International Journal of Hybrid Intelligent Systems 7 (2010) 45–64
DOI 10.3233/HIS-2010-0104
IOS Press

Comparing two genetic overproduce-and-choose strategies for fuzzy rule-based multiclassification systems generated by bagging and mutual information-based feature selection

Oscar Cordón and Arnaud Quirin*,[1]
European Centre for Soft Computing, Edificio Científico-Tecnológico, planta 3. Gonzalo Gutiérrez Quirós, s/n, 33600 - Mieres (Asturias), Spain

Abstract. In [14] we proposed a scheme to generate fuzzy rule-based multiclassification systems by means of bagging, mutual information-based feature selection, and a multicriteria genetic algorithm for static component classifier selection guided by the ensemble training error. In the current contribution we extend the latter component by making use of the bagging approach's capability to evaluate the accuracy of the classifier ensemble using the out-of-bag estimates. An exhaustive study is developed on the potential of the two multicriteria genetic algorithms respectively considering the classical training error and the out-of-bag error fitness functions to design a final multiclassifier with an appropriate accuracy-complexity trade-off. Several parameter settings for the global approach are tested when applied to nine popular UCI datasets with different dimensionality.

1. Introduction

Multiclassification systems (MCSs) (also called multiclassifiers or classifier ensembles) have been shown as very promising tools to improve the performance of single classifiers when dealing with complex, high dimensional classification problems in the last few years [28]. This research topic has become especially active in the classical machine learning area, considering decision trees or neural networks to generate the component classifiers, but also some work has been done using different kinds of fuzzy classifiers (see Section 2.2).

In a previous study [13], we described how fuzzy rule-based multiclassification systems (FRBMCSs) could be generated from classical MCS design approaches such as bagging [4] and random subspace [23] with a basic, heuristic fuzzy classification rule generation method [25]. Later, we analyzed how more advanced feature selection approaches based on the use of mutual information measures – the classical Battiti's mutual information feature selection (MIFS) method [3], a greedy heuristic, and its extension to a greedy randomized adaptive search procedure (GRASP) [19]— allowed us to obtain better performing FRBMCSs [14].

The latter generation approach was embedded into an overproduce-and-choose strategy [36] with the aim to both reduce the final multiclassifier complexity and even try to increase its accuracy by eliminating redundant classifiers. To do so, we proposed a multicriteria genetic algorithm (GA) [9] for static component classifier selection guided by the training error which allowed us to generate linguistic fuzzy rule-based clas-

*Corresponding author. E-mail: arnaud.quirin@softcomputing.es.

[1]This work has been supported by the Spanish Ministry of Education and Science, under grants TIN2005-08036-C05-05 and TIN2006-00829.

sification system (FRBCS) ensembles with different accuracy-complexity trade-offs in a single run.

The resulting FRBMCS design technique thus belong to the genetic fuzzy systems family, one of the most successful approaches to hybridize fuzzy systems with learning and adaptation methods in the last fifteen years [11,12]. It is also quite novel in the fuzzy systems area since no previous work has been done on bagging FRBCSs up to our knowledge.

The aim of the current contribution is to take a step ahead on those first developments by paying more attention to the genetic classifier selection stage. To do so, we will make use of another of the bagging inherent advantages, its ability to test the accuracy of the ensemble without the need of removing any pattern from the data set (i.e., no need to use a validation set) by means of the "Out-Of-Bag" Error (OOBE) [6]. Hence, a new variant of the multicriteria genetic component classifier selection technique will be proposed by adapting the whole FRBMCS design framework in order the latter stage can be guided by the OOBE. In principle, the original fitness function based on the use of the training error could lead to the generation of overfitted FRBCS ensembles due to the use of the same patterns for the individual classifier generation and MCS selection stages. Proceeding in the former way, the FRBCS ensemble configurations selected by the OOBE-guided GA will be evaluated on those instances not considered to learn the component classifiers, i.e., not chosen by the bagging resampling to be included in each bag. We aim to check if the new GA fitness function will allow us to reduce the possible overfitting while still being competitive in terms of accuracy regarding to the initial ensemble.

An exhaustive study will be developed to test the two GA variants based on the use of the two fitness functions guided by the classical training error and OOBE, respectively, on nine popular data sets from the UCI machine learning repository with different characteristics of dimensionality (i.e., with different numbers of examples and features). Several parameter settings for the global approach (e.g., different granularity levels for the fuzzy partitions) will be tested and the performance of the two kinds of genetically selected FRBMCSs will be compared between them, as well as to both the individual FRBCSs and the initial FRBCS ensembles.

This paper is set up as follows. In the next section, the preliminaries required for a good understanding of our work (popular classifier ensemble design approaches, fuzzy MCSs, the need of classifier selection, and the existing GA-based methods to perform it) are reviewed. Section 3 recalls our approach for designing FRBMCSs considering bagging and feature selection, while Section 4 describes the proposed multicriteria GA for component classifier selection. The experiments developed and their analysis are shown in Section 5. Specifically, an example of the analysis of one chromosome regarding the FRBCS ensemble accuracy-complexity trade-off is presented in Section 6. Finally, Section 7 collects some concluding remarks and future research lines.

2. Background and related work

This section explores the current literature related to the generation of a FRBMCS. The techniques used to generate MCSs and FRBCSs are described in Sections 2.1 and section 2.2 respectively. Some ways to reduce the size of the ensembles are described in Section 2.3. The use of GAs, for this purpose, is explored in Section 2.4.

2.1. Related work on MCSs

A MCS is the result of the combination of the outputs of a group of individually trained classifiers in order to get a system that is usually more accurate than any of its single components [28].

According to the existing literature, there are different methods to generate a MCS, all of them based on altering the training process in such way there is disagreement between the component classifiers. Different taxonomies can be considered, but it is usually agreed that there is a well known group comprising approaches considering *data resampling* to obtain different training sets to derive each individual classifier, i.e. *bagging* and *boosting*:

1. *Bagging* [4]: In the *bootstrap aggregation* approach, the individual classifiers are independently learnt from resampled training sets ("bags"), which are randomly selected with replacement from the original training data set, following the statistical bootstrapping procedure. In this way, bagging must be used in combination with "unstable" learning algorithms where small changes in the training set result in large changes in the predictions given by the classifier [5].

2. *Boosting* [43]: Boosting is a family of different methods following the same operation mode: the individual classifiers are generated sequentially by selecting the training set for each of them based on the performance of the previous classifier(s) in the series. Opposed to bagging, the resampling process gives a higher probability of selection to the incorrectly predicted examples by the previous classifiers.

These methods have gain a large acceptance in the machine learning community during the last two decades due to their high performance. Decision trees are the most usual classifier structure considered by them and much work has been done on the topic, although they can be used with any type of classifier (the use of neural networks is also very extended, see for example [35]).

On the other hand, a second group can be found comprised by a more diverse set of approaches which *induct the individual classifier diversity using some ways different from resampling* [52]. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features. *random subspace* [23], where each feature subset is randomly generated, is one of the most representative methods of this kind. Although it was initially proposed for decision tree ensembles, it can be clearly used with any kind of classifier inductor in the same way that resampling approaches. Other generic approaches considering more advanced feature selection strategies can be found in [49,51].

Finally, there are some advanced proposals that can be considered as *combinations of the two groups*. The most extended one could be *random forests* [6], where the individual classifiers are decision trees learnt from a resampled “bag" of examples, a subset of random variables is selected at each construction step, and the best split for those selected variables is chosen for that node.

The interested reader is referred to [2,35] for two reviews for the case of decision tree ensembles (both) and neural networks (the latter), including exhaustive experimental studies. The next subsection reviews the case of the fuzzy MCSs.

2.2. Previous work on fuzzy MCSs

The use of boosting for the design of fuzzy classifier ensembles has been considered in some works, taking the weak learners as fuzzy variants of neural networks [37,50]: as granular models [38], as neuro-fuzzy systems [45], as well as single fuzzy rules [16,24,40].

However, only a few contributions for bagging fuzzy classifiers have been proposed considering, fuzzy adaptive neural networks [37], fuzzy clustering-based classifiers [48], and neuro-fuzzy systems [7] as component classifier structures. Up to our knowledge, no proposal has been made considering FRBCSs.

Two advanced GFS-based contributions are worthy to be mentioned. On the one hand, an FRBMCS design technique is proposed in [1] based on the use of some niching GA-based feature selection methods to generate the diverse component classifiers, and of another GA for classifier fusion by learning the combination weights. On the other hand, another interval and FRBCS ensemble design method based on the use of a single- and multi-objective genetic rule selection is introduced in [33]. In this case, the coding scheme allows an initial set of either intervals or fuzzy rules, considering the use of different features in their antecedents, to be distributed among different component classifiers, trying to make them as diverse as possible by means of two accuracy and one entropy measures. Besides, the same authors presented a previous proposal in [26], where a multi-objective GA generated a Pareto set of FRBCSs with different accuracy-complexity trade-offs to be combined into an ensemble.

Finally, some works making use of fuzzy techniques for classifier ensemble fusion have also been proposed, but they are out out of the scope of the current contribution.

The next two subsections reviews the techniques used to optimize the ensemble size.

2.3. Determination of the optimal set of component classifiers in the MCS

Typically an ensemble of classifiers is post-processed in such a way only a subset of them are kept for the final decision. It is a well known fact that the size of this MCS is an important issue for its trade-off between accuracy and complexity [2,35] and that most of the error reduction occurs with the first few additional classifiers [4,35]. Furthermore, the selection process also participates in the elimination of the duplicates or the poor-performing classifiers.

While in the first studies on MCSs a very small number (around ten) of component classifiers was considered as appropriate to sufficiently reduce the test set prediction error, later research on boosting (that also holds for bagging) suggested that error can be signifi-

cantly reduced by largely exceeding this number [44]. This has caused the use of very large ensemble sizes (for example comprised by 1,000 individual classifiers) in the last few years [2].

Hence, the determination of the optimal size of the ensemble is an important issue for obtaining both the best possible accuracy in the test data set without overfitting it, and a good accuracy-complexity trade-off. In pure bagging and boosting approaches, the optimal ensembles are directly composed of all the component classifiers generated until an specific stopping point, which is determined according to different means (validation data set errors, likelihood, ...). For example, in [2] it is proposed an heuristic method to determine the optimal number guided by the "OOBE" error.

