International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 19, No. 4 (2011) 589–633
© World Scientific Publishing Company
DOI: 10.1142/S0218488511007155

World Scientific
www.worldscientific.com

ON DESIGNING FUZZY RULE-BASED MULTICLASSIFICATION SYSTEMS BY COMBINING FURIA WITH BAGGING AND FEATURE SELECTION

KRZYSZTOF TRAWIŃSKI*, OSCAR CORDÓN† and ARNAUD QUIRIN‡

European Centre for Soft Computing, Mieres, Spain
Edificio Científico-Tecnológico,
Calle Gonzalo Gutiérrez Quirós S/N, 33600 Mieres, Asturias
* *krzysztof.trawinski@softcomputing.es*
† *oscar.cordon@softcomputing.es*
‡ *arnaud.quirin@softcomputing.es*

Received 18 December 2010
Revised 1 June 2011

In this work, we conduct a study considering a fuzzy rule-based multiclassification system design framework based on Fuzzy Unordered Rule Induction Algorithm (FURIA). This advanced method serves as the fuzzy classification rule learning algorithm to derive the component classifiers considering bagging and feature selection. We develop an exhaustive study on the potential of bagging and feature selection to design a final FURIA-based fuzzy multiclassifier dealing with high dimensional data. Several parameter settings for the global approach are tested when applied to twenty one popular UCI datasets. The results obtained show that FURIA-based fuzzy multiclassifiers outperform the single FURIA classifier and are competitive with C4.5 multiclassifiers and random forests.

Keywords: Multiclassification systems; classifier ensembles; fuzzy rule-based classification systems; fuzzy rule-based multiclassification systems; FURIA; bagging, feature selection; MIFS.

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.[29] 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 recently using different kinds of fuzzy classifiers.[4,9,30,32,37,45,52]

Fuzzy Unordered Rule Induction Algorithm (FURIA)[23,24] is a powerful fuzzy classification rule learning algorithm that can deal with a very common problem of fuzzy rule-based classification systems (FRBCSs), the so-called curse of dimensionality.[26] By combining advantages of the RIPPER algorithm[10] with fuzzy logic, this algorithm is able to generate simple and compact sets of fuzzy classification rules, even when tackling datasets with a large amount of features. Apart from its ability to deal with high dimensional datasets, this approach has shown a performance advantage in comparison to classical machine learning methods such like RIPPER[10] and C4.5.[38]

An individual classifier must provide different patterns of generalization in order to obtain a diverse set of classifiers composing a highly accurate ensemble.[29,51] Otherwise, the ensemble would be composed of the same or similar classifiers and would provide a similar accuracy to the single one. There are several techniques in order to obtain diversity among the classifiers. Bagging[7] and boosting[41] are the two most popular generic approaches to do so.[19] There are also other more recent proposals considering other ways to promote disagreement between the component classifiers, with feature selection being an extended strategy.[20] All in all, it turned out that a combination between bagging and feature selection is a generic approach leading to good MCS designs for any kind of classifier learning method.[34,44]

In this paper we aim to study the performance of FURIA-based fuzzy MCSs, and propose a new framework being able to deal with high dimensional datasets. Our proposal focuses on the combination of a quick FRBCS design method with bagging and a quick feature selection method. We will show how this combination is both efficient, due to its inherent parallelism, and accurate, thanks to the high quality of the base classifier. Several FURIA-based fuzzy MCS composition designs are tested including bagging, feature selection, and the combination of bagging and feature selection. We considered three different types of feature selection algorithms: random subspace,[20] mutual information-based feature selection (MIFS),[3] and the random-greedy feature selection based on MIFS and the GRASP approach,[18] although the methodology is flexible to incorporate any other feature selection approach.

In order to test the accuracy of the proposed fuzzy MCSs, we conduct comprehensive experiments with 21 datasets taken from the UCI machine learning repository and provide a deep study of the results obtained. Finally, our approach is compared against two state-of-the-art MCS algorithms (bagging decion trees[17] and random forests[8]) and also with an application of the fuzzy MCS generation approach[13,14] with other, less powerful fuzzy classifier derivation method.[26]

This paper is structured as follows. The next section presents a state of the art about MCSs and fuzzy MCSs. In Sec. 3 the FURIA algorithm is described, while Sec. 4 recalls our approach for designing FURIA-based fuzzy MCSs. The experiments developed and their analysis are shown in Sec. 5. Finally, Sec. 6 collects some concluding remarks and future research lines.

2. Background and Related Work

This section explores the current literature related to the generation of fuzzy rule-based multiclassification systems (FRBMCSs). The techniques used to generate MCSs and fuzzy MCSs are described in Secs. 2.1 and 2.2, respectively.

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.[29] These kinds of methods have gained a large acceptance in the machine learning community during the last two decades due to their high performance. Decision trees are the most common classifier structure considered and much work has been done in the topic,[2,17] although they can be used with any other type of classifiers (the use of neural networks is also very extended, see for example Ref. 33).

There are different ways to design a classifier ensemble. On the one hand, there is a classical group of approaches considering *data resampling* to obtain different training sets to derive each individual classifier. In *bagging*,[7] they are independently learnt from resampled training sets ("bags"), which are randomly selected with replacement from the original training data set. *Boosting* methods[41] sequentially generate the individual classifiers (weak learners) 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 selection probability to the incorrectly predicted examples by the previous classifiers.

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.[54] Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features.[51,53] *Random subspace*,[20] where each feature subset is randomly generated, is one of the most representative methods of this kind.

