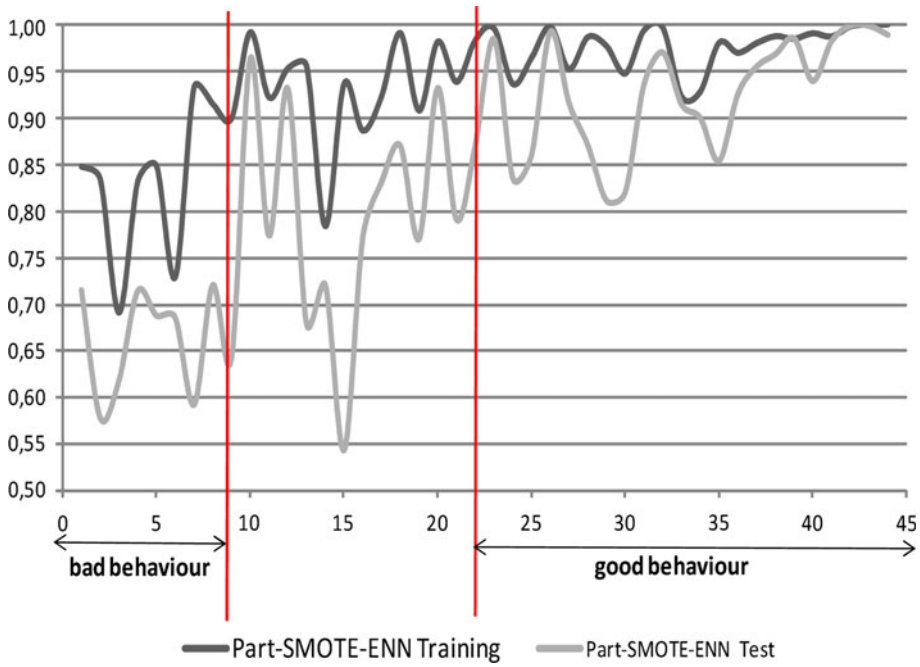

Fig. 20 PART with SMOTE-ENN AUC in Training/Test sorted by F1

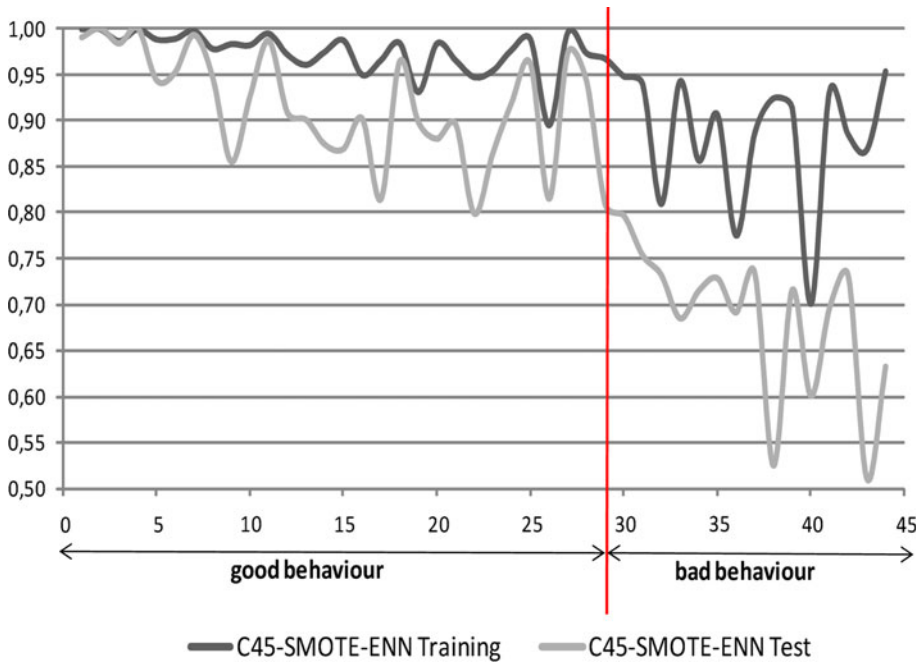

Fig. 21 C4.5 with SMOTE-ENN AUC in Training/Test sorted by N4

- In the case of EUSCHC preprocessing, we add the "-EUS" suffix.

From the collective rules for both learning methods, it is observed that the support has been increased from the single rules for PRD, while NRD (and NRD∧¬PRD for EUSCHC) obtains similar support. On the other hand, the training and test AUC differences are similar to the single rules from Tables 7 and 8 in both with and without preprocessing situations.