However, there is the chance that the optimal ensemble is not comprised by all the component classifiers first generated but on a subset of them carrying a larger degree of disagreement/diversity. This is why different classifier selection methods [15] has been proposed. They could be mainly grouped in two kinds of strategies. The first one is named the *overproduce-and-choose strategy* (OCS) [36], also known as the *test-and-select methodology* [47] or the *static strategy* [42] in the literature. In this strategy, a large set of classifiers is produced and then selected to extract the best performing subset. The second one is named the *dynamic classifier selection approach* (DCS) [21]. In this approach, the accuracy of each classifier surrounding the region of the feature space where the unknown pattern to be classified is located is previously estimated, and the best ones are selected to classify that specific pattern.

GAs have been commonly used for the both strategies as we will show in the following subsection.

2.4. Related work on genetic selection of FRBMCSs

GAs are a popular technique used to select the classifiers, especially within the OCS strategy. Usually, performance, complexity and diversity measures considered used as search criteria. Complexity measures are employed to increase the interpretability of the system whereas diversity measures are used to avoid overfitting.

Among the different genetic OCS proposals, we can remark the following ones. In [34], a hierarchical multiobjective GA (MOGA) algorithm, performing feature selection at the first level and classifier selection at the second level, is presented which outperforms classical methods for two handwritten recognition problems. The MOGA allows both performance and diversity to be considered for MCS selection. In [22] a GA is used to select from seven diversity heuristics for k-means cluster-based ensembles and the ensemble size is also encoded in the genome. Even if the experiments conducted on 18 datasets showed that no particular combination of heuristics have been chosen by the GA across all the datasets, this study dealt with the three families of criteria: performance, complexity and diversity. Another extensive comparison between 15 different classifiers, 27 datasets, 7 search methods (among them three evolutionary algorithms) and 16 selection criteria (diversity measures and classifier error) is presented in [39], but the conclusion does not agree with the other studies: the diversity measures seem not to be useful to improve the error rate. In the study of Martínez-Munoz et al. [31], a GA is compared to five other techniques for ensemble selection. Even if the performance of the GA was the worst obtained, they showed that while selecting a small subset of classifiers, the generalization error was significantly decreased. In [20], the authors developed a multidimensional GA to optimize two weight-based models, in which the weights are assigned to each classifier or to each class. They applied their system to 6 different classifiers (only linear and quadratic classifiers are explored), but on only two small datasets and without comparing to the results obtained on a single classifier. Another study [27] aimed to develop a weighted-based GA for combining diverse classifiers, driven from machine-learning techniques or human experts. The authors obtained promising results, but they applied their methodology only on a small dataset, due to the difficulty of collecting a large expert dataset. Our own previous studies [13,14] also consider a multicriteria GA for the ensemble selection in an OCS fashion, with performance (training error) and complexity as criteria to guide the GA.

Some conclusions drawn in the cited papers are similar to all of them: in general, the performance obtained after the genetic selection of an ensemble outperforms the initial MCS, while quite drastically simplifying the system. But in all of them, the error rate is measured on the initial training set or a pre-defined validation set. The aim of the current contribution is to analyze a new selection methodology based on the use of the OOBE to select the ensembles by the means of a GA, taking advantage of the bagging approach.

The other strategy, the DCS approach, is still less extended in the specialized literature. One of the available studies, presented in [41], is a comparison of a single-objective GA and a MOGA for 14 different ob-

jective functions of the mentioned three families of criteria (12 diversity measures, the training error, and the number of classifiers as a complexity measure). The authors applied their study on only one dataset, a digit handwritten recognition problem with 10 classes and 118,735 instances. They conclude saying that the training error is the best criterion for a single GA and a combination of training error and one diversity measure is the best criterion for a MOGA. In [42], the two OCS and DCS strategies are combined to form a *dynamic overproduce-and-choose strategy* to allow, respectively, the generation of a set of highly accurate ensembles, and to select the one with the highest degree of confidence, in a two-step process. This strategy outperforms both a static strategy and the initial ensemble of classifiers on seven datasets. In [18], the authors proposed a GA selecting the *votes* of each classifier in an ensemble for its reliability to classify each class, instead of discarding the classifiers at a whole. They obtained good results with respect to static strategies, but they tested their proposal on only one application. In [30], an ensemble of neural networks are evolved using an evolutionary algorithm based on negative correlation, in order they learn different parts of the training set. Very competitive results are presented, but in only two datasets.

We can also notice that GAs are also popular techniques for feature selection. For instance, in [49], a GA is compared to four other techniques for feature selection on a high number of datasets (21) and using different diversity measures. For all the experimentations, the GA outperformed the remaining methods regarding the MCS test accuracy.

3. Bagging and feature selection-based FRBMCSs

In this section we will both detail how the individual classifiers and the FRBMCSs are designed. Figure 1 shows the framework of the whole approach. A normalized dataset is split into two parts, a training set and a test set. The training set is submitted to an instance selection and a feature selection procedure in order to provide individual training sets (the so-called *bags*) to train simple FRBCSs (through the method described in Section 3.1). The instance selection and the feature selection procedures are described in Section 3.2. After the training, we got an initial MCS, which is validated using the training and the test errors (*Ensemble Training Error* and *Ensemble Test Error*), as well as a measure of complexity based on the total number of rules in the FRBCSs. This ensemble is selected using a GA (described in Section 4) using either the Training Error or the OOBE. The final MCS is validated using different accuracy (Training Error, OOBE, Test Error) and complexity measures (number of classifiers, total number of rules).

3.1. Individual FRBCS composition and design method

The FRBCSs considered in the ensemble will be based on fuzzy rules with a class C_j and a certainty degree CF_j in the consequent:

$$\text{Rule } R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{and } x_n \text{ is } A_{jn}$$
$$\text{then Class } C_j \text{ with } CF_j; \quad j = 1, 2, ..., N,$$

and they will take their decisions by means of the single-winner method, which gives as classifier output the class associated to the rule with the largest value for the product of its firing degree and the certainty degree [10,25]. The *and* conjunctive operator in the rule antecedent is modeled by the product T-Norm. This fuzzy reasoning method has been selected due to its high simplicity and interpretability. The use of other more advanced ones [10] is left for future works.

To derive the fuzzy knowledge bases, one of the heuristic methods proposed by Ishibuchi et al. in [25] is considered. All the fuzzy rule derivation methods in this family start from a fuzzy partition definition for each variable, and are based on generating a fuzzy rule R_j for each fuzzy input subspace A_j where at least a training example is located. The consequent class C_j and certainty degree CF_j are statistically computed from all the examples located in the specific subspace $D(A_j)$ (grid-based methods, see [8]).

In the chosen method, C_j is computed as the class h with maximum confidence according to the rule compatible training examples $D(A_j) = \{x_1, \ldots, x_m\}$. That confidence value for every class is computed as:

$$c(A_j \Rightarrow Class\ h) = \frac{|D(A_j) \bigcap D(Class\ h)|}{|D(A_j)|} =$$
$$= \frac{\sum_{p \in Class\ h} \mu_{A_j}(x_p)}{\sum_{p=1}^{m} \mu_{A_j}(x_p)} h = 1, 2, \ldots, M;$$

CF_j is obtained as the difference between the confidence of the consequent class and the sum of the confidences of the remainder (called CF_j^{IV} in [25]):

$$CF_j = c(A_j \Rightarrow Class\ C_j) - \sum_{h=1; h \neq C_j}^{m} c(A_j \Rightarrow Class\ h).$$

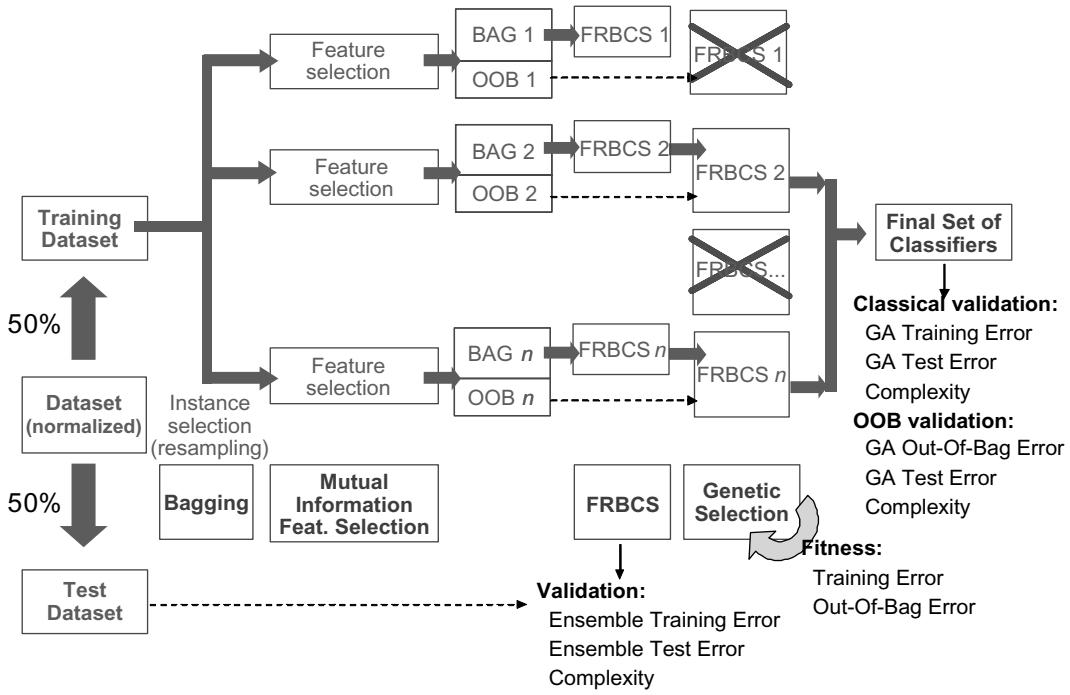

Fig. 1. Our framework: after the instance and the feature selection processes, the individual FRBCSs are learned. Finally, they are selected by a GA to compose the FRBMCS.

This method is good for our aim of designing FRBMCSs since it is simple and quick. Besides, we experimentally checked it fulfils the most important requirement for creating an ensemble, that of being unstable enough to generate uncorrelated classifiers when run on different bootstrapped samples of a training set.

However, it carries some drawbacks. The first one is that of generating an excessive number of rules, which will make impossible to run it on pure bagging approaches without feature selection when the number of problem attributes and the granularity are high.

On the other hand, it is well known that heuristic, data-driven fuzzy classification rule generation methods result in FRBCSs with a low accuracy by themselves, which will also affect the final accuracy of the generated FRBMCSs. Even so, we prefer considering it in this study due to the said advantages.

3.2. FRBMCS design approaches

In this contribution we are applying a bagging approach combined with a feature selection method in order to generate FRBMCSs [14]. Three different feature selection methods, random subspace and two variants of Battiti's MIFS, greedy and GRASP, are considered.

