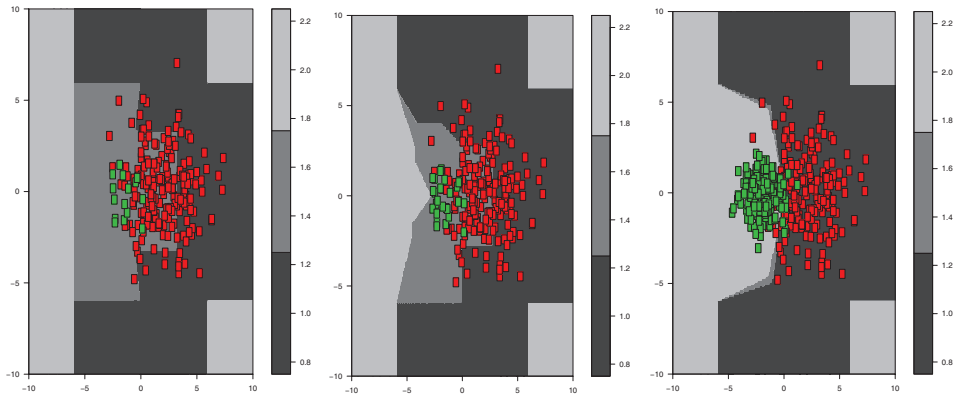

Fig. 3. Boosting fuzzy rule-based with imprecise inputs classifier for different imbalance ratios.

3.3. *Uncertainty in the input variables*

The sensitivity of the decision surfaces to the imbalance ratio causes that small changes in the input variables produce large deviations in the outcome of the learning, as mentioned. In this third set of experiments each instance has been surrounded by a rectangle of size 0.25. The output of the extension of boosting to uncertain data is plotted in Fig. 3. Each point in the domain of the variables has been colored in a different shade of grey: dark grey if it has labelled as majority, light grey if minority or medium grey if non-conclusive.

Non-conclusiveness results from interval-valued numbers of votes of the majority class having non-null intersection with those of the minority class, as mentioned in the preceding section. Observe that a possible deviation of ± 0.25 units in the input has a negligible influence in the problem where IR $= 1$, but most of the points of the combined sample are in the indecision region for IR $= 13.3$ or IR $= 8$. The influence of the imbalance in the spread of the non-conclusive area starts with IR $= 5$ (not shown in the figure). These figures also show that the effect of a preprocessing stage for equalizing the number of instances of each class is relevant for problems with uncertainty in the input variables, as the apparent effect of this uncertainty (the presence of an area where the class of an object cannot be decided) is magnified for imbalance ratios as low as 5.

3.4. *Uncertainty in the class labels: semi-supervised learning*

The fourth set of experiments illustrate the influence of the uncertainty in the class labels in the results of the learning. Points of the majority class to the left of the abscissa $x = 1$, and points of the minority class to the right of $x = -1$, have been labeled with both classes. Given the interpretation of multiple labels followed in this study,[19] this assignment produces a semi-supervised problem,[2] since a fraction of the training set is unlabeled. The results of the low quality data-based boosting

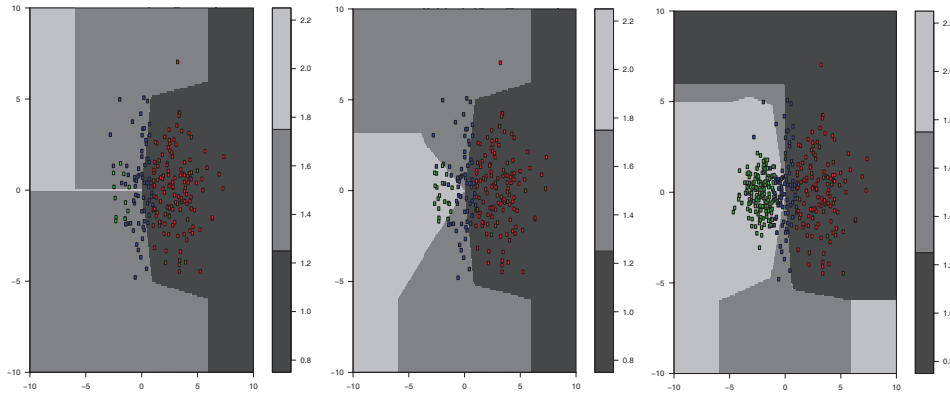

Fig. 4. Boosting fuzzy rule-based with imprecise outputs classifier for different imbalance ratios, without removing the unlabelled examples.

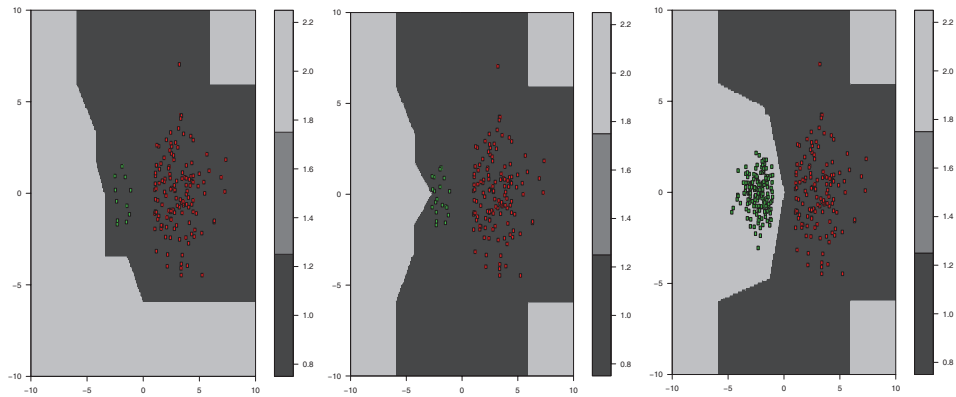

Fig. 5. Boosting fuzzy rule-based with imprecise outputs classifier for different imbalance ratios, after removal of unlabelled examples.