With the PRD and NRD (NRD∧¬PRD-EUS for PART) there is no uncovered data sets by the rules for C4.5 and

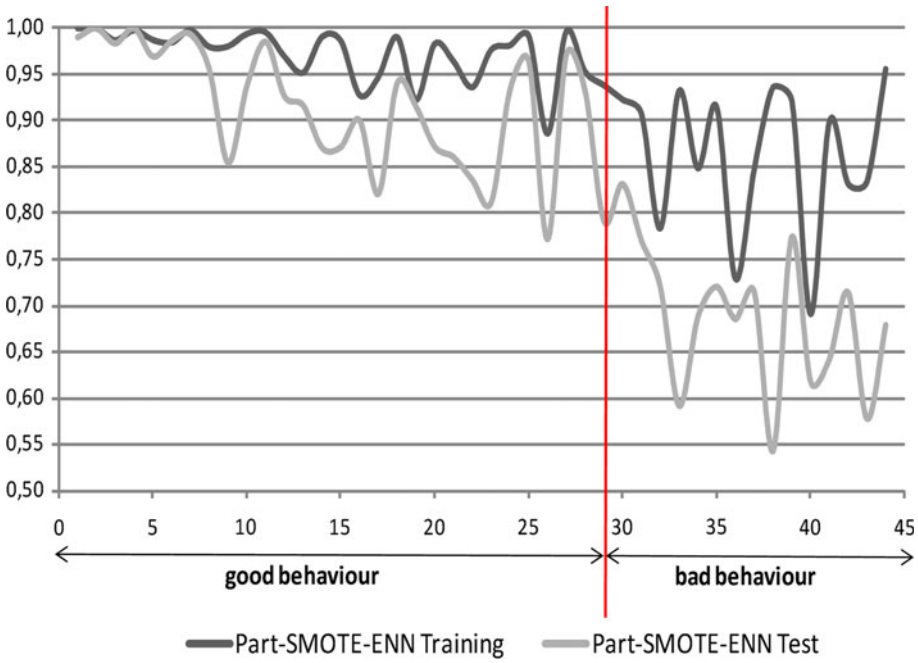

Fig. 22 PART with SMOTE-ENN AUC in Training/Test sorted by N4

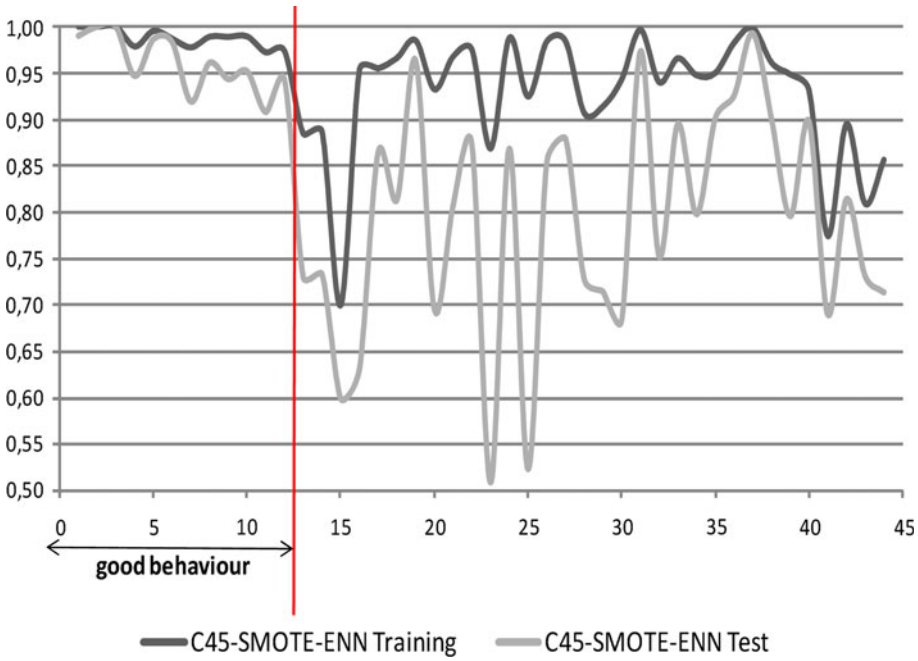

Fig. 23 C4.5 with SMOTE-ENN AUC in Training/Test sorted by L3

PART in combination with SMOTE, SMOTE-ENN, and EUSCHC preprocessing methods. Therefore, we can consider a two block representation of the data sets.

- The first block (the left-side one) will represent the data sets covered by the correspondent PRD rule. They are the data sets recognized as being those in which C4.5 and PART have good AUC when preprocessing.
- The second (the right-side one) will plot the data sets for the correspondent NRD (NRD$\wedge\neg$PRD-EUS for PART) rule, which are bad data sets for C4.5 and PART after preprocessing.

In Figs. 7, 9 and 11 we have depicted the two block representation for C4.5 considering the three cases of preprocessing. In Figs. 8, 10 and 12 we have depicted the