As said before, *random subspace* [23] is a method in which we select randomly a set of features from the original dataset. The greedy Battiti's MIFS method [3] is based on a forward greedy search using the Mutual Information measure [46], with regard to the class. This method selects the set S of the most informative features about the output class which cannot be predicted with the already selected features. The Mutual Information $I(C,F)$ for a given feature F is defined as:

$$I(C,F) = \sum_{c,f} P(c,f) \log \frac{P(c,f)}{P(c)P(f)} \qquad (1)$$

where $P(c)$, $P(f)$ and $P(f)$ are respectively the values of the density function for the class and the feature variables, and the joint probability density. In the MIFS method, we select as a first feature f, the one that maximizes $I(C,f)$, and then the features f that maximize $Q(f) = I(C,f) - \beta \sum_{s \in S} I(f,s)$, until S reaches the desired size. β is a coefficient to set up the penalization on the information brought by the already selected features.

The MIFS-GRASP variant is an approach where the set is generated by iteratively adding features randomly chosen from a Restricted Candidate List (RCL) composed of the best τ percent decisions according to the

Q measure. Parameter τ is used to control the amount of randomness injected in the MIFS selection. With $\tau = 0$, we get the original MIFS method, while with $\tau = 1$, we get the random subspace method.[1]

For the bagging approach, the bags are generated with the same size as the original training set, as commonly done. In every case, all the classifiers will consider the same fixed number of features.

Finally, no weights will be considered to combine the outputs of the component classifiers to take the final MCS decision, but a pure voting approach will be applied: the ensemble class prediction will directly be the most voted class in the component classifiers output set.

4. A multicriteria genetic-based MCS selection method

As described in Section 2.3, several studies have demonstrated most of the gain in a MCS's performance comes in the first few classifiers combined [2,35], and several proposals have been made either to determine when enough component classifiers have been generated for the ensemble or to select a subset of them with a large degree of disagreement. In the current contribution we propose to use a multicriteria GA in order to be able not only to obtain a single solution, i.e., a classifier ensemble composition, but a list of possible MCS designs, ranked by their quality, *from a single chromosome*.

In one of our previous studies [13], we used this GA approach considering the likelihood instead of the training error as the fitness function guiding criterion, as it seems to be more appropriate when basic feature selection methods are used. In this extension of the study published in [14], we are comparing two approaches for the fitness function. In the first one, we use the same training set as the one used to generate the bags on which each single classifier are trained. In the following, we will refer to it as the *Training Error-based Fitness Function* (TEFF).

This training error is computed as follows. Let $h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_l(\mathbf{x})$ be the outputs of the component classifiers of the selected ensemble for an input value $\mathbf{x} = (x_1, \ldots, x_n)$. For a given sample $\{(\mathbf{x}^k, C^k)\}_{k\in\{1\ldots m\}}$, the training error of that MCS is:

$$\frac{1}{m} \cdot \#\{k \mid C^k \neq \arg_{j\in\{1\ldots M\}} h_j(\mathbf{x}^k)\} \qquad (2)$$

In the second one, we allow the GA to compute the error measure of an ensemble by using only the "Out-Of-Bag" instances of each classifier, i.e. the equation above is computed considering only the instances $\mathbf{x}^k$ not found into the bag k (see Fig. 1). In the following, we will refer to it as the *Out-Of-Bag Error-based Fitness Function* (OOBEFF). Not only these Out-Of-Bag instances have not been seen during the learning of each individual classifier, thus leading to less overfitting, but also the size of the datasets used for the genetic selection is reduced as only a 37% of the instances from the original training set is comprised in the Out-Of-Bags in average [4], thus improving the selection stage computation time.

The GA looks for an optimal ordering of the component classifiers, so that the most relevant classifiers have the lowest indexes and those redundant members that can be safely discarded are in the last places. The coding scheme is thus based on an order-based representation, a permutation $\Pi = \{j_1, j_2, \ldots, j_l\}$ of the l originally generated individual classifiers. In this way, each chromosome encodes l different solutions to the problem, based on considering a "basic" MCS comprised by a single classifier, that one stored in the first gene; another one composed of two classifiers, those in the first and the second genes, and so forth.

The degree to which a permutation fulfills this goal is measured by means of the *cumulative error* of the ensemble, defined as the vector containing the training or Out-Of-Bag error values (depending on the considered approach) of the first classifier; the subset formed by the first and the second; and so on. The fitness function is thus multicriteria, being composed of an array of l values, $L^i = L'_{\{j_1,j_2,\ldots,j_i\}}$, corresponding to the cumulative error of the l mentioned MCS designs. The best chromosome is that member in the population with the lowest minimum cumulative error. Then, the final design is the MCS comprising the classifiers from the first one to the one having the minimum cumulative error value (although any other design not having the optimal error but, for example, showing a lowest complexity can also be directly extracted, see Fig. 7 in Section 6).

Instead of using a Pareto-based approach [9], a lexicographical order is considered to deal with the mul-

[1] We should note that, although this procedure is called MIFS-GRASP, it does not completely match the usual GRASP structure [19] since it does not include a second stage with a local optimizer. In our case, the use of only the first randomized greedy stage is a better choice since more diverse feature subsets (and thus more diverse individual classifiers) will be obtained at a lowest computational cost.

Table 1
Data sets considered

Data set	#attr.	#examples	#classes
Pima	8	768	2
Glass	9	214	7
Vehicle	18	846	4
Sonar	60	208	2
Breast	9	699	2
Heart	13	270	2
Yeast	8	1,484	10
Phoneme	5	5,404	2
P-Blocks	10	5,473	5

ticriteria optimization, since we think it better matches our scenario. When comparing two chromosomes, one is better than the other if it takes a better (lower) minimum value of the cumulative error. In case of tie, the first positions of the fitness arrays are compared. If both first positions are of equal value, the second positions are compared, and so on.

To increase its convergence rate, the GA works following a steady-state approach. The initial population is composed of randomly generated permutations. In each generation, a tournament selection of size 3 is performed, and the two winners are crossed over to obtain a single offspring that directly substitutes the loser. In this study, we have considered OX crossover and the usual exchange mutation [32].

5. Experiments and analysis of results

In this section, we discuss the performance obtained by a single FRBCS, an FRBMCS and two GA-selected FRBMCSs on nine chosen data sets.

5.1. Experimental setup

To evaluate the performance of the FRBMCSs generated, we have selected nine data sets from the UCI machine learning repository (see Table 1). In order to compare the accuracy of the considered classifiers, we used Dietterichs 5×2-fold cross-validation (5×2-cv), which is considered to be superior to paired k-fold cross validation in classification problems [17]. In 5×2-cv, five stratified two-fold cross-validations are performed. The data set is randomly broken into two halves, and one is used for training and the other for testing and *vice versa*. The procedure is repeated five times, each with a new half/half partition, and a single index is finally computed by averaging the ten test errors.

Three different granularities, 3, 5 and 7, are tested for the single FRBCS derivation method, for feature sets of size 5 selected by means of three approaches: the greedy Battiti's MIFS filter feature selection method [3], the Battiti's method with GRASP (with τ equal to 0.5, see Section 3.2), and random subspace [23]. Battiti's method has been run by considering a discretization of the real-valued attribute domains in ten parts and setting the β parameter to 0.1.

The FRBMCSs generated are initially comprised by 50 classifiers. The GA for the component classifier selection works with a population of 50 individuals and runs during 50 generations. The mutation probability considered is 0.05.

All the experiments have been run in an Intel quadricore Pentium 2.4 GHz computer with 2 GBytes of memory, under the Linux operating system.

5.2. Single FRBCS vs. bagging + feature selection FRBMCSs

The statistics (5×2-cv error, number of rules, and run time required for each run, expressed in seconds) for the single FRBCSs are collected in Table 2. There are three subtables for each feature selection method considered: Battiti's method (greedy), Battiti's method combined with GRASP with 50% of randomness (GRASP 0.50), and the random subspace method. The best results for a given feature selection method are shown in bold and the best values overall are outlined.

In our previous study [13], we showed that the best results for the four datasets considered in that contribution were obtained using 5 labels for the smaller problems (pima and glass), and 7 labels for the largest ones (vehicle and sonar). This is not the case with this larger study as sonar and some other problems with a higher dimension (breast and heart) give their best results with 3 labels using respectively the GRASP 0.50 and the greedy approaches. For the largest problems (yeast, phoneme, and p-blocks), the best performance is still obtained with the largest number of labels.

Overall, the best single FRBCS results were obtained with GRASP 0.50 for four datasets, and with the greedy approach for only two datasets (in the remaining three datasets, these two approaches gave the same results). Pure random subspace only achieves a draw in the best results for a single dataset. This confirms the fact that controlled randomness in the feature selection process is useful when combined with FRBCSs.

The results for the FRBMCSs of 50 classifiers generated from the three different feature selection approaches considered are shown in Table 3, which present the same structure than Table 2.

Table 2
Results for the single FRBCSs with feature selection

		Greedy								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
3 labels	5 × 2-cv	0.266	0.446	0.549	**0.261**	**0.048**	**0.197**	0.565	0.287	0.089
5 #attr.	#rules	178.50	135.30	136.40	146.60	232.90	98.60	212.00	240.70	118.00
	time	0.08	0.04	0.12	0.08	0.07	0.02	0.29	0.57	0.62
5 labels	5 × 2-cv	**0.246**	**0.376**	0.430	0.287	0.064	0.227	0.481	0.207	0.077
5 #attr.	#rules	682.70	291.00	437.60	615.20	1128	198.90	937.40	1181	358.20
	time	0.42	0.25	0.65	0.16	0.40	0.11	2.57	3.16	4.27
7 labels	5 × 2-cv	0.262	0.414	**0.402**	0.291	0.136	0.258	**0.442**	**0.181**	**0.067**
5 #attr.	#rules	1600	431.20	1021	1218	1752	277.20	2012	3180	731.90
	time	1.75	1.32	3.27	0.52	1.49	0.53	13.84	14.27	23.23

		Random Subspace								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
3 labels	5 × 2-cv	0.265	0.457	0.512	**0.319**	**0.051**	**0.252**	0.616	0.287	0.088
5 #attr.	#rules	161.80	109.50	154.50	174.50	207.40	67.60	66.10	240.70	139.00
	time	0.07	0.03	0.12	0.08	0.07	0.02	0.21	0.57	0.65
5 labels	5 × 2-cv	**0.262**	0.435	0.460	0.329	0.083	0.262	0.539	0.207	0.081
5 #attr.	#rules	604.20	259.60	587.80	773.60	804.90	118.70	167.90	1181	504.40
	time	0.36	0.24	0.67	0.17	0.34	0.11	2.10	3.17	4.53
7 labels	5 × 2-cv	0.276	**0.418**	**0.415**	0.340	0.150	0.279	**0.496**	**0.181**	**0.071**
5 #attr.	#rules	1432	410.90	1266	1536	1261	164.50	297.40	3180	1124
	time	1.66	1.32	3.37	0.63	1.35	0.52	11.82	14.31	24.20