are shown in Fig. 4. The ratio between the proportion of instances of majority and minority classes for the three problems displayed in this last figure are, from left to right, $[2.8, 29.7]$, $[2.3, 17.8]$ and $[0.5, 1.3]$. The first and the second are possibly imbalanced datasets, and in the rightmost problem there is not information enough for determining which class appears more frequently.

Unlabeled examples cannot be handled by the standard Adaboost algorithm, thus in Fig. 5 these were removed. The comparison of the results of Figs. 4 and 5 shows that the extension of Adaboost to low quality data is exploiting the information in the unlabeled elements for IR $= 6.6$; removing these elements causes that the decision surface ignores the minority class, as shown in the middle part of Fig. 5. On the other hand, the decision surfaces for IR $= 1$ are similar in both figures, showing again the usefulness of a preprocessing algorithm in this context.

4. Preprocessing Imbalanced Low Quality Datasets

In the preceding section it was shown that the Adaboost algorithm may not produce a meaningful classifier in the presence of imbalanced datasets. It was also mentioned that a possible solution for this problem consists in combining this algorithm with a preprocessing stage capable of equalizing imprecise datasets.

In previous works,[18] three different categories of preprocessing algorithms for imbalanced problems and low quality data were proposed and their effect over Generic Cooperative-Competitive Learning was compared. In the next section, similar experiments will be carried to determine whether the algorithms that are discussed in the following serve the same purpose:

- **Under-sampling methods:** These obtain a subset of the original dataset by eliminating some of the instances of the majority class. Given that the concept of "minority class" is not precise under our assumptions, as discussed in the preceding section, those algorithms discarding elements of all classes are preferred to those removing elements from either the majority or the minority classes.

  This category comprises the Condensed Nearest Neighbour rule (CNN),[8] Tomek links,[23] One-sided selection (OSS),[12] Neighbourhood cleaning rule (NCL)[13] based on the Wilson's Edited Nearest Neighbour (ENN)[24] and the random under-sampling. In the context of FRBCS for imprecise data, the most relevant extended methods are LQD_NCL, LQD_ENN and LQD_CNN. In particular, the extended ENN algorithm is useful because this algorithm removes elements from all classes.
- **Over-sampling methods:** These obtain a superset of the original dataset by replicating some of the instances of the minority class or creating new ones from the original minority class instances. This is the assumed behavior for the preprocessing algorithm in the previous section, where highly imbalanced datasets were transformed into balanced sets by adding new synthetic instances, drawn from the known statistical distribution of the minority class, until the imbalance ratio is one. In the preceding section this statistical distribution was assumed known, however in practical problems it is approximated by interpolation. The considered methods are Synthetic Minority Over-sampling Technique (SMOTE)[3] and random over-sampling. LQD_SMOTE consistently obtained better results than the random-over-sampling for GCCL. In the next section these experiments will be extended to boosting.
- **Hybrid methods:** These combine over-sampling and under-sampling, and obtain a set by combining the two previous methods. This behavior has been modeled in the preceding section, when elements with a similar probability of being assigned to either class were removed from the training set. For instance, SMOTE+Tomek Link and SMOTE+ENN. The best combination for GCCL was SMOTE+ENN. This last algorithm and SMOTE+Tomek links produced similar results, near to those obtained with SMOTE, however ENN tended to remove more instances than the Tomek links did, so it produced a deeper data

cleaning.[7] The application of these techniques to Adaboost is studied in the following section.

Summarizing this section, the catalog of preprocessing algorithms that will be combined with Adaboost comprises LQD_NCL, LQD_ENN, LDQ_CNN, LQD_SMOTE or LQD_SMOTE+ENN. The performance of the combination of Adaboost and a preprocessing algorithm for low quality data will be assessed in the next section, for different synthetic and real-world datasets.

5. Numerical Results

Nine binary datasets comprising vague data are analyzed in this section. Some problems are possibly imbalanced, thus they have been preprocessed with different algorithms, as mentioned. Part of this data is taken from two different real-world scenarios addressed by us in previous works. These are "future performance of athletes"[17] and "ice adhesion strength measurement from helicopter rotor blades".[1]

The structure of this section is as follows: a brief description of the datasets used in the experimentation is included first. Second, the experimental settings and the metrics used for evaluating the results are discussed, as well as those mechanisms used for removing the uncertainty in the data, that are needed for comparing this algorithm to crisp classification systems. Finally, the results obtained by Adaboost with and without combining it with preprocessing algorithms are explained, and will also be contrasted with other learning algorithms: the cooperative-competitive (GCCL) method,[16] the combination of GCCL and same preprocessing algorithms used in this study[18] and in the last place the cost-oriented GCCL.[17]

5.1. *Description of the datasets*

The name, the number of examples (Ex.), number of attributes (Atts.), the classes (Classes) and the fraction of patterns of each class (%Classes) for each dataset are displayed in Table 1. Observe that the proportions of the different patterns are intervals, because the class labels of some instances are imprecise.[a]

5.1.1. *Athleticism at Oviedo University*

The group of datasets "Athleticism at Oviedo University" comprises eight different sets, whose descriptions are as follows:

(1) Dataset "B200ml-I": This dataset is used to predict whether an athlete will improve certain threshold in 200 meters. All the indicators or inputs are fuzzy-valued and the outputs are sets.