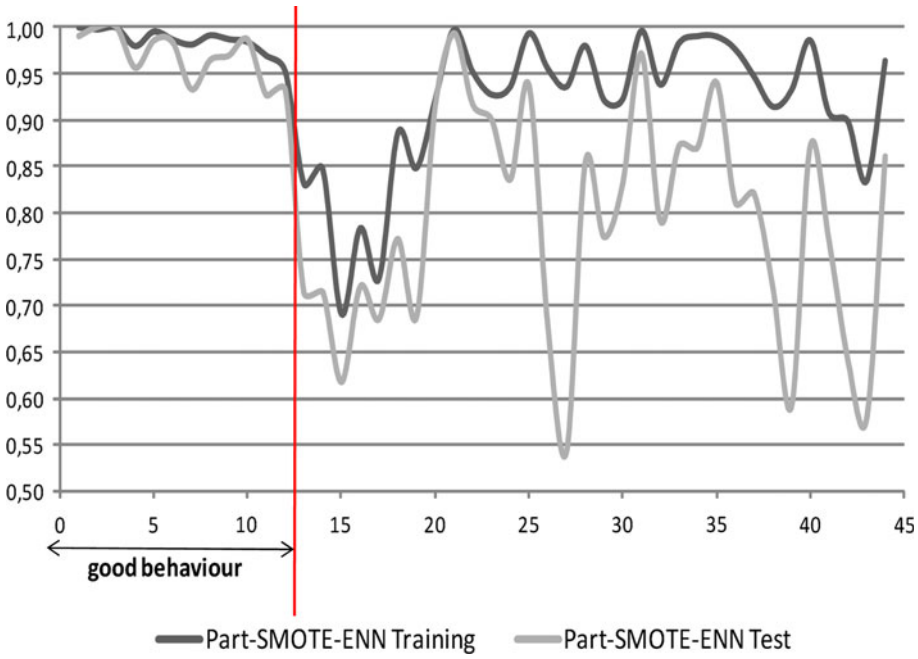

Fig. 24 PART with SMOTE-ENN AUC in Training/Test sorted by L3

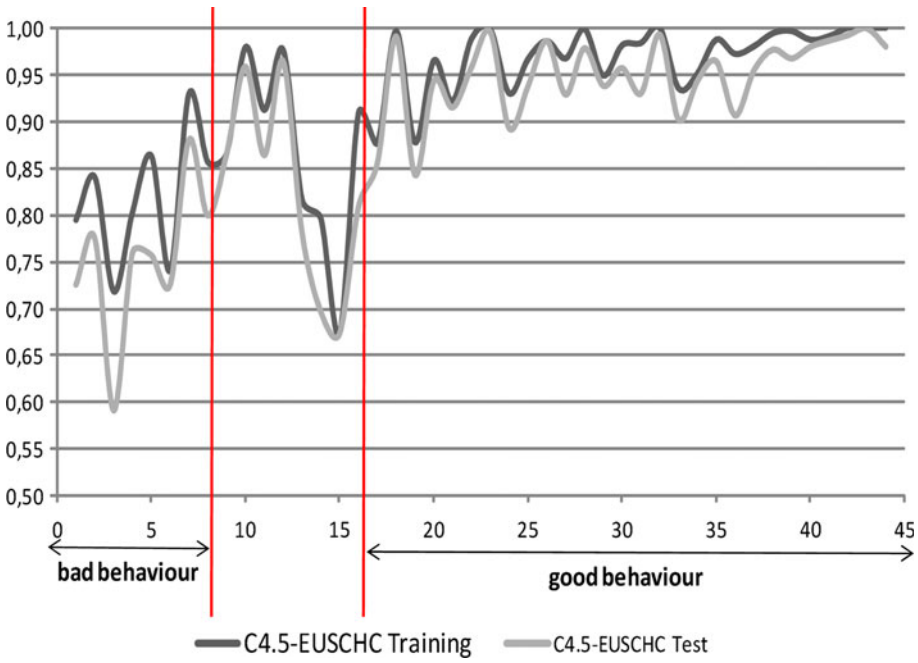

Fig. 25 C4.5 with EUSCHC AUC in Training/Test sorted by F1

same representations for PART. The data sets have the same order in the X axis in all the figures to facilitate the comparisons.

Table 14 represents the data sets following the order of the latter figures indicating those data sets which are covered by the PRD and NRD (NRD$\wedge\neg$PRD-EUS for PART) rules as indicate by the vertical lines in the two blocks representation.

We can observe that the 100% of the analyzed data sets are covered by the two considered rules for each preprocessing method. Since SMOTE and SMOTE-ENN obtained the same intervals in the previous subsection, the