Finally, there are some advanced proposals that can be considered as *combinations of the two groups*. The most extended one could be *random forests*,[8] 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[2,33] for two surveys for the case of decision tree (both) and neural network ensembles (the latter), including exhaustive experimental studies.

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:[36,52] as granular models,[37] as neuro-fuzzy systems,[42] as well as single fuzzy rules.[15,21,39]

However, only a few contributions for bagging fuzzy classifiers have been proposed considering, fuzzy adaptive neural networks,[36] fuzzy neural networks (together with feature selection),[46] fuzzy clustering-based classifiers,[50] neuro-fuzzy systems,[9] and fuzzy decision trees[4,30] as component classifier structures.

Especially worth mentioning is the contribution of Bonissone *et al.*[4] This approach hybridizes Breimann's idea of random forests[8] with fuzzy decision trees.[28] Such resulting fuzzy random forest combines characteristics of MCSs with randomness and fuzzy logic in order to obtain a high quality system joining robustness, diversity, and flexibility to not only deal with traditional classification problems but also with imperfect and noisy datasets. The results show that this approach obtains good performance in terms of accuracy for all the latter problem kinds.

In our previous studies,[12,13,48,49] we proposed a MCS methodology based on classical MCS design techniques such as bagging and feature selection with a fuzzy rule-based classification system (FRBCS) as a base classifier. The fuzzy classification rule learning algorithm considered was the basic heuristic method proposed by Ishibuchi.[26] A multicriteria genetic algorithm (GA) was used for a static component classifier selection from FRBMCSs guided by several fitness functions based on training error and likelihood, as well as bicriteria fitness functions based on training error and likelihood or diversity measures.

Some other contributions based on the use of GAs should also be remarked. On the one hand, an FRBCS ensemble design technique is proposed in Ref. 1 considering some niching GA-based feature selection methods to generate the diverse component classifiers, and another GA for classifier fusion by learning the combination weights. On the other hand, another interval and fuzzy rule-based ensemble design method using a single- and multiobjective genetic selection process is introduced in.[31,32] In this case, the coding scheme allows an initial set of either interval 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 Ref. 27, where an EMO algorithm generated a Pareto set of FRBCSs with different accuracy-complexity trade-offs to be combined into an ensemble.

3. FURIA

Fuzzy Unordered Rules Induction Algorithm (FURIA)[23,24] is an extension of the state-of-the-art rule learning algorithm called RIPPER,[10] having its advantages such like simple and comprehensible fuzzy rule base, and introducing new features. FURIA provides three different extensions of RIPPER: i) it takes an advantage of fuzzy rules instead of crisp ones, ii) it applies unordered rule sets instead of rule lists, and iii) it proposes a novel rule stretching method in order to manage uncovered examples. Below the said features of FURIA are reviewed.

3.1. *Unordered rule base instead of the list of rules*

The first extension of FURIA is the following. It deals with a standard unordered rule base (RB) instead of a decision list, as the latter provides one crucial disadvantage. Particularly, a list of rules favors a default class, that introduces a bias. Here, for each class, a set of rules is generated using the one-vs.-rest strategy. Thus, FURIA separates each class from the other classes. In consequence, there is no default rule and the order of the rules is not important.

However, this new approach has two drawbacks. The first one concerns a conflict which arises when having the same coverage of several rules from different classes. The second one may take place when an example is not covered by any of the rules. The first drawback is rather unlikely to occur, even though in case it occurs, it may be resolved easily. The latter issue is solved by introducing a novel rule stretching method as described below.

3.2. *Fuzzification of the RIPPER rules*

The fuzzification of the RIPPER (crisp) rules corresponds to the transformation of the crisp values into the fuzzy ones, that is fuzzy sets with trapezoidal membership functions. Based on the training set the best fuzzy interval is generated. Considering the intervals of the crisp rules I_i as the cores $[b_i, c_i]$ of the fuzzy rule, a learning process aims at determining the optimal size of the supports of each of the antecedents $[a_i, d_i]$. It must be pointed that only the subset D_T^i of the training set D_T that have not been already covered by any of the antecedents ($A_j \in FI_j$, $j \neq i$) is considered in order to build a single antecedent ($A_i \in I_i$):

$$D_T^i = \{x = (x_1 \cdots x_k) \in D_T | FI_j(x_j) > 0 \text{for all } j \neq i\} \subseteq D_T \tag{1}$$

Then, the D_T^i is divided into two subsets, the positive subset D_{T+}^i and the negative subset D_{T-}^i. The following measure, called rule purity, is used in order to check the quality of the fuzzification:

$$pur = \frac{p_i}{p_i + n_i} \tag{2}$$

where

$$p_i = \sum_{x \in D_{T+}^i} \mu_{A_i}(x) \; ; \; n_i = \sum_{x \in D_{T-}^i} \mu_{A_i}(x)$$

The rule fuzzification procedure is greedy and it iterates over all antecedents calculating the best fuzzification in terms of purity (see Eq. (2)). The candidate values for a are those values laying on the left side from b belonging to D_T^i, and are expressed as: $x_i | x = (x_1, \ldots, x_k) \in D_T^i, x_i < b$. The candidate values for d are those values laying on the right side from c belonging to D_T^i, and are expressed as: $x_i | x = (x_1, \ldots, x_k) \in D_T^i, x_i > c$. In case of a tie, the larger fuzzy set, the one having a larger distance from the core, is selected. Then, the antecedent with the highest purity value is selected to be fuzzified. The whole process ends up when all antecedents are fuzzified. This procedure is repeated only once, as it has been noticed that in almost all cases convergence is obtained after the first iteration.