[a]These datasets are available online: `https://ccia35.edv.uniovi.es/datasets`.

Table 1. Summary descriptions of datasets with meta-information.

Dataset	**Ex.**	**Atts.**	**Classes**	**%Classes**
B200mlI	19	4	2	$([0.47, 0.73],\ [0.26, 0.52])$
B200mlP	19	5	2	$([0.47, 0.73],\ [0.26, 0.52])$
Long	25	4	2	$([0.36, 0.64],\ [0.36, 0.64])$
BLong	25	4	2	$([0.36, 0.64],\ [0.36, 0.64])$
100mlI	52	4	2	$([0.44, 0.63],\ [0.36, 0.55])$
100mlP	52	4	2	$([0.44, 0.63],\ [0.36, 0.55])$
B100mlI	52	4	2	$([0.44, 0.63],\ [0.36, 0.55])$
B100mlP	52	4	2	$([0.44, 0.63],\ [0.36, 0.55])$
Ice_shedding	42	7	2	$([0.47, 0.54],\ [0.46, 0.53])$

(2) Dataset "B200mlP": Same dataset as "B200mlI", with an extra feature: the subjective grade that the trainer has assigned to each athlete. All the indicators are fuzzy-valued and the outputs are sets.
(3) Dataset "Long": This dataset is used to predict whether an athlete will improve certain threshold in the long jump. All the features are interval-valued and the outputs are sets. The coach has introduced his personal knowledge.
(4) Dataset "BLong": Same dataset as "Long", measurements or inputs are defined by fuzzy-valued data, obtained by reconciling different measurements taken by three different observers.
(5) Dataset "100ml": Used for predicting whether a threshold in the 100 metres sprint race is being achieved. Each measurement was repeated by three observers. Input variables are intervals and outputs are sets.
(6) Dataset "100mlP": Same dataset as "100mlI", measurements have been replaced by the subjective grade the trainer has assigned to each indicator.
(7) Dataset "B100mlI": Same dataset as "100mlI", measurements are defined by fuzzy-valued data.
(8) Dataset "B100mlP": Same dataset as "100mlP", measurements are defined by fuzzy-valued data.

5.1.2. *Ice adhesion strength measurement from helicopter rotor blades*

This data was taken from an ongoing study at the Pennsylvania State University (U.S.), where ice adhesion strength testing of different coatings for helicopter rotor blades is being conducted under the Boeing Company funding. An Adverse Environment Rotor Test Stand (AERTS) facility is available to reproduce icing conditions for a 9 feet diameter helicopter rotor. The rotor blades are instrumented to quantify ice accretion mass and to calculate the shear adhesion strength of the accreted ice to a given coating.[1] The datasets based on the ice adhesion strength measurement from helicopter rotor blades ("Ice_shedding") are obtained from the study realized by the expert from several runs in the facility. This is a binary problem, whose

Table 2. List of the parameters controlled in the ice adhesion strength measurement from helicopter rotor blades.

Parameter	Unidades	Description
Initial temperature	$^{\circ}C$	Temperature measured before icing
Final temperature	$^{\circ}C$	Temperature read at the end of the test
MVD of the water particle	μm	Size of the water droplets in the cloud
Liquid water concentration	g/m^3	Amount of water in an icing cloud
Revolutions per minute	$cycles/sec$	Speed of the rotor
Roughness	μin	Surface roughness of tested material
Young's modulus	Pa	Linear strain of tested material

inputs (the parameters that need to be controlled, see Table 2) are imprecise, and whose outputs are the groups determined by the expert (A,B).

5.2. *Experimental settings*

All the experiments have been run with a population size of 100, probabilities of crossover and mutation of 0.9 and 0.1, respectively, and limited to 150 generations. The fuzzy partitions of the labels are uniform and their size is 5.

A bootstrap-based experimental design has been used. Each algorithm has been trained with 100 samples with replacement of the training set. Each sample has the same size as the training set, however a number of the train elements are repeated and therefore not all the training instances are used at each repetition. These unused or "out of the bag" elements are used for testing. For making an accurate estimation of the bounds of the classification error in these imprecise elements, 1000 crisp samples compatible with these test elements are generated, and upper and lower bounds of the classification error are assigned to the best and worst result of these tests. For instance, let the training set of an hypothetical classification problem with one input variable comprise five instances

$$\begin{array}{ll} ([2,3] & C_1) \\ ([3,4] & \{C_1, C_2\}) \\ ([2,4] & C_2) \\ ([3,5] & \{C_1, C_2\}) \\ ([2,6] & C_1) \end{array} \tag{27}$$

and let a training resampling be

$$\begin{array}{ll} ([2,3] & C_1) \\ ([2,3] & C_1) \\ ([2,4] & C_2) \\ ([2,4] & C_2) \\ ([3,5] & \{C_1, C_2\}) \end{array} \tag{28}$$

thus its corresponding test partition is

$$\begin{array}{ll}([3,4] & \{C_1, C_2\}) \\ ([2,6] & C_1). \end{array} \tag{29}$$