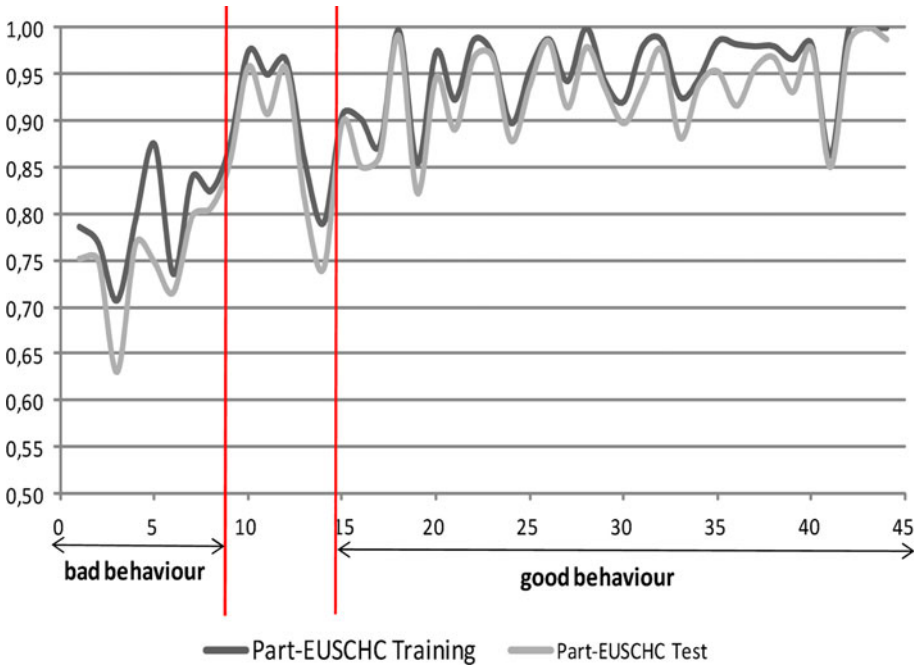

Fig. 26 PART with EUSCHC AUC in Training/Test sorted by F1

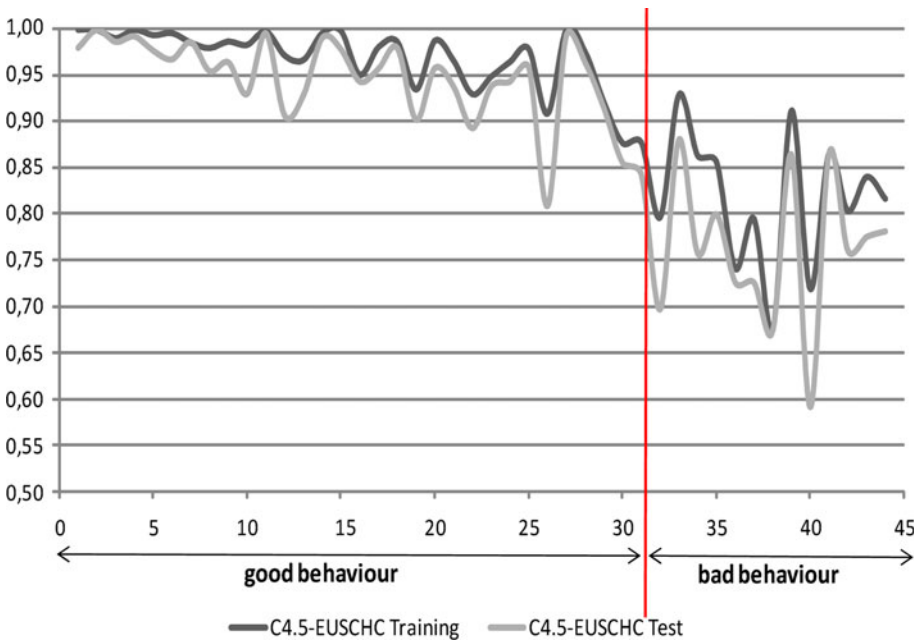

Fig. 27 C4.5 with EUSCHC AUC in Training/Test sorted by N4

support of the PRD and NRD rules are the same, and they cover the same data sets.

The EUSCHC approach obtains a wider support for the PRD rule with respect to the SMOTE and SMOTE-ENN approaches. This is due to the wider support of the individual intervals which conform the PRD rule. This difference indicates that the undersampling approach is more beneficial for C4.5 and PART, since more data sets are characterized as good for these two learning methods.

From these results we can point out that the data complexity measures are useful to evaluate the behavior of the undersampling and oversampling approaches. Differences in their results have been characterized, finding that