		GRASP $\tau = 0.50$								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
3 labels	5 × 2-cv	0.267	0.447	0.546	0.316	**0.047**	**0.209**	0.565	0.287	0.089
5 #attr.	#rules	179.50	137.00	135.80	169.00	233.30	92.10	212.00	240.70	120.20
	time	0.09	0.04	0.12	0.09	0.08	0.03	0.29	0.57	0.63
5 labels	5 × 2-cv	**0.246**	**0.375**	0.425	**0.314**	0.066	0.237	0.481	0.207	0.077
5 #attr.	#rules	682.70	293.50	418.90	752.70	1187	176.90	937.40	1181	367.00
	time	0.39	0.26	0.63	0.17	0.41	0.11	2.57	3.17	4.30
7 labels	5 × 2-cv	0.266	0.423	**0.399**	0.317	0.145	0.270	**0.442**	**0.181**	**0.065**
5 #attr.	#rules	1599	437.20	907.50	1470	1886	250.20	2012	3180	757.00
	time	1.71	1.34	3.25	0.55	1.52	0.52	13.82	14.35	23.32

Comparing the best results for each dataset for the single FRBCS and the FRBMCSs, the FRBMCS outperforms the single FRBCS in five cases (pima, vehicle, sonar, breast, and heart), the FRBCS outperforms the FRBMCS in three cases (glass, yeast, and phoneme) and there is a tie in the remaining case (p-blocks). As can be seen, there is no clear methodology to get the best FRBMCS: all feature selection approaches give their best result on at least one dataset, and there is no optimal granularity for all of the datasets. But in general, the highest number of times the best results are obtained is with the random subspace method (4 datasets), followed the GRASP 0.50 approach (3 datasets), plus the additional draw in the phoneme dataset. The same sequence is obtained using respectively 7 labels (4 datasets) and 5 labels (3 datasets).

Finally, over all the different feature selection approaches, the bagging+feature selection approach allowed a decrease of 6% of the test error, while reducing by 13% the average size of the individual classifiers. The best example is produced on the breast dataset, with the random subspace approach, using 5 labels and 5 attributes, in which the bagging allowed us to get a decrease of a 40% in the test error, while reducing the size of the rule base by a 4%. The best reduction of the rule base was obtained on the p-blocks dataset (-50%), with the random subspace approach, using 7 labels and 5 attributes, but at the cost of increasing the test error by 17%.

The Mann-Whitney U test, also known as the Wilcoxon Ranksum test, has been used for a deeper insight of the results. Unlike the commonly used t test,

Table 3
Results for the FRBCS ensembles

		Bagging + Greedy								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.261	0.463	0.525	0.255	**0.048**	**0.187**	0.576	0.287	0.088
3 labels	#rules	8578	6208	6843	7282	11067	4144	10080	11913	5404
5 #attr.	avg. #rules	171.55	124.16	136.87	145.65	221.34	82.87	201.60	238.26	108.07
	time	3.43	1.51	4.87	2.52	3.48	0.96	13.81	27.99	26.27
	5 × 2-cv	**0.235**	**0.396**	0.400	**0.240**	0.057	0.207	0.481	0.207	0.077
5 labels	#rules	29405	12877	22177	26769	43019	7630	41392	54448	15870
5 #attr.	avg. #rules	588.11	257.54	443.55	535.37	860.37	152.60	827.83	1089	317.39
	time	17.93	12.11	31.21	6.66	18.50	5.21	128.61	161.12	211.93
	5 × 2-cv	0.243	0.430	**0.375**	0.262	0.160	0.257	**0.444**	**0.182**	**0.066**
7 labels	#rules	64891	18633	48479	49587	61451	10430	85372	143827	31700
5 #attr.	avg. #rules	1298	372.66	969.58	991.74	1229	208.61	1707	2877	634.00
	time	84.70	67.36	166.51	24.72	71.86	25.58	699.56	712.24	1164

		Bagging + Random Subspace								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.299	0.450	0.453	0.250	**0.035**	**0.166**	0.612	0.287	0.100
3 labels	#rules	7936	5671	8008	8174	10479	3821	4608	11913	5641
5 #attr.	avg. #rules	158.71	113.42	160.16	163.47	209.58	76.43	92.16	238.26	112.82
	time	3.34	1.49	5.06	2.58	3.38	1.00	11.30	29.16	26.84
	5 × 2-cv	**0.260**	0.430	0.378	**0.221**	0.050	0.201	0.491	0.207	0.090
5 labels	#rules	27199	11998	30799	31824	38502	7471	13732	54448	15209
5 #attr.	avg. #rules	543.97	239.96	615.97	636.47	770.04	149.42	274.65	1089	304.17
	time	17.64	11.94	33.91	7.13	17.70	5.33	112.41	161.80	205.27
	5 × 2-cv	0.263	**0.402**	**0.330**	0.241	0.208	0.223	**0.444**	**0.182**	**0.083**
7 labels	#rules	59824	17999	67936	57298	54426	10659	24388	143827	28178
5 #attr.	avg. #rules	1196	359.98	1359	1146	1089	213.18	487.77	2877	563.57
	time	82.12	66.06	174.24	25.57	70.38	25.36	621.38	725.05	1130

		Bagging + GRASP $\tau = 0.50$								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.262	0.464	0.494	0.246	**0.040**	**0.180**	0.578	0.287	0.088
3 labels	#rules	8609	6289	7362	7951	11228	4335	9936	11913	5738
5 #attr.	avg. #rules	172.18	125.77	147.24	159.03	224.56	86.70	198.72	238.26	114.77
	time	3.45	1.53	4.91	2.57	3.54	1.07	13.87	28.37	27.30
	5 × 2-cv	**0.234**	**0.405**	0.399	**0.220**	0.056	0.210	0.482	0.207	0.076
5 labels	#rules	29748	13302	25578	30068	44695	8243	40314	54448	16962
5 #attr.	avg. #rules	594.95	266.04	511.56	601.36	893.90	164.86	806.29	1089	339.25
	time	18.05	12.23	32.79	6.96	18.75	5.24	128.66	159.31	214.29
	5 × 2-cv	0.247	0.425	**0.353**	0.242	0.186	0.247	**0.445**	**0.182**	**0.065**
7 labels	#rules	65802	19272	54721	54684	63352	11480	83519	143827	33599
5 #attr.	avg. #rules	1316	385.45	1094	1094	1267	229.60	1670	2877	671.98
	time	85.27	68.27	170.48	25.49	72.93	25.45	698.20	713.56	1168

the Wilcoxon test does not assume normality of the samples [29], which would be unrealistic in the case of the UCI datasets. The significance tables presented in this paper contain three symbols: '$+$' when the significance is favorable for the method in the row, '$-$' when the significance is favorable for the method in the column, and '$=$' when there is no significance about which method is better than the other. When not specified, the confidence level considered for the null hypothesis rejection is 5%. Table 4 shows the statistical significance for the methodology used to create the classifier ensembles. Each set of parameters is compared and the one giving the best result for a given dataset is marked with a star '*'. The experimental design is shown in Fig. 2. As said before, in general, the random subspace method performs well, and the same could be said independently for the approaches using 7 labels. Here the statistical test proves it for two datasets: the com-

Table 4

Statistical test for the comparison of the FRBMCS design methodologies (see Fig. 2). For each dataset, the best method is marked ('*') and the others are compared to it

	Approach	Greedy			GRASP $\tau = 0.50$			Random Subspace		
		3 labels	5 labels	7 labels	3 labels	5 labels	7 labels	3 labels	5 labels	7 labels
Pima	$\mu \pm \sigma$	0.261 ± 0.0220	0.235 ± 0.0156	0.243 ± 0.0167	0.262 ± 0.0233	0.234 ± 0.019	0.247 ± 0.0156	0.299 ± 0.0284	0.260 ± 0.0220	0.263 ± 0.0138
	Symbol	+	=	=	+	*	=	+	+	+
Glass	$\mu \pm \sigma$	0.463 ± 0.0491	0.396 ± 0.0568	0.430 ± 0.0584	0.464 ± 0.0489	0.405 ± 0.046	0.425 ± 0.0648	0.450 ± 0.0702	0.430 ± 0.0621	0.402 ± 0.0519
	Symbol	+	*	=	+	=	=	+	=	=
Vehicle	$\mu \pm \sigma$	0.525 ± 0.0247	0.400 ± 0.0267	0.375 ± 0.0150	0.494 ± 0.034	0.399 ± 0.0339	0.353 ± 0.0239	0.453 ± 0.0209	0.378 ± 0.0244	0.330 ± 0.0179
	Symbol	+	+	+	+	+	+	+	+	*
Sonar	$\mu \pm \sigma$	0.255 ± 0.0355	0.240 ± 0.0301	0.262 ± 0.0323	0.246 ± 0.0605	0.220 ± 0.0445	0.242 ± 0.0356	0.250 ± 0.046	0.221 ± 0.0336	0.241 ± 0.0483
	Symbol	=	=	+	=	*	=	=	=	=
Breast	$\mu \pm \sigma$	0.0478 ± 0.0092	0.0575 ± 0.0160	0.160 ± 0.0187	0.0403 ± 0.00965	0.0561 ± 0.0148	0.186 ± 0.0245	0.0355 ± 0.00468	0.0503 ± 0.00959	0.209 ± 0.0214
	Symbol	+	+	+	=	+	+	*	+	+
Heart	$\mu \pm \sigma$	0.187 ± 0.0322	0.207 ± 0.0376	0.257 ± 0.0541	0.180 ± 0.0331	0.210 ± 0.0350	0.247 ± 0.0469	0.166 ± 0.0335	0.201 ± 0.035	0.223 ± 0.0513
	Symbol	=	+	+	=	+	+	*	+	+
Yeast	$\mu \pm \sigma$	0.576 ± 0.0445	0.481 ± 0.0199	0.444 ± 0.0131	0.578 ± 0.0477	0.482 ± 0.0198	0.445 ± 0.0144	0.612 ± 0.0314	0.491 ± 0.0204	0.444 ± 0.0124
	Symbol	+	+	=	+	+	=	+	+	*
Phoneme	$\mu \pm \sigma$	0.287 ± 0.0057	0.207 ± 0.00614	0.182 ± 0.00933	0.287 ± 0.0057	0.207 ± 0.00614	0.182 ± 0.00933	0.287 ± 0.0057	0.207 ± 0.00614	0.182 ± 0.00933
	Symbol	+	+	*	+	+	=	+	+	=
P-Blocks	$\mu \pm \sigma$	0.0882 ± 0.00247	0.077 ± 0.00347	0.0661 ± 0.00319	0.0876 ± 0.00285	0.0757 ± 0.00437	0.0653 ± 0.00310	0.100 ± 0.00331	0.0897 ± 0.00576	0.0834 ± 0.00636
	Symbol	+	+	=	+	+	*	+	+	+

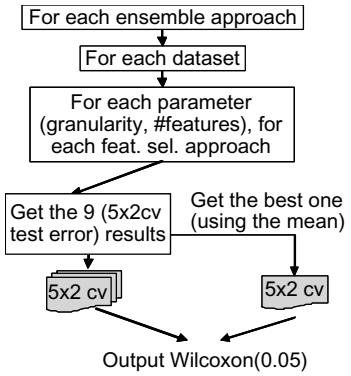

Fig. 2. Experimental design for the statistical test to compare the FRBMCS design methodologies.

bination random subspace + 7 labels obtained the best significant result (over all the other approaches) for the vehicle and the yeast datasets.