1000 crisp test sets compatible with this imprecise test partition are generated at random. These test sets have two elements each, of the form

$$\begin{array}{ll}(x_1 & c_1) \\ (x_2 & c_2), \end{array} \tag{30}$$

where $x_1 \in [3,4]$, $c_1 \in \{C_1, C_2\}$, $x_2 \in [2,6]$ and $c_2 = C_1$. The FRBCs will be tested in these 1000 test sets, and the minimum and maximum of the 1000 test error values computed. The whole process is repeated 100 times, and the test error is defined as the mean, using interval arithmetic, of the 100 pairs (minimum test error, maximum test error).

For those experiments involving preprocessed data, the classes were equalized before the learning stage whenever a low quality dataset was possibly imbalanced. The preprocessing algorithms are intended to balance the relative frequencies of all the classes, taking into account the imprecise outputs in the process. The same processing is applied to the 100 bootstrapped resamples of the training set.

5.3. *Evaluating algorithms over a mix of crisp and imprecise data*

When an interval is used to express the result of an algorithm, this interval defines our best bounds about the mean values of the test error. Wider intervals denote a lesser knowledge about the result, which might be any point in that interval. For instance, an algorithm scoring a result $[0, 0.10]$ is not necessarily better than other scoring $[0.05, 0.10]$.

Interval-valued errors are computed by means of the expression that follows (recall Eq. 8):

$$\overline{\text{error}} = \left\{ \frac{1}{m} \sum_{i=1}^{m} e_i \mid e_i \in \overline{e}_i \right\} \tag{31}$$

where

$$\overline{e}_i = \begin{cases} 0 & \text{bclass}(x_i) = \overline{y}_i \text{ and } \#(\overline{y}_i) = 1\,, \\ 1 & \text{bclass}(x_i) \cap \overline{y}_i = \emptyset\,, \\ \{0,1\} & \text{else} \end{cases} \tag{32}$$

and the accuracy is defined as

$$\overline{\text{accuracy}} = 1 \ominus \overline{\text{error}}\,. \tag{33}$$

In words, if an instance is correctly classified without doubt (because the target is not set-valued and the classifier produced the correct class) this instance does not contribute to the error. If the instance is misclassified for sure (because the output of the classifier does not intersect with the set of classes of the target) the contribution to the total error is of $1/m$. Otherwise, the contribution is the pair of values $\{0, 1/m\}$, meaning that the instance might have been misclassified or not. The same course of reasoning has been used to generalize the confusion matrix of a classifier, the geometric mean and all the needed measurements of the quality of a classifier in imbalanced problems.[18]

The statistical comparison between samples of interval or fuzzy data is not a mature field yet. There still exist some controversy in the definition of the most appropriate statistical tests. Some authors propose the use of interval or fuzzy p-values,[5] while other researchers define a crisp distance between fuzzy values and use this distance for formulating statistical tests with crisp p-values between fuzzy data.[11,15] The interpretation used in this study is compatible with the first point of view, however at this moment only bootstrap tests for paired comparisons have been defined.[6]

That being said, in many cases the same information provided by a multiple comparisons test can be obtained with graphical representations.[14] It was decided to use a graphical representation, based on our own extension of the boxplots.[16] In this case, the boxes show the 75% percentile of the maximum and the 25% percentile of the minimum errors. The interval-valued median of the maximum and minimum errors are represented too, as well as the mean of the minimum and maximum fitness, using dotted lines in this last case, because these are not part of an standard boxplot.

5.4. *Analysis of the results and discussion*

In this section the behavior of the Adaboost algorithm for low quality data[19] either with raw or preprocessed information, will be compared with that of other approaches in the literature. In the first place, the preprocessing methods LQD_SMOTE and LQD_SMOTE+ENN will be combined with Adaboost and GCCL,[16] and also with a cost-oriented algorithm.[17] In a second set of experiments, the preprocessing methods LQD_SMOTE, LQD_SMOTE+ENN, LQD_ENN, LQD_CNN and LQD_NCL will be applied in combination with the best classifier in the first stage.

It is remarked that these preprocessing methods are not designed for improving the accuracy, but a different quality metric suitable for imbalanced data. In this work, the geometric mean of the diagonal of the confusion matrix was chosen. Nevertheless, in this section we will show that, under certain conditions, preprocessing the data not only improves this metric but it also improves the misclassification rate.

Table 3. Behaviour of "LQD_GCCL", isolated and in combination with the preprocessing algorithms LQD_SMOTE and LQD_SMOTE+ENN in several datasets (Athleticism and Ice_shedding).