3.3. *Fuzzy classification rule structure and fuzzy reasoning method*

Fuzzy rules of FURIA are composed of a class C_j and a certainty degree CD_j in the consequent, the most extended fuzzy classification rule structure.[11,26] The final form of a rule is the following:

$$\begin{aligned} &R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{and } x_n \text{ is } A_{jn} \\ &\text{then Class } C_j \text{ with } CD_j;\;\; j = 1, 2, \ldots, N\,. \end{aligned}$$

The certainty degree of a given example x is defined as follows:

$$CD_j = \frac{2\frac{|D_T^{C_j}|}{|D_T|} + \sum_{x\in D_T^{C_j}} \mu_r^{C_j}(x)}{2 + \sum_{x\in D_T} \mu_r^{C_j}(x)} \tag{3}$$

where $D_T^{C_j}$ stands for a subset of the training set in which the instances are affected to the class C_j. The fuzzy reasoning method used is the so-called voting-based method.[11,25] In this approach, each fuzzy rule makes a vote for its consequent class. The vote strength of the rule is calculated as the product of the firing degree $\mu_r^{C_j}(x)$ and the certainty degree CD_j. The final decision given as the output is the class with the largest value of the accumulated vote, which is calculated as follows:

$$V_h = \sum_{\substack{R_j\in \mathrm{RB}\\ C_j=h}} \mu_r^{C_j}(x) * CD_j \tag{4}$$

where h is the class for which the accumulated vote is computed. In this approach, all compatible fuzzy rules are responsible for the classification, which should provide a higher robustness. It must be pointed that when there is no rule of any class covering a given example x, a rule stretching procedure, explained in Sec. 3.4, is executed.

3.4. *Rule stretching*

In case some examples of the training dataset not covered by any rule exist, a procedure, called rule stretching or rule generalisation, is applied. This algorithm enlarges the covering surface of the rules by deleting at least one antecedent from each of the rules. The generalization procedure aims to reach a minimal state i.e. only the minimal amount of antecedents are removed. In FURIA, rule stretching treats antecedents in the same order in which they were learned. Thus, it introduces implicitly a degree of importance among the antecedents, which decreases the complexity of the approach. The final list is then obtained by cutting the entire antecedents list at the point where an antecedent not satisfying a given example is encountered. To check that general rules are obtained, the following measure is used:

$$\frac{p+1}{p+n+2} \times \frac{k+1}{m+2}$$

where p and n are respectively the number of positive and negative examples covered by the rule, while m is the size of the entire antecedents list and k is the size of the generalized list. Note that the second part of the measure aims at discarding heavily pruned rules, as pruning is rather decreasing the relevance of the rule.

The interested reader is referred to[23,24] for more details regarding the description of FURIA and its improvements with respect to the RIPPER algorithm.

4. Bagging FURIA-Based Fuzzy MCSs

In this section we will detail how the FURIA fuzzy MCSs are designed. 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 procedures in order to provide individual training sets (the so-called *bags*) to train FURIA classifiers. After the training, we get a FURIA-based fuzzy MCS, which is validated using the training and the test errors, as well as a measure of complexity based on the total number of component classifiers obtained from FURIA. The whole procedure is graphically presented in Fig. 1.

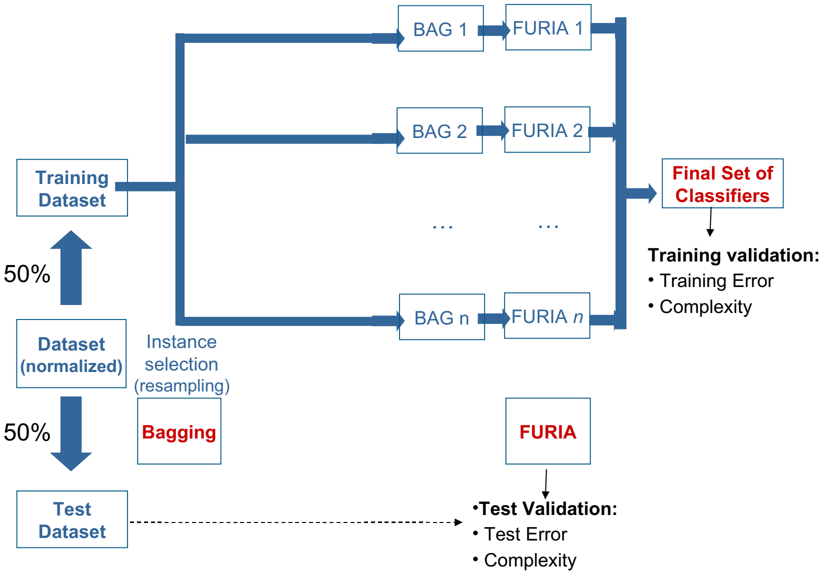

Fig. 1. Our framework: after the instance and the feature selection procedures, the component fuzzy classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based combination method.

4.1. *FURIA-based fuzzy MCS design approaches*

In Refs. 34 and 44, it was shown that a combination between bagging and feature selection composed a general design procedure which usually leads to good MCS designs, regardless the classifier structure considered. Hence, we decided to follow that idea and we integrate FURIA into a framework of that kind. We aim to combine the diversity induced by the MCS design methods and the robustness of the FURIA method in order to derive good performance fuzzy rule-based MCSs for high dimensional problems. We also try a combination of FURIA with bagging and feature selection separately in order to analyze which is the best setting for the design of FURIA-based fuzzy MCSs.