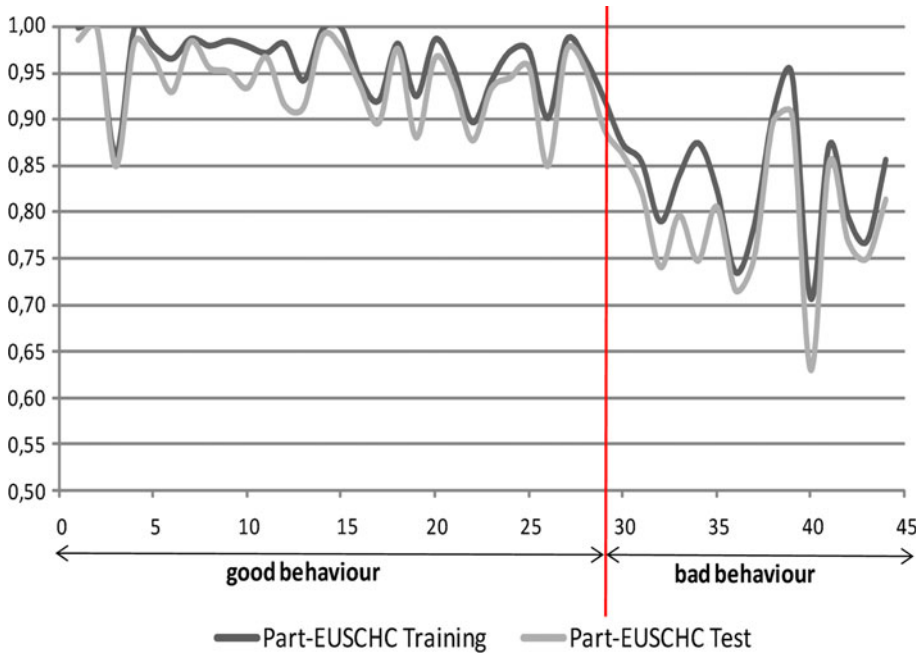

Fig. 28 PART with EUSCHC AUC in Training/Test sorted by N4

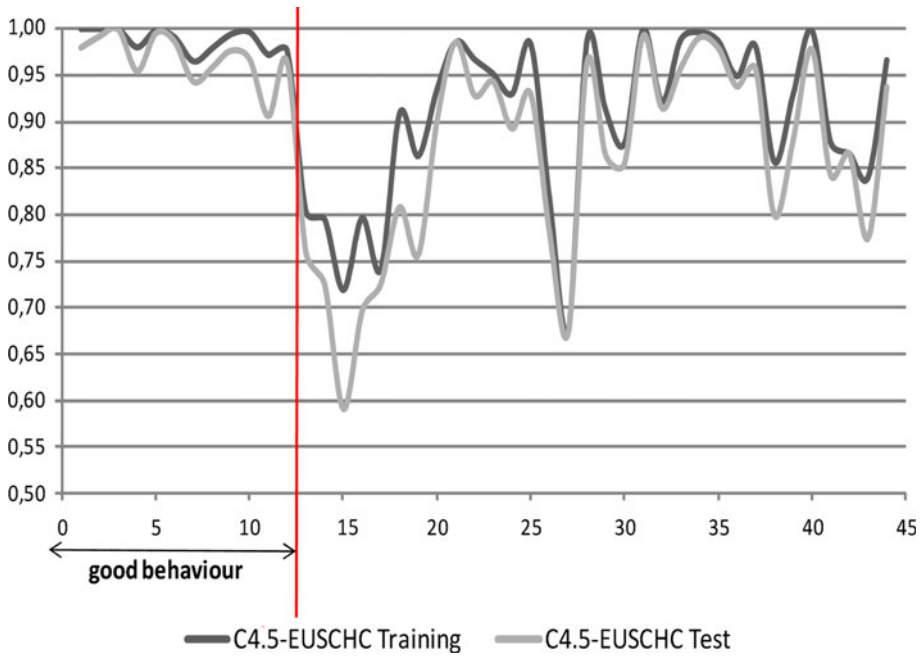

Fig. 29 C4.5 with EUSCHC AUC in Training/Test sorted by L3

EUSCHC is more robust than SMOTE-based approaches due to its wider region of good behavior.

There is a bunch of data sets for which both undersampling and oversampling techniques do not work well (indicated in Table 14), as the data set covered by all NRD and the NRD$\wedge\neg$PRD-EUS rule. These data sets are therefore opened to improvements by means of these or other techniques, but already identified by the rules.

6 Concluding remarks

In this work we have analyzed the preprocessing effect in the framework of imbalanced data sets by means of data complexity measures. We have considered two oversampling methods: SMOTE and SMOTE-ENN, and an evolutionary undersampling approach: EUSCHC.

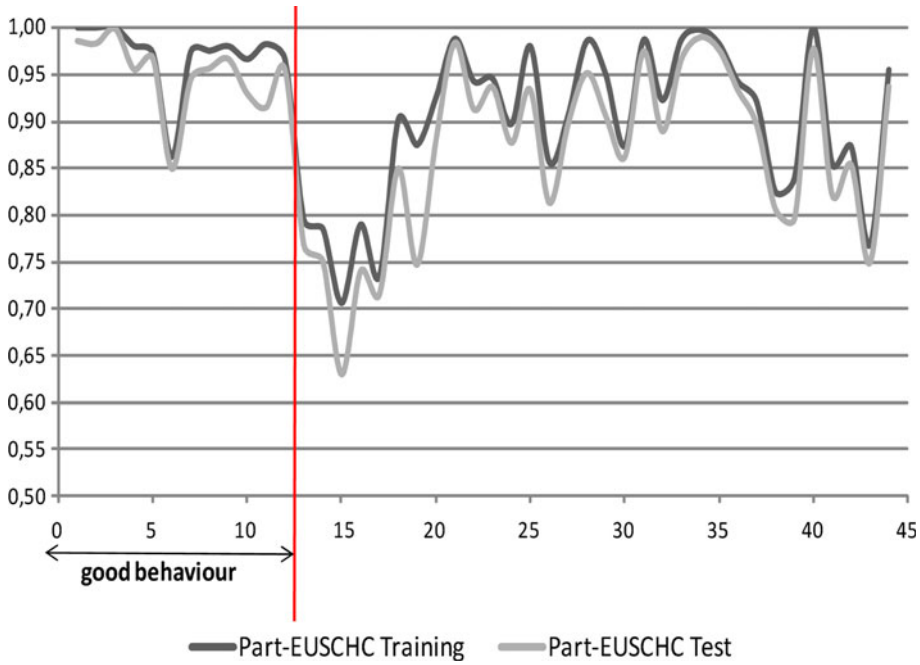

Fig. 30 PART with EUSCHC AUC in Training/Test sorted by L3

We have observed that the IR considered as a measure of data complexity is not enough to predict when C4.5 and PART perform good or bad. As an alternative approach, we have computed the data complexity measures over the imbalanced data sets in order to obtain intervals of such metrics in which C4.5 and PART performance is significantly good and bad when using the three preprocessing methods. From these intervals we have built descriptive rules, which have a wide support and a significative difference with respect to the global methods' performance.

We have obtained two final rules from the initial ones, which are simple and precise to describe both good and bad performance of C4.5 and PART. These two rules are capable of identifying all good and bad data sets for SMOTE, SMOTE-ENN, and EUSCHC. An interesting consequence of the characterization obtained by the rules is that the evolutionary undersampling approach is capable of preprocessing successfully more data sets for C4.5 and PART.

As a final note, it is interesting to indicate that the Fisher's Discriminant Ratio (F1) was also found interesting by the studies of Mollineda et al. (2005); Kim and Oommen (2009) considering prototype selection, and it is informative for our analysis in the imbalance framework as well.