	LQD_GCCL	**LQD_SMOTE+GCCL**		**LQD_SMOTE+ENN+GCCL**	
Dataset	Acc_{Tst}	Acc_{Tst}	GM_{Tst}	Acc_{Tst}	GM_{Tst}
100mlI	[0.622,0.824]	**[0.625,0.826]**	**[0.609,0.806]**	[0.619,0.821]	[0.602,0.796]
100mlP	[0.640,0.824]	**[0.653,0.832]**	**[0.628,0.806]**	[0.646,0.825]	[0.627,0.808]
B100mlI	[0.631,0.828]	**[0.633,0.831]**	**[0.620,0.812]**	[0.581,0.779]	[0.541,0.694]
B100mlP	[0.651,0.840]	**[0.650,0.839]**	[0.619,0.809]	[0.642,0.831]	**[0.633,0.809]**
Mean	[0.636,0.829]	**[0.641,0.832]**	**[0.619,0.808]**	[0.622,0.814]	[0.600,0.776]
Long	[0.410,0.679]	**[0.486,0.755]**	**[0.377,0.605]**	[0.414,0.683]	[0.296,0.467]
BLong	[0.375,0.674]	**[0.444,0.744]**	[0.333,0.596]	**[0.445,0.743]**	**[0.355,0.612]**
B200mlI	[0.527,0.768]	[0.516,0.760]	**[0.402,0.643]**	**[0.594,0.838]**	**[0.429,0.621]**
B200mlP	**[0.520,0.738]**	[0.518,0.736]	**[0.399,0.604]**	[0.514,0.732]	[0.361,0.534]
Mean	[0.458,0.714]	**[0.510,0.748]**	**[0.377,0.612]**	[0.491,0.749]	[0.360,0.558]
Athletics mean	[0.547,0.790]	**[0.575,0.775]**	**[0.498,0.710]**	[0.555,0.778]	[0.480,0.667]
Ice_shedding	[0.550,0.619]	**[0.660,0.729]**	[0.623,0.689]	[0.655,0.724]	**[0.625,0.690]**

Table 4. Behaviour of "LQD_GCCL" with costs, and "LQD_Boost", isolated and in combination with the preprocessing algorithms LQD_SMOTE and LQD_SMOTE+ENN in several datasets (Athleticism and Ice_shedding).

	MR_LQD_GCCL	**LQD_Boost**	**LQD_SMOTE_Boost**		**LQD_SMOTE+ENN_Boost**	
Dataset	Acc_{Tst}	Acc_{Tst}	Acc_{Tst}	GM_{Tst}	Acc_{Tst}	GM_{Tst}
100mlI	[0.620,0.822]	[0.624,0.830]	**[0.639,0.841]**	**[0.637,0.830]**	[0.612,0.814]	[0.604,0.799]
100mlP	[0.633,0.812]	[0.642,0.820]	**[0.659,0.838]**	[0.636,0.808]	[0.654,0.833]	**[0.640,0.816]**
B100mlI	[0.615,0.812]	**[0.644,0.842]**	[0.628,0.825]	**[0.632,0.820]**	[0.606,0.803]	[0.600,0.784]
B100mlP	[0.650,0.839]	[0.650,0.839]	**[0.670,0.859]**	**[0.654,0.840]**	[0.642,0.830]	[0.630,0.808]
Mean	[0.629,0.821]	[0.640,0.832]	**[0.649,0.840]**	**[0.639,0.824]**	[0.628,0.820]	[0.618,0.801]
Long	[0.443,0.712]	[0.492,0.760]	**[0.525,0.794]**	**[0.492,0.772]**	[0.472,0.741]	[0.418,0.683]
BLong	[0.414,0.714]	[0.470,0.770]	[0.459,0.769]	[0.443,0.739]	**[0.489,0.789]**	**[0.447,0.744]**
B200mlI	[0.582,0.822]	**[0.585,0.829]**	[0.511,0.756]	**[0.459,0.691]**	[0.545,0.789]	[0.453,0.670]
B200mlP	[0.567,0.785]	**[0.594,0.812]**	[0.554,0.782]	**[0.525,0.718]**	[0.556,0.773]	[0.441,0.626]
Mean	[0.501,0.758]	**[0.534,0.791]**	[0.512,0.775]	**[0.479,0.730]**	[0.515,0.770]	[0.439,0.680]
Athletics mean	[0.566,0.790]	**[0.587,0.813]**	[0.580,0.807]	**[0.559,0.777]**	[0.571,0.795]	[0.528,0.740]
Ice_shedding	—	[0.639,0.708]	**[0.682,0.751]**	**[0.684,0.745]**	[0.657,0.726]	[0.651,0.712]

5.4.1. *Comparison of a combination of Adaboost and oversampling to other combinations of GFSs with preprocessing, and cost-based GFSs*

The compared accuracies between the extended cooperative-competitive algorithm GCCL[16] (labelled "LQD_GCCL") and Adaboost (labelled "LQD_Boost"),[19] with the preprocessing algorithms LQD_SMOTE and LQD_SMOTE+ENN,[18] are shown in Tables 3 and 4 for binary data. These tables contain the accuracies of the different

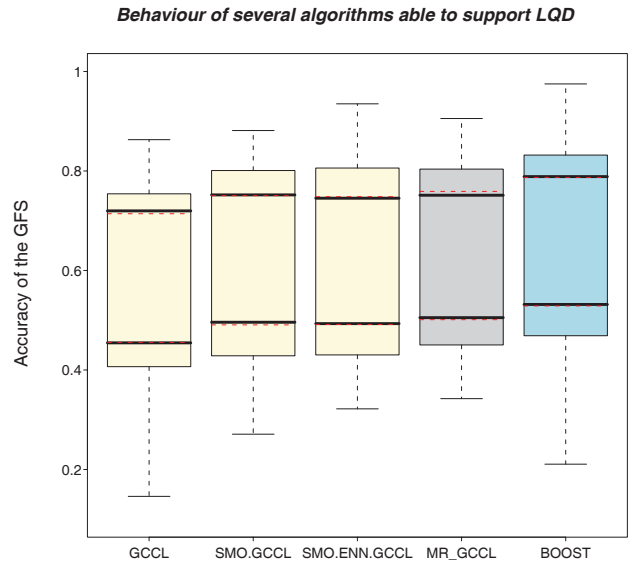

Fig. 6. Accuracies of several GFSs able to support low quality data, highlighting the behaviour of LQD_Boosting.