5.3. Genetic FRBMCS selection using the TEFF

The values for the genetically selected FRBMCSs using the TEFF are collected in Table 5. Table 5 shows the results obtained with the TEFF and the greedy and the random subspace feature selection methods. Table 5 shows the results obtained with the TEFF and the GRASP 0.50 feature selection method.

The first conclusion we can draw is that the TEFF was able to reduce the best test error for all the problems in comparison with the use of a single classifier. The best improvement was obtained on the sonar dataset (-17%). TEFF is better than a single classifier in all the cases (see Table 2). Comparing the individual test errors between the single classifiers and the GA selection using the TEFF, the best improvement of test error (-33%, with 19x more rules) was obtained with the heart dataset, using the random subspace approach with 3 labels and 5 attributes, proving that randomness is very useful for the improvement of the error. It seems there is a direct relation between the randomness injected in the feature selection method and the amount of improvement of the test error observed between a single classifier and an ensemble selected by the TEFF. In average, over all datasets and all granularities values, the random subspace causes a decrease of a 11% of the test error, while the GRASP 0.50 causes a decrease of a 7% and the greedy approach of only a 5%. Concerning the number of rules, the TEFF produces 10.4x more rules than a single classifier in average, but produces 4.2x less rules than the initial FRBMCSs.

Moreover, it reduces the best test error obtained with the initial FRBMCSs with 50 component classifiers for many datasets, including those with the highest dimension (glass, sonar, yeast, phoneme, p-blocks). The best improvement of the best test error compared to the initial FRBMCSs was obtained on the glass dataset (-9%).

5.4. Genetic FRBMCS selection using the OOBEFF

The values for the genetically selected FRBMCSs using the OOBEFF are collected in Table 6 shows the results obtained with the OOBEFF and the greedy and the random subspace feature selection methods. Table 6 shows the results obtained with the OOBEFF and the GRASP 0.50 feature selection method.

The FRBMCSs based on the OOBEFF are better than the single FRBCSs in seven cases, and slightly worse in the other two cases (glass and yeast). Comparing the individual test errors between the single classifiers and the GA selection using the OOBEFF, the best improvement of test error (-36%, with 24x more rules) was obtained with the breast dataset, using the random subspace approach with 5 labels and 5 attributes. When comparing with the same result using the TEFF, it seems the random subspace allows the best increase for both fitness functions, proving again that a feature selection method based on randomness is very useful for the improvement of the performance. In average, over all datasets and all granularities values, the random subspace causes a decrease of a 12% of the test error, while the GRASP 0.50 causes a decrease of a 6% and the greedy approach of only a 3%.

Concerning the number of rules, the OOBEFF produces 15.9x more rules than a single classifier in average (so 54% more than the number obtained for the TEFF function), but produces only 2.7x less rules than a FRBMCS.

It reduces the best test error obtained with the initial FRBMCSs with 50 component classifiers in five cases (glass, vehicle, sonar, phoneme and p-blocks), and the performance is equal in one more case (yeast). The best improvement of the best test error compared to the FRBMCS was obtained on the sonar dataset (-4%).

5.5. Comparison of the TEFF and the OOBEFF genetic OCS FRBMCS strategies

Comparing the two fitness functions, the OOBEFF is able to outperform the TEFF in the individual test error for 26 cases, mainly for some configurations applied

Table 5
Results for the FRBCS ensembles selected by the GA using the TEFF

		Bagging + Greedy								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.257	**0.360**	0.461	**0.235**	**0.047**	**0.185**	0.498	0.286	0.084
3 labels	#classifiers	4.1	7.3	10.3	12.3	3.2	11.8	6.9	5.9	2.7
5 #attr.	#rules	696.5	904.3	1431.0	1842.1	714.3	964.7	1406.0	1411.4	303.2
	avg. #rules	171.5	125.4	138.3	148.3	221.8	82.2	203.4	239.4	112.7
	time	94.06	26.35	103.26	25.32	83.90	32.86	184.60	656.29	648.74
	5 × 2-cv	**0.242**	0.383	0.392	0.247	0.063	0.214	0.476	0.205	0.075
5 labels	#classifiers	11.5	15.9	15.5	10.4	5.4	16.6	12.7	6.6	4.8
5 #attr.	#rules	6744.9	4233.1	7338.4	5757.7	4795.2	2809.5	10513.1	7162.0	1532.0
	avg. #rules	592.8	268.7	481.9	567.0	898.7	162.2	832.7	1088.4	330.2
	time	93.48	26.10	103.48	25.17	84.57	32.45	182.64	663.86	643.33
	5 × 2-cv	0.258	0.393	**0.374**	0.258	0.156	0.250	**0.446**	**0.180**	**0.064**
7 labels	#classifiers	12.7	8.9	14.6	6.3	20.9	16.3	17.0	10.6	6.0
5 #attr.	#rules	16614.3	3524.3	16102.3	6427.0	26455.1	3716.1	29091.4	30490.3	3949.9
	avg. #rules	1313.9	404.5	1115.7	1040.9	1256.1	227.3	1715.8	2872.8	655.6
	time	92.87	26.50	102.90	24.85	84.00	32.48	186.18	647.24	656.13

		Bagging + Random Subspace								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	**0.256**	**0.381**	0.428	**0.216**	**0.042**	**0.170**	0.505	0.286	0.080
3 labels	#classifiers	4.2	13.7	13.4	20.1	6.4	15.7	5.7	5.9	4.9
5 #attr.	#rules	703.4	1546.0	2239.5	3376.7	1352.7	1260.3	546.1	1411.4	597.0
	avg. #rules	168.1	113.1	168.9	168.3	209.7	80.1	99.1	239.4	124.6
	time	92.77	26.39	103.24	25.08	84.81	32.63	182.90	657.48	650.83
	5 × 2-cv	0.263	0.392	0.378	0.249	0.063	0.204	0.483	0.205	0.074
5 labels	#classifiers	11.9	13.7	13.0	9.4	5.5	13.5	14.1	6.6	7.4
5 #attr.	#rules	6680.0	3312.2	9455.9	6208.8	4690.0	2341.9	4225.5	7162.0	2682.9
	avg. #rules	555.8	245.0	734.3	668.8	856.8	177.1	301.3	1088.4	385.7
	time	91.47	26.18	104.81	24.83	85.60	32.94	183.64	658.75	645.97
	5 × 2-cv	0.265	0.393	**0.337**	0.267	0.187	0.250	**0.441**	**0.180**	**0.065**
7 labels	#classifiers	17.0	15.5	17.5	6.4	10.2	10.5	21.5	10.6	5.4
5 #attr.	#rules	21289.5	5980.6	28854.2	7655.2	11141.2	2902.1	12849.9	30490.3	4253.2
	avg. #rules	1248.4	386.2	1680.2	1203.7	1092.8	279.1	602.2	2872.8	840.2
	time	92.31	26.08	103.52	25.19	84.10	32.78	184.77	639.66	649.01

		Bagging + GRASP $\tau = 0.50$								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.254	0.372	0.449	**0.237**	**0.047**	**0.183**	0.504	0.286	0.083
3 labels	#classifiers	4.4	10.2	12.9	13.9	6.0	15.0	6.9	5.9	6.7
5 #attr.	#rules	763.0	1317.9	1991.6	2252.6	1330.4	1333.1	1377.5	1411.4	811.0
	avg. #rules	174.3	126.0	155.9	161.7	222.4	89.9	200.0	239.4	124.6
	time	93.37	26.49	102.09	25.18	83.65	32.78	184.07	655.97	647.14
	5 × 2-cv	**0.239**	**0.363**	0.399	0.252	0.061	0.202	0.475	0.205	0.074
5 labels	#classifiers	10.9	14.7	12.0	7.8	5.2	13.1	15.3	6.6	9.8
5 #attr.	#rules	6497.4	3986.7	7227.3	4893.9	4709.5	2448.1	12415.5	7162.0	3332.2
	avg. #rules	593.5	282.0	611.3	630.0	907.1	183.6	815.7	1088.4	383.4
	time	92.58	26.16	103.75	24.86	84.30	32.24	184.11	654.60	648.25
	5 × 2-cv	0.256	0.395	**0.356**	0.257	0.174	0.241	**0.441**	**0.180**	**0.063**
7 labels	#classifiers	16.4	10.3	13.2	6.7	11.8	15.4	18.0	10.6	8.6
5 #attr.	#rules	21836.6	4140.6	18296.2	7767.8	15168.6	4132.7	30697.7	30490.3	6033.9
	avg. #rules	1346.2	401.9	1386.5	1148.7	1285.4	276.3	1703.5	2872.8	698.7
	time	92.49	26.18	102.93	25.31	84.24	32.44	187.73	640.89	652.24