The term *bagging* is an acronym of bootstrap aggregation and refers to the first successful method proposed to generate MCSs.[7] This approach was originally designed for decision tree-based classifiers, however it can applied to any type of model for classification and regression problems. Bagging is based on bootstrap and consists of reducing the variance of the classification by averaging many classifiers that have individually been tuned to random samples that follow the sample distribution of the training set. The final output of the model is the most frequent value, called voting, of the learners considered. Bagging is the most effective when dealing with unstable classifiers, what means a small change in the training set can cause a significant change in the final model. In addition, it is recommended when a given dataset is composed of small amount of examples. Furthermore, bagging enables a parallel and independent learning of the learners in the ensemble.

In this contribution, the bags are generated with the same size as the original training set, as commonly done. Three different feature selection methods, random subspace,[20] mutual information-based feature selection (MIFS),[3] and a random-greedy feature selection method based on MIFS and the GRASP approach,[18] are considered. For each feature selection algorithm three different feature subsets of different sizes, which are based on the initial number of features in the classification problem, are tested.

Random subspace is a method in which a subset of features is randomly selected from the original dataset. Alternatively, the greedy Battiti's MIFS method is based on a forward greedy search using the mutual information measure,[43] with regard to the class. This method orders a given set S of features by the information they bring to classify the output class considering the already selected features. The mutual information $I(C,F)$ for a given feature F is defined as:

$$I(C,F) = \sum_{c,f} P(c,f) \log \frac{P(c,f)}{P(c)P(f)} \tag{5}$$

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

The random-greedy 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 features according to the Q measure at each selection step. 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.

Random search such as random subspace for feature selection is a well-known approach in the multiclassifiers research field.[4,5,8,17,20] Nevertheless, the use of a heuristic such as a randomized variant of greedy Battiti's MIFS[3] combined with FURIA, which is a tree-based fuzzy rule generation approach, may lead to a performance improvement. Note that the greedy Battiti's MIFS leads always to the same subset of features, thus this approach fails to provide MCSs with enough diversity when considered as the only MCS approach, i.e., without being combined with bagging. No matter which its size is, such ensemble will always provide the same result and will be skipped in the experimentation part regarding FURIA-based fuzzy MCSs combined with feature selection.

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

5. Experiments and Analysis of Results

This section presents all the experiments performed. Section 5.1 introduces the experimental setup. In Sec. 5.2 we check the good quality of single FURIA dealing with high dimensional problems with many features. Section 5.3 presents the combination of FURIA-based fuzzy MCSs with bagging, but without feature selection. Section 5.4 is devoted to the construction of FURIA-based fuzzy MCSs combined with feature selection only. Then, Sec. 5.5 shows results of FURIA-based fuzzy MCSs combined with bagging and feature selection. Section 5.6 summarizes all the experiments developed reporting an advantage of our FURIA-based fuzzy MCS with bagging and compares them against some other well established MCS design methodologies such as bagging decision trees, random forests, and Ishibuchi-based fuzzy MCSs, which is based on the same fuzzy MCS design methodology but with a different fuzzy classifier design method.

5.1. *Experimental setup*

To evaluate the performance of the generated FURIA-based fuzzy MCSs, we have selected twenty one datasets with different characteristics concerning the number of examples, features, and classes from the UCI machine learning repository (see Table 1). In order to compare the accuracy of the considered classifiers, we used

Table 1. Datasets considered.

Abbrev.	Dataset	#Examples	#Attr	#Classes
aba	abalone	4178	7	28
bre	breast	700	9	2
gla	glass	214	9	7
hea	heart	270	13	2
ion	ionosphere	352	34	2
let	letter	20000	16	26
mag	magic	19020	10	2
opt	optdigits	5620	64	10
pbl	pblocks	5474	10	5
pen	pendigits	10992	16	10
pho	phoneme	5404	5	2
pim	pima	768	8	2
sat	sat	6436	36	6
seg	segment	2310	19	7
son	sonar	208	60	2
spa	spambase	4602	57	2
tex	texture	5500	40	11
veh	vehicle	846	18	4
wav	waveform	5000	40	3
win	wine	178	13	3
yea	yeast	1484	8	10

Dietterich's 5×2-fold cross-validation (5×2-cv), which is considered to be superior to paired k-fold cross validation in classification problems.[16]

Three different feature subsets of different sizes (called Small, Medium, and Large), which are relative with respect to the initial size of features of the classification problem, are tested for the FURIA-based fuzzy MCSs using feature selection. The considered rule to select a feature subset size is following: if the size of an initial feature set is smaller than 10, then the Small feature subset size is equal to 3, the Medium feature subset size is equal to 4, and the Large feature subset size is equal to 5. If the size of an initial feature set is between 10 and 20, then the Small feature subset size is equal to 5, the Medium feature subset size is equal to 7, the Large feature subset size is equal to 9. Finally, if a size of an initial feature set is larger than 30, then the Small feature subset size is roughly equal to 10% of the initial set, the Medium feature subset size is roughly equal to 20% of the initial set, and the Large feature subset size is roughly equal to 30% of the initial set (see Table 2).

As described in Sec. 4.1, these features are to be selected by means of three different approaches: the greedy Battiti's MIFS filter feature selection method,[3] the Battiti's method with GRASP (with τ equal to 0.5, see Sec. 4.1), and random subspace.[20] Battiti's method has been run by considering a discretization of the real-valued attribute domains in ten parts and setting the β coefficient to 0.1.

The FURIA-based fuzzy MCSs generated are initially comprised by 3, 5, 7, and 10 classifiers in order to evaluate the impact of the ensemble size in the accuracy

Table 2. Feature subset sizes for each of the datasets considered.