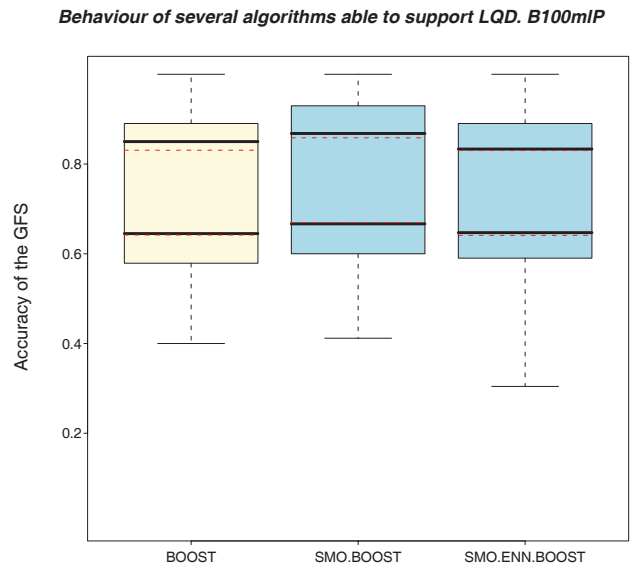

Fig. 7. The combination of preprocessing and Adaboost is not significantly better than the application of Adaboost to the raw data in athleticism datasets.

classifiers in the test data (columns labelled $\mathrm{Acc}_{\mathrm{Tst}}$) and the geometric mean of the diagonal of the confusion matrix (columns labelled $\mathrm{GM}_{\mathrm{Tst}}$), thus higher values are better.

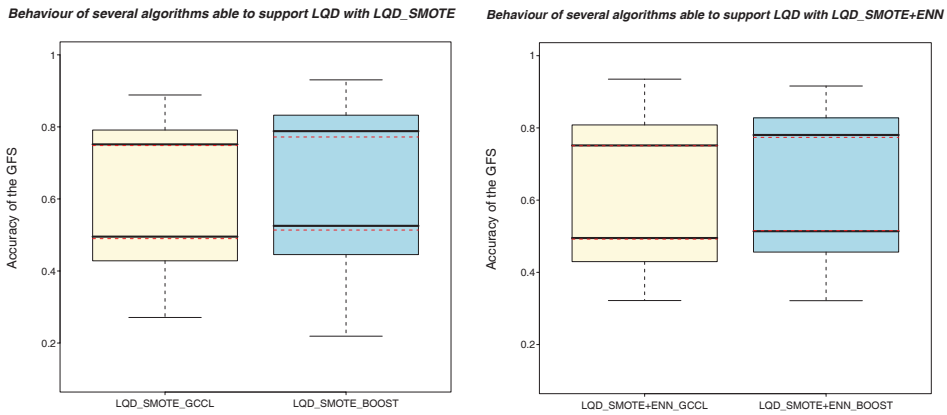

Fig. 8. Behaviour of several GFSs able to support low quality data with LQD_SMOTE and LQD_SMOTE+ENN, highlighting the behaviour of Boosting.

The following conclusions can be drawn from these tables:

- The Adaboost algorithm improves the best combination of GCCL either with preprocessing or costs. In Fig. 6 the mean errors of the datasets being studied were displayed: "200 ml" and "Long" corresponding to LQD_GCCL (with and without the preprocessing stages LQD_SMOTE and LQD_SMOTE+ENN), LQD_boost and LQD_GCCL with a fitness function that penalizes the different misclassifications with the help of a linguistic cost matrix defined by a human expert (MR_LQD_GCCL).
- The preprocessing of the dataset "Ice_shedding" improves the accuracy of GCCL. The results of Adaboost were similar to those of the combination of GCCL and preprocessing in this dataset. However, Adaboost is better than GCCL if the preprocessing is not used.
- The number of misclassifications in athleticism datasets is not reduced when the preprocessing algorithms are applied (LQD_SMOTE and LQD_SMMOTE+ENN, see Fig. 7) in the Adaboost algorithm. Similar to the preceding case, Adaboost is better than GCCL but the combination of GCCL and preprocessing offers a performance similar to that of Adaboost alone.
- Boosting achieves a better accuracy in athleticism and "Ice_shedding" than GCCL, when SMOTE and ENN are applied. This improvement is more noticeable for those datasets with a higher imbalance ratio; in Fig. 8 a combined boxplot of all the datasets is plotted that shows that there is a slight improvement in the final results when SMOTE or SMOTE+ENN are applied prior to Adaboost than prior to GCCL.
- The linguistic quality of the Adaboost is better, because the size of the knowledge base is smaller for the same or better accuracy. This improvement is consistent for all uses of the algorithm, isolated or in combination with any preprocessing, as shown in Table 5.

Table 5. Average number of rules obtained with several preprocessing methods combined with "LQD_GCCL" and "LQD_Boost".