Acknowledgments This work has been supported by the Spanish Ministry of Education and Science under Project TIN2008-06681-C06-(01 and 02). J. Luengo holds a FPU scholarship from Spanish Ministry of Education.

Appendix 1: Figures with the intervals of PART and C4.5

In this appendix, the figures sorted by the F1, N4 and L3 data complexity measures are depicted. We have used a two-column representation for the figures, so in each row we present the results for C4.5 and PART for the same case of type of preprocessing and data complexity measure used.

- Figures from 13, 14, 15, 16, 17, 18 represents the figures for the case of SMOTE preprocessing.
- Figures from 19, 20, 21, 22, 23, 24 represents the figures for the case of SMOTE-ENN preprocessing.
- Figures from 25, 26, 27, 28, 29, 30 represents the figures for the case of EUSCHC preprocessing.

Appendix 2: Tables of results

In this appendix we present the average AUC results for C4.5 and PART in Tables 15 and 16 respectively.

Table 15 Average AUC results for C4.5

Data sets	SMOTE Training	SMOTE Test	SMOTE-ENN Training	SMOTE-ENN Test	EUSCHC Training	EUSCHC Test
Ecoli0vs1	0.9927	0.9796	0.9870	0.9832	0.9909	0.9864
Haberman	0.7426	0.6309	0.6999	0.6003	0.7190	0.5914
Iris0	1.0000	0.9900	1.0000	0.9900	1.0000	0.9800
Pima	0.8411	0.7145	0.8089	0.7312	0.7966	0.6966
Vehicle2	0.9895	0.9492	0.9890	0.9611	0.9793	0.9586
Wisconsin	0.9832	0.9545	0.9784	0.9467	0.9802	0.9546
Yeast2	0.8049	0.7109	0.7744	0.6904	0.7411	0.7257
Glass0	0.9433	0.7856	0.8950	0.8143	0.9089	0.8085
Glass1	0.8978	0.7577	0.8563	0.7141	0.8633	0.7571
Vehicle1	0.9551	0.7030	0.8866	0.7335	0.7952	0.7258
Vehicle3	0.9493	0.7444	0.8844	0.7304	0.8035	0.7601
Ecoli1	0.9631	0.7755	0.9313	0.8979	0.9353	0.9019
Glass0123vs456	0.9908	0.9032	0.9721	0.9078	0.9720	0.9063
New-Thyroid1	0.9922	0.9802	0.9888	0.9433	0.9942	0.9767
New-Thyroid2	0.9957	0.9659	0.9895	0.9520	0.9965	0.9674
Page-Blocks0	0.9846	0.9485	0.9737	0.9421	0.9763	0.9644
Segment0	0.9985	0.9927	0.9985	0.9927	0.9859	0.9861
Vehicle0	0.9897	0.9118	0.9775	0.9192	0.9651	0.9433
Ecoli2	0.9517	0.9162	0.9610	0.9002	0.9673	0.9287
Yeast3	0.9565	0.8876	0.9500	0.9016	0.9513	0.9434
Ecoli3	0.9815	0.8921	0.9474	0.7980	0.9301	0.8928
Glass6	0.9959	0.8450	0.9825	0.9257	0.9836	0.9299
Abalone9-18	0.9531	0.6215	0.9539	0.6322	0.8167	0.7812
Abalone19	0.8544	0.5202	0.9245	0.5246	0.6751	0.6736
Ecoli4	0.9769	0.8310	0.9839	0.8544	0.9873	0.9643
Glass2	0.9571	0.5424	0.9139	0.7148	0.9124	0.8640
Yeast4	0.9101	0.7004	0.9484	0.7960	0.8770	0.8551
Vowel0	0.9967	0.9494	0.9965	0.9733	1.0000	0.9929
Yeast2vs8	0.9125	0.8066	0.9677	0.8078	0.9217	0.9149
Glass4	0.9844	0.8508	0.9844	0.8794	0.9883	0.9580
Glass5	0.9976	0.8829	0.9753	0.8732	0.9965	0.9907
Yeast5	0.9777	0.9233	0.9851	0.9635	0.9870	0.9798
Yeast6	0.9242	0.8280	0.9549	0.8647	0.9491	0.9380
Ecoli0137vs26	0.9678	0.8136	0.9660	0.8136	0.9813	0.9572
Shuttle0vs4	0.9999	0.9997	0.9999	0.9997	1.0000	0.9995
YeastB1vs7	0.9351	0.7003	0.9066	0.7278	0.8567	0.7996
Shuttle2vs4	0.9990	0.9917	1.0000	1.0000	1.0000	0.9923
Glass016vs2	0.9716	0.6062	0.9430	0.6840	0.9296	0.8806
Glass016vs5	0.9921	0.8129	0.9879	0.8686	0.9986	0.9784
Page-Blocks13vs4	0.9975	0.9955	0.9952	0.9865	0.9984	0.9957
Yeast05679vs4	0.9526	0.7602	0.9401	0.7527	0.8778	0.8429
Yeast1289vs7	0.9465	0.6832	0.9323	0.6955	0.8659	0.8659
Yeast1458vs7	0.9158	0.5367	0.8685	0.5102	0.8405	0.7750
Yeast2vs4	0.9814	0.8588	0.9659	0.8953	0.9664	0.9377
Global	0.9546	0.8217	0.9438	0.8362	0.9241	0.8914