Dataset	#Attr.	Small feat. subset size	Medium feat. subset size	Large feat. subset size
abalone	7	3	4	5
breast	9	3	4	5
glass	9	3	4	5
heart	13	5	7	9
ionosphere	34	5	7	9
letter	16	5	7	9
magic	10	5	7	9
optdigits	64	6	12	18
pblocks	10	5	7	9
pendigits	16	5	7	9
phoneme	5	3	4	5
pima	8	3	4	5
sat	36	4	8	12
segment	19	5	7	9
sonar	60	6	12	18
spambase	57	6	12	18
texture	40	4	8	12
vehicle	18	5	7	9
waveform	40	4	8	12
wine	13	5	7	9
yeast	8	3	4	5

of the obtained MCS. A small number of component fuzzy classifiers (up to 10) is considered in this first study. Larger numbers are left for future works as well as the consideration of a classifier selection mechanism.

All the experiments have been run in a cluster at the University of Granada on Intel quadri-core Pentium 2.4 GHz nodes with 2 GBytes of memory, under the Linux operating system.

As there are many different variants and parameter values to be tested, analysis of the obtained results will be performed in parts and following an incremental approach for the sake of comprehensibility.

Despite of accuracy, which is not always believed to be the best choice, more advanced metrics are considered. From a confusion matrix presented in Table 3, which considers independently positive and negative class examples, one can obtain four performance metrics considering positive and negative classes independently:

- True positive rate. It is defined as the percentage of positive examples correctly classified as being of the positive class $TP_r = \frac{TP}{TP+FN}$.
- True negative rate. It is defined as the percentage of negative examples correctly classified as being of the negative class $TN_r = \frac{TN}{FP+TN}$.
- False positive rate. It is defined as the percentage of negative examples incorrectly classified as being of the positive class $FP_r = \frac{FP}{FP+TN}$.
- False negative rate. It is defined as the percentage of positive examples incorrectly classified as being of the negative class $FN_r = \frac{FN}{TP+FN}$

Table 3. Confusion matrix representing the metrics assessing a binary classification problem.

		Prediction	
		Positive Class	Negative Class
Real value	Positive Class	True Positive (TP)	False Negative (FN)
	Negative Class	False Positive (FP)	True Negative (TN)

A well-known method of presenting the performance of classification is the Receiver Operating Characteristic (ROC) curve,[6] showing a trade-off between the benefits (TP_r) and costs (FP_r) of a classifier. From that, the Area Under the ROC Curve (AUC)[22] can be obtained, which summarizes the performance of the classifier. The AUC is calculated as follows:

$$AUC = \frac{1 + TP_r - FP_r}{2} \tag{6}$$

Since we deal with multi-class problems in opposite to what the AUC metric was designed for (in principle, it only serves for binary problems), we use the well-known one-versus-all strategy. In this case, for each class we calculate the AUC treating all the examples belonging to the given class as positive ones and the examples belonging to any other class as the negative ones. In the final results we consider the average AUC value.

We use this metric to perform the final comparison between the best choices of FURIA-based fuzzy MCSs against some other well established MCS design methodologies such as bagging decision trees and random forests, as well as against Ishibuchi-based fuzzy MCSs.

5.2. *Single FURIA-based fuzzy classifier for high dimensional problems*

In the first place, we have conducted experiments on a single FURIA-based fuzzy classifier without feature selection in order to observe its behavior on the different datasets selected. Notice that, some of them can be considered to be high dimensional, either with respect to the number of features or with respect to the number of examples.

We may observe that FURIA in isolation is able to deal with high dimensional datasets with many features (for instance optdigits, which has 64 features) as well as with many examples (for instance letter, which has 20.000 examples), providing good quality results (see Table 4). Our aim in the reminder of this section is to check if the use of fuzzy MCSs based on FURIA allows us to improve the latter capability by obtaining a more accurate classification system.

5.3. *Bagging FURIA-based fuzzy MCSs*

In this subsection, we would like to analyze the behavior of bagging FURIA-based fuzzy MCSs composed of a small number of classifiers. As said, ensembles of sizes

Table 4. Results for a single FURIA-based fuzzy classifier without feature selection.

(a) First subset of datasets

FURIA single classifier — All features										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.023	0.336	0.141	0.041	0.038	0.143	0.633	0.018	0.003
test err.	0.805	0.049	0.377	0.227	0.163	0.123	0.157	0.683	0.033	0.027

(b) Second subset of datasets

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.132	0.193	0.042	0.008	0.154	0.043	0.007	0.331	0.043	0.004	0.433
test err.	0.160	0.245	0.122	0.042	0.298	0.070	0.055	0.364	0.187	0.056	0.441

3, 5, 7, and 10 are considered. Table 5 collects the obtained results (the best result for each dataset is shown in boldface). As expected, it can be seen how the use of bagging outperforms the single FURIA-based fuzzy classifier (Table 4) in 19 out of 21 cases for all sizes of the ensembles in terms of testing error. Overall, it outperforms a single FURIA-based fuzzy classifier in 76 out of 84 cases (4 ensemble sizes × 21 datasets). Pima and wine are the only datasets where the single FURIA-based fuzzy classifier turned out to be a better choice.

Thus, we may conclude that FURIA-based fuzzy MCSs with bagging only is a good approach.