	LQD_GCCL	**LQD_SMOTE+GCCL**	**LQD_SMOTE+ENN+GFS**
100mlI	24	23	23
100mlP	23	26	30
B100mlI	23	33	32
B100mlP	23	25	25
Long	16	35	28
BLong	16	26	25
B200mlI	13	30	25
B200mlP	13	35	22
Ice_shedding	50	50	50
Avg. Rules	22	30	27
	LQD_Boost	**LQD_SMOTE+Boost**	**LQD_SMOTE+ENN+Boost**
100mlI	24	24	24
100mlP	7	7	7
B100mlI	20	20	20
B100mlP	7	7	7
Long	26	26	26
BLong	22	22	22
B200mlI	7	7	—
B200mlP	7	7	7
Ice_shedding	50	50	50
Avg. Rules	19	19	19

- If the geometric mean is considered, LQD_SMOTE is the best technique, alone or in combination with ENN. For instance, the dataset "Long" scores $[0.377, 0.605]$ with GCCL and LQD_SMOTE, and this value is improved up to $[0.492, 0.772]$ if the same preprocessing is applied to boosting. The improvement is higher, from $[0.296, 0.467]$ to $[0.418, 0.683]$ if LQD_SMOTE+ENN is considered. Generally speaking, the best preprocessing algorithm from the point of view of the geometric mean is LQD_SMOTE, which is also the best technique for crisp data.[7,18]

5.4.2. *Adaboost combined with a selection of oversampling and undersampling algorithms*

In Table 6 the behavior of the Adaboost algorithm with respect to the preprocessing methods LQD_SMOTE, LQD_SMOTE+ENN, LQD_ENN, LQD_CNN and LQD_NCL is detailed. The conclusions that can be obtained are:

- LQD_SMOTE is the best alternative even for those cases where the imbalance ratio is low. Our results confirm that the conclusions in the literature about these algorithms[7,18] can also be extended to vague data.
- The use of LQD_CNN is not advocated, since there are cases where the preprocessing degrades the performance of the classification algorithm.

Table 6. Behaviour of boosting with several preprocessing algorithms in low quality and possibly imbalanced binary datasets.

Dataset	LQD_SMT +Boost GM_{Tst}	LQD_SMT+ENN +Boost GM_{Tst}	LQD_ENN +Boost GM_{Tst}	LQD_NCL +Boost GM_{Tst}	LQD_CNN +Boost GM_{Tst}
100mlI	**[0.637,0.830]**	[0.604,0.799]	[0.620,0.809]	[0.610,0.802]	[0.360,0.505]
100mlP	[0.636,0.808]	**[0.640,0.816]**	[0.610,0.772]	[0.591,0.756]	[0.344,0.478]
B100mlI	**[0.632,0.820]**	[0.600,0.784]	[0.614,0.793]	[0.623,0.811]	[0.339,0.483]
B100mlP	**[0.654,0.840]**	[0.630,0.808]	[0.606,0.777]	[0.617,0.792]	[0.414,0.568]
Long	**[0.492,0.772]**	[0.418,0.683]	[0.431,0.656]	[0.445,0.703]	[0.441,0.693]
BLong	[0.443,0.739]	**[0.447,0.774]**	[0.416,0.711]	[0.433,0.736]	[0.256,0.497]
B200mlI	**[0.459,0.691]**	[0.453,0.670]	[0.379,0.522]	**[0.463,0.651]**	[0.302,0.432]
B200mlP	**[0.525,0.718]**	[0.441,0.626]	[0.232,0.327]	[0.341,0.463]	[0.334,0.375]
Athletics mean	**[0.559,0.777]**	[0.528,0.740]	[0.436,0.670]	[0.515,0.714]	[0.348,0.503]
Ice_shedding	**[0.487,0.537]**	[0.405,0.447]	[0.410,0.450]	[0.424,0.465]	[0.334,0.375]

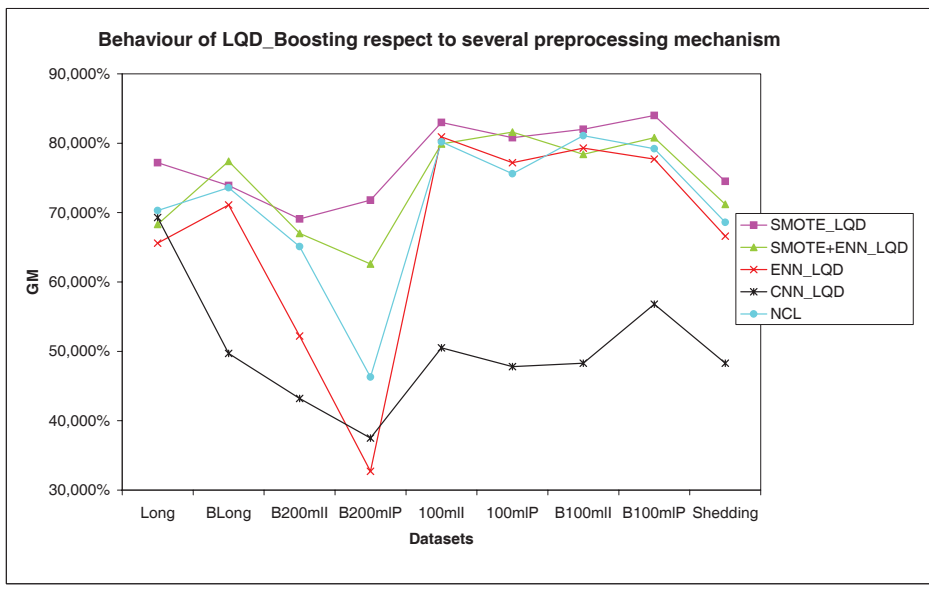

Fig. 9. Behaviour of low quality data in the Boosting algorithm respect to several preprocessing mechanism (upper bound of GM metric).