Table 6
Results for the FRBCS ensembles selected by the GA using the OOBEFF

		Bagging + Greedy								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.255	0.417	0.497	**0.239**	**0.047**	**0.187**	0.530	0.286	0.087
3 labels	#classifiers	11.5	12.3	14.2	14.7	12.9	12.0	10.8	10.2	9.3
5 #attr.	#rules	1996.8	1516.8	1952.4	2172.6	2856.8	1014.7	2174.4	2431.1	1036.4
	avg. #rules	172.9	123.2	137.7	145.4	221.0	84.2	202.1	238.2	110.1
	time	103.61	29.12	116.53	28.22	94.52	36.47	205.65	732.06	722.17
	5 × 2-cv	**0.239**	**0.380**	0.394	0.247	0.059	0.212	0.477	0.207	0.076
5 labels	#classifiers	15.6	12.2	18.7	16.9	19.5	16.8	17.1	15.0	12.9
5 #attr.	#rules	9121.5	3186.1	8382.3	9305.7	16850.9	2636.3	14160.0	16311.2	4117.0
	avg. #rules	589.7	262.9	449.7	550.2	863.5	160.1	827.1	1089.1	318.2
	time	103.56	29.36	116.06	28.08	94.61	36.08	206.10	728.76	722.00
	5 × 2-cv	0.252	0.417	**0.365**	0.262	0.158	0.258	**0.446**	**0.180**	**0.065**
7 labels	#classifiers	18.3	12.6	21.7	18.0	17.9	15.8	18.9	17.7	15.7
5 #attr.	#rules	23663.2	4616.8	21619.7	18010.3	22124.3	3491.5	32317.7	51100.6	9992.5
	avg. #rules	1287.5	364.6	969.4	998.4	1234.5	217.5	1707.8	2879.4	639.3
	time	104.59	29.37	114.82	28.02	94.26	36.45	204.82	716.90	722.82

		Bagging + Random Subspace								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.264	0.401	0.448	0.236	**0.036**	**0.174**	0.551	0.286	0.086
3 labels	#classifiers	11.6	14.9	19.5	19.5	16.1	20.2	14.0	10.2	10.7
5 #attr.	#rules	1843.1	1653.0	3243.9	3196.2	3363.9	1614.6	1402.4	2431.1	1237.9
	avg. #rules	159.5	111.4	166.6	163.8	209.8	80.3	101.1	238.2	115.4
	time	104.26	29.42	114.90	28.15	93.50	36.51	204.74	730.77	722.48
	5 × 2-cv	**0.252**	0.403	0.374	**0.211**	0.053	0.204	0.493	0.207	0.078
5 labels	#classifiers	15.1	14.2	23.2	26.2	24.5	20.1	27.3	15.0	13.9
5 #attr.	#rules	8327.6	3410.0	14449.0	16535.2	19118.5	3161.8	8139.7	16311.2	4686.5
	avg. #rules	553.4	239.1	625.8	632.0	786.4	154.2	297.2	1089.1	341.3
	time	104.13	29.59	116.17	27.95	94.83	36.65	203.66	731.34	722.34
	5 × 2-cv	0.263	**0.380**	**0.329**	0.238	0.192	0.230	**0.444**	**0.180**	**0.069**
7 labels	#classifiers	19.2	14.9	26.6	23.9	16.1	20.7	33.8	17.7	15.7
5 #attr.	#rules	22950.1	5568.1	36149.9	27271.6	17731.8	4677.2	16597.5	51100.6	10227.8
	avg. #rules	1199.9	381.2	1356.3	1142.8	1097.0	225.4	492.7	2879.4	658.8
	time	103.51	29.43	116.30	28.28	94.73	36.28	206.29	731.03	726.54

		Bagging + GRASP $\tau = 0.50$								
		Pima	Glass	Vehicle	Sonar	Breast	Heart	Yeast	Phoneme	P-Blocks
	5 × 2-cv	0.260	0.419	0.474	**0.226**	**0.045**	**0.185**	0.528	0.286	0.086
3 labels	#classifiers	9.9	13.8	16.7	18.7	15.7	15.4	13.3	10.2	9.1
5 #attr.	#rules	1702.0	1736.3	2496.4	3000.3	3520.8	1360.4	2665.5	2431.1	1070.2
	avg. #rules	171.6	124.1	148.2	160.9	224.3	87.7	199.9	238.2	115.8
	time	104.61	29.55	114.91	28.05	94.02	36.20	205.21	731.10	725.43
	5 × 2-cv	**0.237**	**0.388**	0.386	0.233	0.056	0.215	0.481	0.207	0.076
5 labels	#classifiers	16.0	12.7	19.4	19.3	21.2	16.8	16.4	15.0	15.2
5 #attr.	#rules	9514.2	3414.0	9924.2	11700.3	19118.3	3004.7	13242.0	16311.2	5213.4
	avg. #rules	592.0	268.1	508.5	603.0	902.1	178.8	802.0	1089.1	343.2
	time	103.22	29.26	115.06	27.99	94.65	35.94	203.20	724.91	719.22
	5 × 2-cv	0.253	0.435	**0.348**	0.235	0.179	0.241	**0.445**	**0.180**	**0.064**
7 labels	#classifiers	16.5	13.3	23.4	22.8	16.0	18.7	18.5	17.7	14.8
5 #attr.	#rules	21592.2	5114.6	25346.8	24704.1	20646.1	4475.4	31183.4	51100.6	10088.3
	avg. #rules	1318.6	377.7	1077.7	1088.2	1290.1	237.9	1684.9	2879.4	685.3
	time	103.88	29.00	115.02	28.54	93.57	36.89	204.52	717.69	726.78

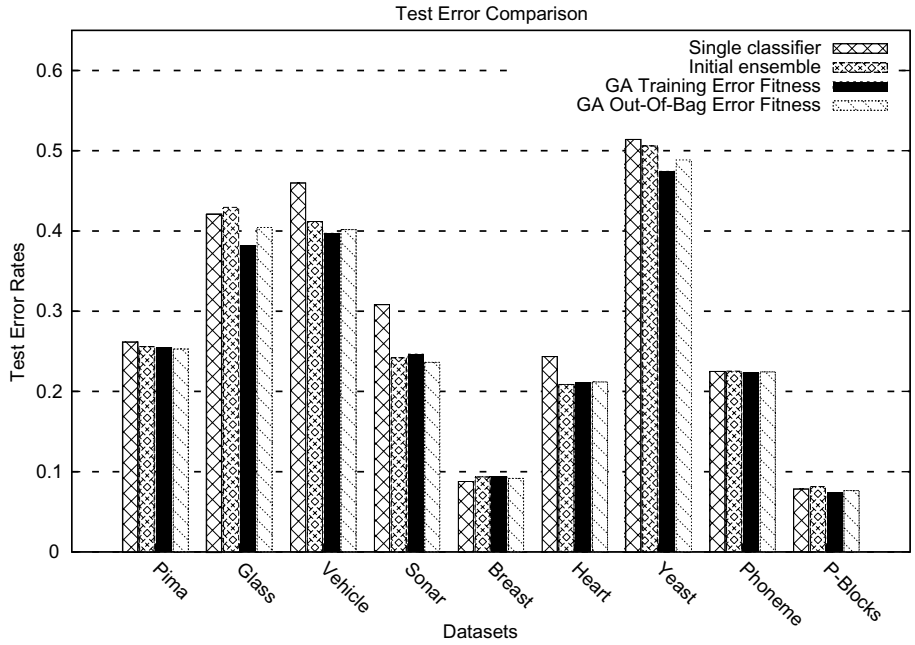

Fig. 3. Comparison of the average test errors of a single classifier, the initial FRBMCS, and those generated using the TEFF and the OOBEFF-based genetic selection.

on the smallest datasets (pima, glass, vehicle, sonar, breast, and heart). The best individual improvement observed was on the breast dataset (-16%), with random subspace (again!) with 5 labels and 5 attributes. We have compared the average improvement over the feature selection approaches, over all the datasets, and only the random subspace shows an improvement with the OOBEFF (-0.52%), while an increase of 0.86% with GRASP 0.50 and of 1.42% with the greedy approach are found. Comparing the number of labels, regardless the datasets and the feature selection approaches, it seems that the best test error improvement was obtained with 5 labels (-1.2%, $+3.1\%$ with 3 labels and -0.1% with 7 labels). For all the remaining cases, it seems the OOBEFF is a little worst than the TEFF, but the results are still better than those obtained by the initial pool. This little decrease in the classification accuracy could be explained by the fact the GA is using less instances (in general, the bootstrapping produce a 37% of instances in the "Out-Of-Bags", this means 63% less instances in average).

Figure 3 shows a comparison between the average test error (taking all the experiments we did with all the parameter settings into account) obtained on the initial FRBMCS of 50 classifiers, and the FRBMCSs derived by the GA using the TEFF and the OOBEFF. As can be seen, the test error obtained by a genetic selection is better in eight cases (pima, glass, vehicle, sonar, breast, yeast, phoneme, and p-blocks). The OOBEFF only outperforms the TEFF in three cases (pima, sonar and breast), corresponding to those cases in which the TEFF was already able to outperform the initial ensemble. In average, the accuracy of the initial ensemble is improved around a 3.2% by the TEFF and around a 2.5% by the OOBEFF.

Looking at the ensemble size, the two fitness functions perform properly. For the TEFF, in general, the number of selected classifiers is very small (10.7 in average, 21.5 for yeast), while keeping the same order of accuracy than the corresponding full 50 FRBCS ensembles. For the OOBEFF, the results are a bit worst (16.7 in average, 33.8 for yeast): it produces a slightly higher number of classifiers, which makes sense considering the fact that the selection is based on non-seen instances. The increase is highly variable depending on the datasets: ranging from $+10\%$ for glass to $+114\%$ for breast. There are only some (six) cases in which the OOBEFF produced smaller ensembles: glass for 5 labels (greedy and GRASP 0.50) and 7 labels (random subspace); sonar for 3 labels (random subspace); and breast and heart for 7 labels (greedy). Thus, the TEFF achieved a good accuracy-complexity trade-off in almost all datasets, but the OOBEFF could be interesting in some cases to improve the accuracy while decreasing

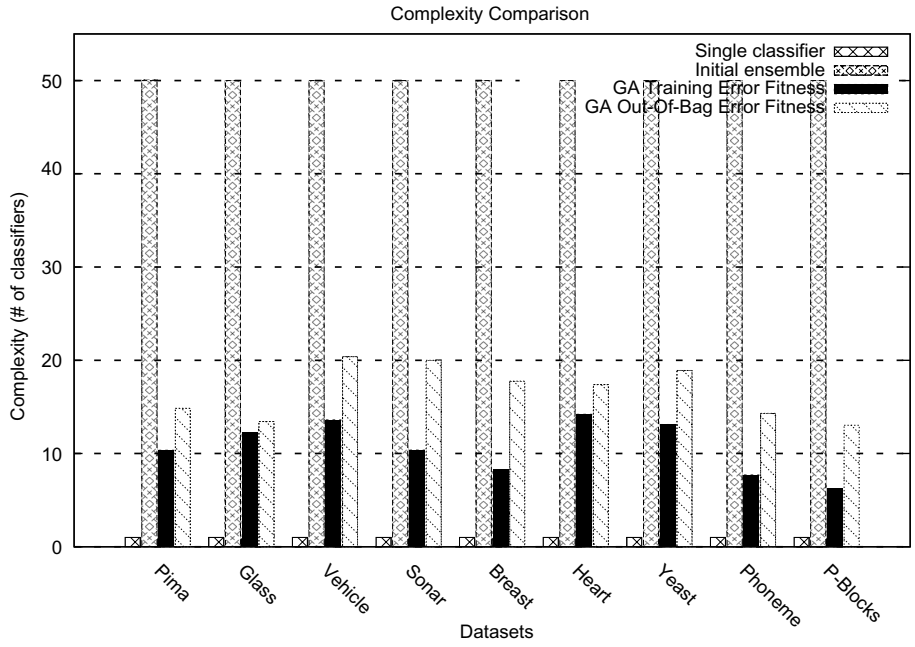

Fig. 4. Comparison of the average complexity (# of classifiers) between the selected FRBMCSs using the TEFF and those generated using the OOBEFF-based genetic selection.