Moreover, we would like to provide an analysis of the influence of the ensemble size on the test error. We will compare the following ensemble size parameters in a pairwise manner: 3 vs. 5; 5 vs. 7; and 7 vs. 10. Comparing the ensemble size of 3 against 5, it can be noticed that bagging FURIA-based fuzzy MCSs composed of 5 classifiers obtain the best results in 20 out of 21 cases (+1 tie). Then, comparing the ensemble size of 5 against 7, it can be noticed that bagging FURIA-based fuzzy MCSs composed of 7 classifiers obtain the best results in 15 out of 21 cases (+5 ties). Finally, comparing the ensemble size of 7 against 10, it can be noticed that bagging FURIA-based fuzzy MCSs composed of 10 classifiers obtain the best results in 15 out of 21 cases (+2 ties). It can be seen that globally, the larger the number of classifiers, the lower the test error. However, in some cases (4 out of 21, +2 ties) bagging FURIA-based fuzzy MCSs composed of 7 classifiers outperform those composed of 10 classifiers. Hence, the optimal number of component classifiers for the bagging FURIA-based fuzzy MCSs seem to be an important parameter to keep in mind when designing a classifier system of this kind. As said, we will consider this issue in future works.

5.4. *Comparison of two feature selection approaches for the generation of FURIA-based fuzzy MCSs*

In this subsection we present results from the experiment conducted concerning the use of two different feature selection approaches to generate FURIA-based fuzzy

Table 5. Results for FURIA-based fuzzy MCSs with bagging.

(a) First subset of datasets

FURIA — Bagging with all features										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.617	0.013	0.140	0.078	0.041	0.040	0.114	0.321	0.015	0.006
test err.	0.771	0.045	0.362	0.204	0.156	0.119	0.139	0.664	0.031	0.024
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.586	0.012	0.111	0.057	0.035	0.027	0.111	0.286	0.014	0.004
test err.	0.760	0.044	0.325	0.189	0.156	0.103	0.136	0.652	0.030	0.019
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.578	0.010	0.096	0.052	0.038	0.021	0.110	0.270	0.014	0.003
test err.	0.756	**0.044**	**0.313**	**0.178**	0.156	0.096	**0.136**	0.648	0.030	0.019
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.570	0.009	0.091	0.059	0.031	0.016	0.113	0.246	0.015	0.002
test err.	**0.755**	0.046	0.318	0.189	**0.152**	**0.091**	0.138	**0.641**	**0.030**	**0.017**

(b) Second subset of datasets

FURIA — Bagging with all features											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.090	0.115	0.037	0.013	0.069	0.032	0.012	0.098	0.044	0.018	0.252
test err.	0.144	0.259	0.115	0.041	0.249	0.064	0.050	0.294	0.171	0.067	0.439
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.098	0.029	0.009	0.049	0.028	0.008	0.080	0.030	0.014	0.235
test err.	0.141	0.253	0.108	0.039	0.238	0.062	0.039	0.284	0.164	0.061	0.426
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.084	0.092	0.026	0.007	0.035	0.026	0.006	0.063	0.024	0.011	0.229
test err.	**0.138**	0.250	0.106	0.036	0.232	0.061	0.037	0.282	0.158	0.069	0.416
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.086	0.075	0.025	0.006	0.018	0.028	0.004	0.051	0.017	0.006	0.223
test err.	**0.141**	**0.246**	**0.105**	**0.035**	**0.230**	**0.061**	**0.036**	**0.276**	**0.156**	**0.060**	**0.408**

MCSs, namely random and randomized greedy feature selection (see Sec. 4.1). Note that, as mentioned in that section, greedy feature selection is not considered due to its lack of diversity.

Tables 6, 9 and 12 presents a set of FURIA-based fuzzy MCSs based on Random-greedy feature selection with Small, Medium, and Large feature subset sizes respectively, while Tables 7, 10 and 13 present a set of FURIA-based fuzzy MCSs based on

Table 6. Results for FURIA MCSs with Random-Greedy feature selection. Small feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.779	0.027	0.274	0.124	0.048	0.265	0.170	0.628	0.018	0.065
test err.	0.804	0.044	0.389	0.198	0.147	0.301	0.179	0.628	0.032	0.110
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.027	0.256	0.118	0.051	0.181	0.168	0.628	0.018	0.053
test err.	0.803	0.041	0.377	0.189	0.142	0.222	0.178	0.628	0.032	0.092
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.778	0.026	0.231	0.121	0.050	0.174	0.167	0.628	0.018	0.048
test err.	0.802	0.040	0.366	0.192	**0.134**	0.213	0.176	0.628	0.032	0.085
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.784	0.027	0.225	0.122	0.043	0.154	0.169	0.628	0.018	0.048
test err.	0.806	0.043	0.352	0.188	0.140	0.193	0.178	0.628	0.032	0.088

(b) Second subset of datasets

FURIA — Random-greedy feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.203	0.216	0.142	0.027	0.108	0.139	0.118	0.298	0.241	0.005	0.478
test err.	0.217	0.252	0.166	0.059	0.264	0.149	0.175	0.351	0.271	0.065	0.544
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.199	0.215	0.129	0.026	0.083	0.133	0.075	0.286	0.206	0.008	0.468
test err.	0.212	0.248	0.150	0.059	0.254	0.143	0.124	0.350	0.240	0.055	0.539
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.199	0.215	0.123	0.024	0.084	0.132	0.071	0.280	0.188	0.006	0.448
test err.	0.212	0.248	0.143	0.055	0.252	0.144	0.119	0.349	0.219	0.059	0.525
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.200	0.215	0.118	0.026	0.065	0.135	0.057	0.285	0.158	0.004	0.434
test err.	0.214	0.248	0.138	0.056	0.249	0.145	0.100	0.349	0.196	0.055	0.509

Random subspace feature selection with Small, Medium, and Large feature subset sizes respectively (the best result for each dataset is shown in boldface). Each table shows different sizes of MCSs from 3 to 10, namely 3, 5, 7, and 10.