- LQD_ENN cleans both majority and minority instances, improving LQD_CNN, as expected. The results of this algoritm are intermediate between LQD_NCL y LQD_CNN, as shown in previous works.[18] See Figure 9 for the summarized differences among these preprocessing algorithms in the atlethics datasets, according to the upper bound of the GM metric.

6. Concluding Remarks

An extension of the Adaboost algorithm for learning fuzzy rules from imprecise data was combined with different preprocessing algorithms in this paper. While the primary purpose of these preprocessors is to equalize the dataset and improve metrics of quality different than the misclassification rate, it has been shown that, for binary problems, these techniques also help to improve the fraction of classification errors. The reasons under this improvement have been studied with the help of a synthetical problem, where the sensitivity of Adaboost to different imbalance ratios and uncertainties in both the input and the output variables were assessed.

Acknowledgements

This study has been supported by the Spanish Ministry of Science and Technology and by European Fund FEDER (projects TIN2008-06681-C06-04 and TIN2011-24302).

References

1. E. Brouwers, A. Peterson, J. Palacios and L. Centolanza, Ice adhesion strength measurements for rotor blade edge materials, *67th Annual Forum Proceedings*, American Helicopter Society, Virginia Beach, VA (2011).
2. O. Chapelle, A. Zien and B. Schölkopf (eds.), *Semi-Supervised Learning* (MIT Press, 2006).
3. N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artificial Intelligent Research* **16** (2002) 321–357.
4. N. V. Chawla, N. Japkowicz and A. Kolcz, Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets* **6**(1) (2004) 1–6.
5. I. Couso and L. Sánchez, Defuzzification of fuzzy p-values, *Advances in Soft Computing: Soft Methods for Handling Variability and Imprecision* **48** (2009) 126–132.
6. I. Couso and L. Sánchez, Mark-recapture techniques in statistical tests for imprecise data, *Int. J. Approximate Reasoning* **52**(2) (2011) 240–260.
7. A. Fernández, S. Garcia, M. J. del Jesús and F. Herrera, A study behaviour of linguistic fuzzy rule based classification system in the framework of imbalanced data-sets, *Fuzzy Sets and Systems* **159** (2008) 2378–2398.
8. P. Hart, The condensed nearest neighbor rule, *IEEE Trans. Inform. Theory* **14** (1968) 515–516.
9. F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, *Evolutionary Intelligence* **1** (2008) 27–46.
10. H. Ishibuchi, T. Nakashima and T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, *Fuzzy Sets and Systems* **103**(2) (1999) 223–239.
11. R. Körner, An asymptotic a-test for the expectation of random fuzzy variables, *J. Statistical Planning and Inference* **83** (2000) 331–346.
12. M. Kubat and S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, *Int. Conf. Machine Learning*, 1997, pp. 170–186.

13. J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, T.R. A-2001-2, University of Tampere (2001).
14. R. McGill, J. Tukey and W. Larsen, Variations of box plots, *The American Statistician* **32**(1) (1978) 12–16.
15. M. Montenegro *et al.*, Testing "two sided" hypothesis about the mean of an interval-valued random set, in *Fourth Int. Workshop on Soft Methods in Probability and Statistics* (*SMPS 2008*), Toulouse, France, 2008.
16. A. Palacios, L. Sánchez and I. Couso, Diagnosis of dyslexia with low quality data with genetic fuzzy systems, *Int. J. Approximate Reasoning* **51** (2010) 993–1009.
17. A. Palacios, L. Sánchez and I. Couso, Linguistic cost-sensitive learning of genetic fuzzy classifiers for imprecise data, *Int. J. Approximate Reasoning* **52** (2011) 841–862.
18. A. Palacios, L. Sánchez and I. Couso, Equalizing imbalanced imprecise datasets for genetic fuzzy classifiers, *Int. J. Computational Intelligence Systems*, in press.
19. A. Palacios, L. Sánchez and I. Couso, Boosting of fuzzy rules with low quality data, *J. Multiple-Valued Logic and Soft Computing*, in press.
20. R. Schapire and Y. Singer, Improved boosting algorithms using confidence-rated predictions, *Machine Learning* **37**(3) (1999) 297–336.
21. R. E. Schapire, *Theoretical Views of Boosting and Applications*, Lecture Notes in Artificial Intelligence, Vol. 1720, 1999, pp. 13–25.
22. Y. Sun, M. Kamel, A. Wong and Y. Wang, Cost-sensitive boosting for classification of imbalanced data, *Pattern Recognition* **40** (2007) 3358–3378.
23. I. Tomek, Two modifications of CNN, *IEEE Trans. Syst. Man Comm.* **6** (1976) 769–772.
24. D. R. Wilson, Asymptotic properties of nearest neighbour rules using edited data, *IEEE Trans. Syst. Man Comm.* **2**(3) (1972) 408–421.