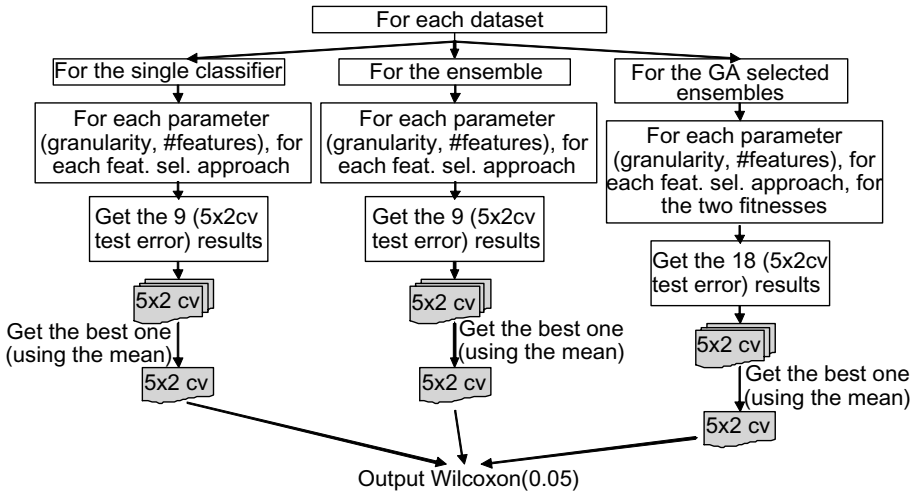

Fig. 5. Experimental design for the statistical test to compare the selected FRBMCS design methodologies.

the complexity (e.g., on glass, 5 labels, 5 attributes and the greedy approach, the ensemble is reduced by a 23% while still decreasing the test error).

Finally, Fig. 4 shows a comparison between the average complexity (computed as the number of existing classifiers in the different FRBMCSs) obtained from the initial ensembles processed by the GA using the TEFF and the OOBEFF. As a reference, the complexity of a single classifier (1) and of the full ensemble (50) are also represented. As can be seen, the TEFF leads to smaller ensembles, and the highest decrease is observed in the p-blocks dataset (-87% with the TEFF, -74% with the OOBEFF). This could be explained by the fact that to reduce overfitting on the non-seen instances, the GA has to include more classifiers in the ensembles. In average, the increase of the size between the two fitness functions is about 63%, but the size of the ensembles obtained with the OOBEFF is still 67%

Table 7
Statistical test for the comparison of the FRBMCSs versus the GA selected FRBMCSs methodologies (see Fig. 5). For each dataset, the best result is marked ('*') and the others are compared to it

		Best single classifier (app./labels)	Best ensemble (app./labels)	Best ens. selected (app./labels/fitness)
Pima	Approach	GRASP/5	GRASP/5	GRASP/5/OOBEFF
	$\mu \pm \sigma$	0.246 ± 0.00991	0.234 ± 0.019	0.237 ± 0.0134
	Symbol	=	*	=
Glass	Approach	GRASP/5	Greedy/5	Greedy/3/TEFF
	$\mu \pm \sigma$	0.375 ± 0.0526	0.396 ± 0.0568	0.360 ± 0.0507
	Symbol	=	=	*
Vehicle	Approach	GRASP/7	Random/7	Random/7/OOBEFF
	$\mu \pm \sigma$	0.399 ± 0.0262	0.330 ± 0.0179	0.329 ± 0.0241
	Symbol	+	=	*
Sonar	Approach	Greedy/3	GRASP/5	Random/5/OOBEFF
	$\mu \pm \sigma$	0.261 ± 0.0463	0.220 ± 0.0445	0.212 ± 0.0413
	Symbol	+	=	*
Breast	Approach	GRASP/3	Random/3	Random/3/OOBEFF
	$\mu \pm \sigma$	0.0466 ± 0.0076	0.0355 ± 0.00468	0.0360 ± 0.0054
	Symbol	+	*	=
Heart	Approach	Greedy/3	Random/3	Random/3/TEFF
	$\mu \pm \sigma$	0.197 ± 0.0284	0.166 ± 0.0335	0.170 ± 0.0360
	Symbol	=	*	=
Yeast	Approach	Greedy/7	Random/7	GRASP/7/TEFF
	$\mu \pm \sigma$	0.442 ± 0.0123	0.444 ± 0.0124	0.441 ± 0.0151
	Symbol	=	=	*
Phoneme	Approach	Greedy/7	Greedy/7	Greedy/7/TEFF
	$\mu \pm \sigma$	0.181 ± 0.00944	0.182 ± 0.00933	0.180 ± 0.00939
	Symbol	=	=	*
P-Blocks	Approach	GRASP/7	GRASP/7	GRASP/7/TEFF
	$\mu \pm \sigma$	0.0648 ± 0.00404	0.0653 ± 0.00310	0.0634 ± 0.00318
	Symbol	=	=	*

smaller than the initial ensemble (79% smaller using the TEFF).

Thus, both fitness functions could be viewed as a proper way to improve the results obtained by the initial ensemble while reducing its complexity, with the TEFF giving better results.

5.6. Statistical significance of the results

Table 7 shows the results of the statistical tests performed to check if the performance of the initial FRBMCSs and the performance of the GA selected FRBMCSs outperform significantly the performance of the single classifier. The best result for each dataset is marked with a star '*'. The experimental design is shown in Fig. 5.

The best results (in average) are always obtained by the initial or the selected ensembles, even if they are only significant for three datasets (on vehicle and sonar, the performance of the GA outperforms significantly the single classifier; on breast the performance of the initial ensemble outperforms significantly the single classifier). In general, the best results are obtained when considering the GA selection (the GA got the best results for 6 datasets, versus only 3 for the initial ensemble).

Thus, combining bagging and the GA selection pro-

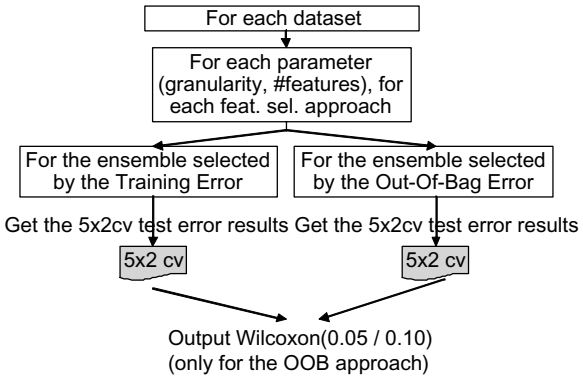

Fig. 6. Experimental design for the statistical test to compare the two fitness functions.

Table 8
Statistical test for the comparison of the two GA fitness functions (see Fig. 6). For each dataset, the OOBEFF is compared to the TEFF for each approach ('+' means the OOBEFF is significantly better). The results are shown with a confidence of 5 and 10%. Only the datasets/approaches with have statistically significant differences are listed

Approach	Greedy	GRASP $\tau = 0.50$		Random Subspace		
Datasets	3 labels	3 labels	7 labels	3 labels	5 labels	7 labels
Glass	–	=/–	=/–			
Vehicle	=/–					
Sonar					+	
Breast					=/+	
Yeast	–	–		–		
P-Blocks	=/–			–	–	–

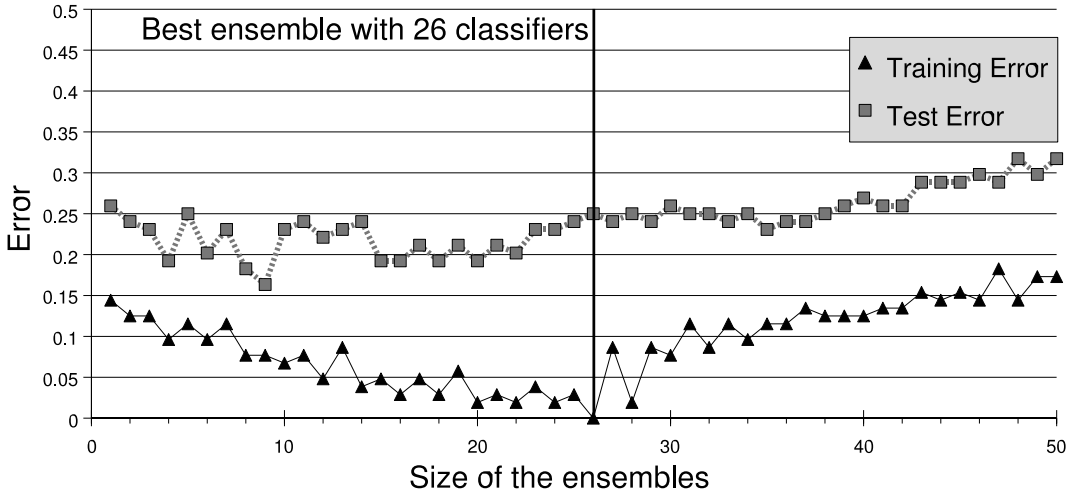

Fig. 7. An example of the training and test errors for the 50 FRBMCSs selected by a chromosome

cess to design FRBMCSs performs better for high dimensional problems with a large number of attributes, producing a smaller rule base while reducing the test errors in some cases, which was our original goal [13]. When combining these two techniques with an advanced feature selection process we also get an improvement on the accuracy for datasets with a higher dimension (glass, vehicle, sonar, and especially, yeast, phoneme and p-blocks, see Table 7).