A comparison between FURIA-based fuzzy MCSs based on Random-greedy feature selection and FURIA-based fuzzy MCSs based on Random subspace feature

Table 7. Results for FURIA MCSs with Random subspace feature selection. Small feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.792	0.031	0.265	0.147	0.053	0.416	0.178	0.630	0.021	0.053
test err.	0.815	0.047	0.395	0.228	0.163	0.446	0.186	0.631	0.035	0.096
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.792	0.026	0.212	0.142	0.049	0.311	0.169	0.628	0.020	0.029
test err.	0.814	0.041	0.363	0.244	0.159	0.347	0.179	0.628	0.035	0.062
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.787	0.026	0.207	0.138	0.048	0.282	0.166	0.628	0.022	0.021
test err.	0.809	0.039	0.384	0.235	0.157	0.315	0.175	0.628	0.034	0.047
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.783	0.027	0.204	0.129	0.041	0.252	0.201	0.628	0.021	0.020
test err.	0.808	0.039	0.380	0.217	0.154	0.285	0.207	0.628	0.035	0.045

(b) Second subset of datasets

FURIA — Random subspace feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.183	0.223	0.174	0.064	0.145	0.180	0.141	0.284	0.354	0.011	0.489
test err.	0.200	0.255	0.200	0.114	0.291	0.187	0.203	0.375	0.385	0.064	0.527
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.174	0.221	0.129	0.034	0.120	0.170	0.097	0.279	0.343	0.005	0.458
test err.	0.193	0.254	0.149	0.073	0.292	0.177	0.151	0.361	0.370	0.054	0.513
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.170	0.218	0.119	0.036	0.092	0.160	0.082	0.271	0.321	0.003	0.505
test err.	0.189	0.259	0.140	0.076	0.271	0.164	0.132	0.354	0.349	0.040	0.554
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.177	0.211	0.116	0.030	0.060	0.165	0.061	0.264	0.247	0.003	0.517
test err.	0.195	0.249	0.135	0.071	0.246	0.171	0.108	0.343	0.275	0.042	0.573

selection with Small, Medium, and Large feature subset sizes respectively in is presented in Tables 8, 11 and 14. These tables are formulated in terms of a summarized matrix showing the number of wins, ties, and loses obtained for the two feature selection algorithms for each ensemble size.

Table 8. Comparison of results for each of the feature selection approaches for Small feature subset size of FURIA-based fuzzy MCSs generated with feature selection only in the form of a summarized matrix.

	Random-greedy vs. Random		
# Classif.	W	T	L
3	17	0	4
5	13	2	6
7	14	1	6
10	13	1	7
Overall	57	4	23

We will do three types of analyses of the obtained results. In the first analysis, we will compare the two different feature selection algorithms between them, in the second we will compare the different sizes of feature selection subsets considered, and finally we will benchmark the FURIA-based fuzzy MCS derived by the best previous feature selection approach against the single FURIA-based fuzzy classifier.

5.4.1. *Feature selection approaches*

In our first analysis, we are analyzing the influence of the use of the two different feature selection algorithms. We will consider Small, Medium, and Large feature subsets separately. We will first focus on Small feature subsets (Table 8). From this table, it can be noticed that the Random-greedy approach seems to perform better when considering Small feature subsets overall.

Let us consider now the analysis of Medium feature subsets (Table 11). From this table, it can be noticed that the conclusion drawn in the previous paragraph is not as clear as in the previous case. Notice that, the performance of the Random subspace approach improves as long as the number of component classifiers is increased obtaining better results when considering the ensemble size 10.

Finally, let us consider Large feature subsets (Table 14). From this table, it can be noticed that the Random subspace approach again performs better as long as the ensemble size is increased.

In summary, taking into account all the ensemble sizes, the Random-greedy approach obtains the best results in 139 out of 252 cases (+24 ties), while Random subspace does so in 89 cases (+24 ties). The summary of the results is presented in Table 15 in terms of a summarized matrix showing the number of wins, ties, and loses obtained for the two feature selection algorithms for each ensemble size. In view of these results, we will consider Random-greedy as the best choice from now on.

5.4.2. *Feature selection subset sizes*

In our second analysis, we are comparing the different sizes (Small, Medium, and Large) for the considered feature selection subsets in order to determine the influence of this parameters. From the results reported in Table 16, it can be

Table 9. Results for FURIA MCSs with Random-Greedy feature selection. Medium feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.786	0.029	0.209	0.118	0.042	0.092	0.163	0.630	0.016	0.014
test err.	0.812	0.043	0.373	0.198	0.160	0.155	0.174	0.631	0.029	0.050

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.785	0.025	0.192	0.117	0.037	0.080	0.164	0.628	0.016	0.011
test err.	0.810	0.040	0.352	0.193	0.154	0.139	0.175	0.629	0.029	0.045

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.779	0.022	0.190	0.113	0.037	0.068	0.163	0.628	0.016	0.013
test err.	0.805	0.042	0.353	**0.185**	0.142	0.124	0.174	0.628	0.029	0.046

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.021	0.199	0.118	0.037	0.064	0.164	0.628	0.017	0.012
test err.	0.807	0.043	0.363	0.196	0.145	0.119	0.175	0.628	0.029	0.045

(b) Second subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.159	0.208	0.083	0.022	0.052	0.102	0.029	0.260	0.118	0.008	0.400
test err.	0.184	0.244	0.129	0.051	0.272	0.114	0.087	0.336	0.187	0.059	0.481