Table 16 Average AUC results for PART

Data sets	SMOTE Training	SMOTE Test	SMOTE-ENN Training	SMOTE-ENN Test	EUSCHC Training	EUSCHC Test
Ecoli0vs1	0.9958	0.9694	0.9870	0.9832	0.8625	0.8500
Haberman	0.6540	0.6086	0.6909	0.6183	0.7067	0.6305
Iris0	1.0000	0.9900	1.0000	0.9900	1.0000	0.9867
Pima	0.7769	0.7312	0.7836	0.7209	0.7900	0.7409
Vehicle2	0.9942	0.9628	0.9917	0.9642	0.9752	0.9574
Wisconsin	0.9848	0.9584	0.9800	0.9559	0.9802	0.9561
Yeast2	0.7468	0.7049	0.7288	0.6858	0.7350	0.7156
Glass0	0.9176	0.7250	0.8861	0.7720	0.9019	0.8503
Glass1	0.9151	0.6927	0.8485	0.6880	0.8750	0.7477
Vehicle1	0.8484	0.7377	0.8475	0.7153	0.7861	0.7518
Vehicle3	0.8757	0.7519	0.8314	0.7144	0.7949	0.7683
Ecoli1	0.9480	0.8923	0.9226	0.9151	0.9256	0.8810
Glass0123vs456	0.9939	0.9104	0.9695	0.9262	0.9825	0.9157
New-Thyroid1	0.9930	0.9659	0.9874	0.9690	0.9802	0.9674
New-Thyroid2	0.9915	0.9516	0.9845	0.9861	0.9663	0.9302
Page-Blocks0	0.9774	0.9439	0.9529	0.9322	0.9657	0.9556
Segment0	0.9987	0.9911	0.9978	0.9932	0.9880	0.9848
Vehicle0	0.9916	0.9382	0.9815	0.9328	0.9743	0.9456
Ecoli2	0.9681	0.8533	0.9521	0.9164	0.9427	0.9137
Yeast3	0.9377	0.8966	0.9277	0.9005	0.9456	0.9373
Ecoli3	0.9693	0.8611	0.9361	0.8359	0.8974	0.8779
Glass6	0.9905	0.9090	0.9939	0.9369	0.9802	0.9344
Abalone9-18	0.9581	0.7006	0.9559	0.6794	0.8567	0.8139
Abalone19	0.8831	0.5401	0.9362	0.5434	0.9066	0.8979
Ecoli4	0.9757	0.8639	0.9804	0.8544	0.9859	0.9524
Glass2	0.9571	0.5878	0.9218	0.7742	0.9497	0.9066
Yeast4	0.8936	0.7486	0.9228	0.8316	0.8738	0.8625
Vowel0	0.9950	0.9228	0.9967	0.9711	0.9868	0.9757
Yeast2vs8	0.9182	0.7599	0.9384	0.7915	0.9227	0.8901
Glass4	0.9901	0.8508	0.9832	0.8718	0.9872	0.9670
Glass5	0.9927	0.9354	0.9909	0.8707	0.9977	0.9907
Yeast5	0.9721	0.9132	0.9905	0.9403	0.9826	0.9771
Yeast6	0.9424	0.8008	0.9767	0.8115	0.9422	0.9340
Ecoli0137vs26	0.9678	0.8172	0.9474	0.8209	0.9208	0.8969
Shuttle0vs4	0.9999	0.9997	0.9999	0.9997	1.0000	0.9997
YeastB1vs7	0.8954	0.7576	0.9147	0.7207	0.8246	0.8061
Shuttle2vs4	0.9980	0.9917	0.9980	1.0000	1.0000	0.9840
Glass016vs2	0.9800	0.5479	0.9323	0.5921	0.8397	0.7969
Glass016vs5	0.9929	0.9686	0.9864	0.8714	1.0000	0.9784
Page-Blocks13vs4	0.9986	0.9932	0.9958	0.9854	0.9725	0.9681
Yeast05679vs4	0.9204	0.7748	0.9076	0.7704	0.8546	0.8221
Yeast1289vs7	0.9433	0.6815	0.8992	0.6427	0.8735	0.8543
Yeast1458vs7	0.9151	0.5351	0.8343	0.5783	0.7688	0.7503
Yeast2vs4	0.9765	0.8762	0.9642	0.8607	0.9548	0.9377
Global	0.9440	0.8298	0.9353	0.8372	0.9172	0.8900

References

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) KEEL: A software tool to assess evolutionary algorithms to data mining problems. Soft Comput 13(3):307–318

Asuncion A, Newman D (2007) UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html

Barandela R, Sánchez JS, García V, Rangel E (2003) Strategies for learning in class imbalance problems. Pattern Recognit 36(3):849–851

Basu M, Ho TK (2006) Data complexity in pattern recognition (advanced information and knowledge processing). Springer-Verlag New York, Inc., Secaucus

Batista GEAPA, Prati RC, Monard MC (2004) A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explor 6(1):20–29

Baumgartner R, Somorjai RL (2006) Data complexity assessment in undersampled classification of high-dimensional biomedical data. Pattern Recognit Lett 12:1383–1389

Bernadó-Mansilla E, Ho TK (2005) Domain of competence of XCS classifier system in complexity measurement space. IEEE Trans Evol Comput 9(1):82–104