Table 8 shows the results of the statistical test performed to check the significance of the performance comparison of the TEFF and the OOBEFF for all the approaches and all the datasets. Only the datasets/approaches combinations giving significant results are shown. In this table, the symbols correspond to a confidence of 5% and 10%, respectively. The experimental design is shown in Fig. 6. As already said, in general, the TEFF performs significantly better for six approaches considered in four datasets, while the OOBEFF performs significantly better only for one approach (5 labels + random subspace) in two datasets. Thus, in most of the cases, the two fitness functions give equal performance, apart from some cases (only 13 cases in comparison to the 81 cases considered), in which the OOBEFF performs slightly worse.

6. On the different FRBCS ensembles contained in the best chromosome

For the readability of the paper, we will only show an example of the multicriteria selection capability. In Fig. 7, a graphical representation of the training and test error trends of all the FRBMCSs encoded in the best chromosome obtained from the TEFF-based genetic selection when applied on the initial FRBCS ensemble for the sonar dataset (bagging+random subspace, 3 labels, 5 attributes) are shown.

The chosen solution (the one with the lowest Training Error $TE = 0$, with 26 classifiers) is highlighted.

Notice that the ensemble of 9 classifiers has a better test error and is actually smaller; and how bigger ensembles lead to bigger training and test errors.

We leave for future works the study of this capability of our algorithm and the analysis of its interrelation with the two fitness functions.

7. Conclusions and future works

We have proposed the use of bagging and feature selection approaches like random subspace and greedy and GRASP-based Battiti's methods, together with a TEFF and a OOBEFF-guided multicriteria GA, to design FRBMCS ensembles with a good accuracy-complexity trade-off. The resulting FRBCS ensembles have shown to be able to deal with classification problems with a large number of features (up to 60) and a large number of instances (up to 5,400). The results obtained in some popular data sets of high dimension are quite promising.

Our future work will be concentrated on the study of the influence of other parameters (the GA parameters for instance), on the design of more advanced genetic MCS selection techniques (for example, the use of Pareto-based algorithms), on the use of more advanced fuzzy reasoning mechanisms both in the component FRBCSs and in the ensemble, on the analysis of the multicriteria GA potentials, and on the design of MCSs of more accurate FRBCSs.

References

[1] J.J. Aguilera, M. Chica, M.J. del Jesus and F. Herrera, Niching genetic feature selection algorithms applied to the design of fuzzy rule based classification systems, *In IEEE International Conference on Fuzzy Systems* (*FUZZ-IEEE*), pages 1794–1799, London, 2007.

[2] R.E. Banfield, L.O. Hall, K.W. Bowyer and W.P. Kegelmeyer, A comparison of decision tree ensemble creation techniques, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(1) (2007), 173–180.

[3] R. Battiti, Using mutual information for selecting features in supervised neural net learning, *IEEE Transactions on Neural Networks* **5**(4) (1994), 537–550.

[4] L. Breiman, Bagging predictors, *Machine Learning* **24**(2) (1996), 123–140.

[5] L. Breiman, Stacked regressions, *Machine Learning* **24**(1) (1996), 49–64.

[6] L. Breiman, Random forests, *Machine Learning* **45**(1) (2001), 5–32.

[7] J. Canul-Reich, L. Shoemaker and L.O. Hall, *Ensembles of Fuzzy Classifiers*, In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1–6, London, 2007.

[8] J. Casillas, O. Cordón and F. Herrera, COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **32**(4) (2002), 526–537.

[9] C.A. Coello, G.B. Lamont and D.A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, (2nd Edition), Springer, 2007.

[10] O. Cordón, M.J. del Jesus and F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, *International Journal of Approximate Reasoning* **20** (1999), 21–45.

[11] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann and L. Magdalena, Ten years of genetic fuzzy systems: Current framework and new trends, *Fuzzy Sets and Systems* **141**(1) (2004), 5–31.

[12] O. Cordóon, F. Herrera, F. Hoffmann and L. Magdalena, *Genetic Fuzzy Systems*, Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, 2001.

[13] O. Cordón, A. Quirin and L. Sánchez, *A first Study on Bagging Fuzzy Rule-Based Classification Systems with multicriteria genetic selection of the Component Classifiers*, In IEEE International Workshop on Genetic and Evolving Fuzzy Systems (GEFS), pages 11–16, Germany, 2008.

[14] O. Cordón, A. Quirin and L. Sánchez, *On the Use of Bagging, Mutual Information-Based Feature Selection and Multicriteria Genetic Algorithms to Design Fuzzy Rule-Based Classification Ensembles*, In International Conference on Hybrid Intelligent Systems (HIS), pages 549–554, Barcelona, 2008.

[15] B.V. Dasarathy and B.V. Sheela, A composite classifier system design: Concepts and methodology, *Proceedings of IEEE* **67**(5) (1979), 708–713.

[16] M.J. del Jesus, F. Hoffmann, L.J. Navascues and L. Sánchez, Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms, *IEEE Transactions on Fuzzy Systems* **12**(3) (2004), 296–308.

[17] T.G. Dietterich, Approximate statistical test for comparing supervised classification learning algorithms, *Neural Computation* **10**(7) (1998), 1895–1923.

[18] N. Dimililer, E. Varoglu and H. Altincay, Classifier subset selection for biomedical named entity recognition, *Applied Intelligence* **31**(3) (December 2009), 267–282.

[19] T.A. Feo and M.G.C. Resende, Greedy randomized adaptive search procedures, *Journal of Global Optimization* **6** (1995), 109–133.

[20] B. Gabrys and D. Ruta, Genetic algorithms in classifier fusion, *Applied Soft Computing* **6**(4) (2006), 337–347.

[21] G. Giacinto and F. Roli, Dynamic classifier selection based on multiple classifier behaviour, *Pattern Recognition* **34**(9) (2001), 1879–1881.

[22] S.T. Hadjitodorov and L.I. Kuncheva, Selecting diversifying heuristics for cluster ensembles, *Lecture Notes in Computer Science* **4472** (2007), 200–209.

[23] T. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8) (1998), 832–844.

[24] F. Hoffmann, Combining boosting and evolutionary algorithms for learning of fuzzy classification rules, *Fuzzy Sets and Systems* **141**(1) (2004), 47–58.

[25] H. Ishibuchi, T. Nakashima and M. Nii, *Classification and Modeling With Linguistic Information Granules*, Springer, 2005.

[26] H. Ishibuchi and Y. Nojima, Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classi-

fiers, *International Journal of Hybrid Intelligent Systems* **3**(3) (2006), 129–145.
[27] M. Kim, S. Min and I. Han, An evolutionary approach to the combination of multiple classifiers to predict a stock price index, *Expert Systems with Applications* **31** (2006), 241–247.
[28] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, 2004.
[29] E. Lehmann, *Nonparametric Statistical Methods Based on Ranks*, McGraw-Hill, New-York, 1975.
[30] Y. Liu, X. Yao and T. Higuchi, Evolutionary ensembles with negative correlation learning, *IEEE Transactions on Evolutionary Computation* **4** (2000), 380–387.
[31] G. Martínez-Munoz, D. Hernández-Lobato and A. Suárez, Selection of decision stumps in bagging ensembles, *Lecture Notes in Computer Science* **4668** (2007), 319–328.
[32] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1996.
[33] Y. Nojima and H. Ishibuchi, Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design, *International Journal of Hybrid Intelligent Systems* **4**(3) (2007), 157–169.
[34] L.S. Oliveira, M. Morita, R. Sabourin and F. Bortolozzi, Multi-objective genetic algorithms to create ensemble of classifiers, *Lecture Notes in Computer Science* **3410** (2005), 592–606.
[35] D. Optiz and R. Maclin, Popular ensemble methods: An empirical study, *Journal of Artificial Intelligence Research* **11** (1999), 169–198.
[36] D. Partridge and W.B. Yates, Engineering multiversion neural-net systems, *Neural Computation* **8** (4) (1996), 869–893.
[37] W. Pedrycz, A. Breuer and N.J. Pizzi, Fuzzy adaptive logic networks as hybrid models of quantitative software engineering, *Intelligent Automation and Soft Computing* **12**(2)(2006), 189–209.
[38] W. Pedrycz and K.C. Kwak, Boosting of granular models, *Fuzzy Sets and Systems* **157**(22) (2006), 2934–2953.
[39] D. Ruta and B. Gabrys, Classifier selection for majority voting, *Information Fusion* **6**(1) (2005), 63–81.
[40] L. Sánchez and J. Otero, Boosting fuzzy rules in classification problems under single-winner inference, *International Journal of Intelligent Systems* **22**(9) (2007), 1021–1034.
[41] E.M. Dos Santos, R. Sabourin and P. Maupin, *Single and Multi-Objective Genetic Algorithms for the Selection of Ensemble of Classifiers*, In International Joint Conference on Neural Networks (IJCNN), pages 3070–3077, Vancouver, 2006.
[42] E.M. Dos Santos, R. Sabourin and P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, *Pattern Recognition* **41**(10) (2008), 2993–3009.
[43] R. Schapire, The strength of weak learnability, *Machine Learning* **5**(2) (1990), 197–227.
[44] R. Schapire, Y. Freund, P. Bartlett and W. Lee, *Boosting the Margin: A new Explanation for the Effectiveness of Voting Methods*, In International Conference on Machine Learning, pages 322–330, Nashville, 1997.
[45] R. Scherer, Boosting ensemble of relational neurofuzzy systems, *Lecture Notes in Computer Science* **4029** (2006), 306–313.
[46] C.E. Shannon and W.Weaver, *The Mathematical Theory of Communication*, University of Illlinois Press, 1949.
[47] A.J.C. Sharkey and N.E. Sharkey, *The Test and Select Approach to Ensemble Combination*, In InternationalWorkshop on Multiclassifier Systems, pages 30–44, Cagliari, 2000.
[48] C.-A. Tsai, T.-C. Lee, I.-C. Ho, U.-C. Yang, C.-H. Chen and J.J. Chen, Multi-class clustering and prediction in the analysis of microarray data, *Mathematical Biosciences* **193**(1) (2005), 79–100.
[49] A. Tsymbal, M. Pechenizkiy and P. Cunningham, Diversity in search strategies for ensemble feature selection, *Information Fusion* **6**(1) (2005), 83–98.
[50] S.J. Verzi, G.L. Heileman and M. Georgiopoulos, Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory, *Neural Networks* **19**(4) (2006), 446–468.
[51] L. Xu, A. Krzyzak and C.Y. Suen, Methods of combining multiple classifiers and their application to handwriting recognition, *IEEE Transactions on Systems, Man, and Cybernetics* **22**(3) (1992), 418–435.
[52] Z.H. Zhou, Ensembling local learners through multimodal perturbation, *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics* **35**(4) (2005), 725–735.