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit 30(7):1145–1159

Brazdil P, Giraud-Carrier C, Soares C, Vilalta R (2009) Metalearning: applications to data mining. Cognitive Technologies, Springer. http://10.255.0.115/pub/2009/BGSV09

Celebi M, Kingravi H, Uddin B, Iyatomi H, Aslandogan Y, Stoecker W, Moss R (2007) A methodological approach to the classification of dermoscopy images. Comput Med Imaging Graphics 31(6):362–373

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chawla NV, Japkowicz N, Kolcz A (2004) Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor 6(1):1–6

Diamantini C, Potena D (2009) Bayes vector quantizer for class-imbalance problem. IEEE Trans Knowl Data Eng 21(5):638–651

Domingos P (1999) Metacost: a general method for making classifiers cost sensitive. In: Advances in neural networks, Int J Pattern Recognit Artif Intell, pp 155–164

Dong M, Kothari R (2003) Feature subset selection using a new definition of classificabilty. Pattern Recognit Lett 24:1215–1225

Drown DJ, Khoshgoftaar TM, Seliya N (2009) Evolutionary sampling and software quality modeling of high-assurance systems. IEEE Trans Syst Man Cybern A 39(5):1097–1107

Eshelman LJ (1991) Foundations of genetic algorithms, chap The CHC adaptive search algorithm: how to safe search when engaging in nontraditional genetic recombination, pp 265–283

Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36

Fernández A, García S, del Jesus MJ, Herrera F (2008) A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets Syst 159(18):2378–2398

Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: ICML '98: Proceedings of the fifteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, pp 144–151

García S, Herrera F (2009a) Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol Comput 17(3):275–306

García S, Fernández A, Herrera F (2009b) Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. Appl Soft Comput 9(4):1304–1314

García S, Cano JR, Bernadó-Mansilla E, Herrera F (2009c) Diagnose of effective evolutionary prototype selection using an overlapping measure. Int J Pattern Recognit Artif Intell 23(8):2378–2398

García V, Mollineda R, Sánchez JS (2008) On the k–NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Appl 11(3–4):269–280

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284

Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300

Hoekstra A, Duin RP (1996) On the nonlinearity of pattern classifiers. In: ICPR '96: Proceedings of the international conference on pattern recognition (ICPR '96) Volume IV-Volume 7472, IEEE Computer Society, Washington, DC, pp 271–275

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

Kalousis A (2002) Algorithm selection via meta-learning. PhD thesis, Université de Geneve

Kilic K, Uncu O, Türksen IB (2007) Comparison of different strategies of utilizing fuzzy clustering in structure identification. Inform Sci 177(23):5153–5162

Kim SW, Oommen BJ (2009) On using prototype reduction schemes to enhance the computation of volume-based inter-class overlap measures. Pattern Recognit 42(11):2695–2704

Li Y, Member S, Dong M, Kothari R, Member S (2005) Classifiability-based omnivariate decision trees. IEEE Trans Neural Netw 16(6):1547–1560

Lu WZ, Wang D (2008) Ground-level ozone prediction by support vector machine approach with a cost-sensitive classification scheme. Sci Total Environ 395(2–3):109–116

Luengo J, Herrera F (2010) Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method. Fuzzy Sets Syst 161(1):3–19

Mazurowski M, Habas P, Zurada J, Lo J, Baker J, Tourassi G (2008) Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. Neural Netw 21(2–3):427–436

Mollineda RA, Sánchez JS, Sotoca JM (2005) Data characterization for effective prototype selection. In: First edition of the Iberian conference on pattern recognition and image analysis (IbPRIA 2005), Lecture Notes in Computer Science 3523, pp 27–34

Orriols-Puig A, Bernadó-Mansilla E (2008) Evolutionary rule-based systems for imbalanced data sets. Soft Comput 13(3):213–225

Peng X, King I (2008) Robust BMPM training based on second-order cone programming and its application in medical diagnosis. Neural Netw 21(2–3):450–457

Pfahringer B, Bensusan H, Giraud-Carrier CG (2000) Meta-learning by landmarking various learning algorithms. In: ICML '00: Proceedings of the seventeenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, pp 743–750

Quinlan JR (1993) C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Mateo–California

Sánchez J, Mollineda R, Sotoca J (2007) An analysis of how training data complexity affects the nearest neighbor classifiers. Pattern Anal Appl 10(3):189–201

Singh S (2003) Multiresolution estimates of classification complexity. IEEE Trans Pattern Anal Mach Intell 25(12):1534–1539

Su CT, Hsiao YH (2007) An evaluation of the robustness of MTS for imbalanced data. IEEE Trans Knowl Data Eng 19(10):1321–1332

Sun Y, Kamel MS, Wong AKC, Wang Y (2007) Cost–sensitive boosting for classification of imbalanced data. Pattern Recognit 40:3358–3378

Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: A review. Int J Pattern Recognit Artif Intell 23(4):687–719

Tang Y, Zhang YQ, Chawla N (2009) SVMs modeling for highly imbalanced classification. IEEE Trans Syst Man Cybern B Cybern 39(1):281–288

Williams D, Myers V, Silvious M (2009) Mine classification with imbalanced data. IEEE Geosci Remote Sens Lett 6(3):528–532

Yang Q, Wu X (2006) 10 challenging problems in data mining research. Int J Inform Tech Decis Mak 5(4):597–604

Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77