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.154	0.212	0.077	0.030	0.045	0.098	0.022	0.252	0.105	0.005	0.398
test err.	0.182	0.247	0.123	0.062	0.252	0.111	0.069	0.331	0.176	0.063	0.481

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.155	0.213	0.075	0.019	0.046	0.097	0.016	0.245	0.095	0.004	0.398
test err.	0.183	0.245	0.119	0.045	0.249	0.110	0.058	0.332	0.169	0.062	0.480

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.157	0.213	0.073	0.019	0.033	0.099	0.013	0.243	0.089	0.004	0.400
test err.	0.184	0.247	0.117	0.041	0.252	0.111	0.051	0.328	**0.164**	0.065	0.482

noticed that the Large feature subsets for generating FURIA-based fuzzy MCSs significantly outperform the other sizes. This is a sensible result keeping in mind that FURIA incorporates an advanced feature selection criterion based on an information gain measure. This conclusion is confirmed in Table 17 showing average and standard deviation values computed for each of the feature selection approaches for the different ensemble sizes.

Table 10. Results for FURIA MCSs with Random subspace feature selection. Medium feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.804	0.030	0.216	0.138	0.054	0.260	0.154	0.628	0.018	0.010
test err.	0.825	0.043	0.372	0.227	0.170	0.324	0.164	0.628	0.032	0.046

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.787	0.028	0.200	0.140	0.046	0.154	0.154	0.628	0.017	0.006
test err.	0.810	0.043	0.361	0.244	0.161	0.216	0.164	0.628	0.030	0.030

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.783	0.027	0.196	0.121	0.038	0.143	0.147	0.628	0.016	0.004
test err.	0.807	0.042	0.363	0.213	0.156	0.203	0.158	0.628	0.029	0.023

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.767	0.027	0.179	0.130	0.036	0.115	0.158	0.628	0.016	0.003
test err.	0.795	0.043	0.346	0.218	0.153	0.173	0.167	0.628	0.029	0.021

(b) Second subset of datasets

FURIA — Random subspace feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.157	0.225	0.088	0.029	0.042	0.133	0.027	0.268	0.268	0.004	0.466
test err.	0.182	0.259	0.132	0.069	0.235	0.144	0.085	0.338	0.311	0.060	0.510

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.155	0.216	0.077	0.025	0.025	0.103	0.019	0.251	0.204	0.004	0.451
test err.	0.181	0.257	0.122	0.066	0.228	0.112	0.069	0.328	0.247	0.049	0.503

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.148	0.214	0.074	0.022	0.013	0.101	0.015	0.244	0.185	0.003	0.438
test err.	0.173	0.257	0.119	0.057	0.208	0.110	0.059	0.326	0.226	0.044	0.493

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.149	0.213	0.072	0.020	0.004	0.101	0.012	0.244	0.171	0.002	0.421
test err.	0.176	0.254	0.117	0.054	0.198	0.110	0.053	0.322	0.215	0.**036**	0.475

Considering the conclusions obtained in the first analysis (see previous subsection) and the current ones, from now on we will select the Random-greedy feature selection approach with Large feature subsets when dealing with FURIA-based fuzzy MCSs with feature selection in isolation. In Table 17 it can be seen that FURIA-based fuzzy MCSs based on Random-greedy outperform FURIA-based fuzzy MCSs

Table 11. Comparison of results for each of the feature selection approaches for Medium feature subset size of FURIA-based fuzzy MCSs generated with feature selection only in the form of a summarized matrix.

	Random-greedy vs. Random		
# Classif.	W	T	L
3	13	1	7
5	11	2	8
7	10	5	6
10	7	4	10
Overall	41	12	31

based on Random subspace for all the feature subset sizes. Notice that, the global average and standard deviation values, which are presented in the last column of the table, also show how Random-greedy presents an advantage over the latter approach.

5.4.3. *Benchmarking against the single FURIA-based fuzzy classifier*

In our third analysis, we are comparing the FURIA-based fuzzy MCS derived by the best previous feature selection approach against the single FURIA-based fuzzy classifier. In view of Table 18, it can be noticed that the performance of FURIA-based fuzzy MCS derived by the Random-greedy is lower than those obtained by the bagging FURIA-based fuzzy MCS without feature selection (see Sec. 5.3). While the latter approach outperformed the single classifier in 76 out of 84 cases, the former one only does so in 64 cases. This performance decrease is related to the already mentioned inner feature selection mechanism on FURIA, which could make bagging better than an additional feature selection approach to induce diversity in a FURIA-based fuzzy MCS. This issue will be analyzed more deeply in Sec. 5.6.

5.5. *Combination of FURIA with bagging and feature selection*

In this subsection, we present the results of the FURIA-based fuzzy MCSs obtained from the combination of bagging and the three feature selection algorithms considered (see Sec. 4.1). In the previous subsection we have skipped Greedy Battiti's MIFS because of its inability to induce an appropriate diversity, however here it could become a good choice when combined with bagging. This experiment is made with the aim to check if, as expected, the additional diversity induced when combining both MCS design methodologies allows us to generate the most accurate ensembles as happened with other kinds of classifiers.[13,14]

Each table (Tables from 19 to 29) presents a set of FURIA-based fuzzy MCSs with different ensemble sizes. The combination of each feature selection algorithm with a different feature subset size is shown in a different table.

A comparison between FURIA-based fuzzy MCSs based on bagging and each feature selection algorithm with Small, Medium, and Large feature subset sizes

Table 12. Results for FURIA MCSs with Random-Greedy feature selection. Large feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.023	0.196	0.104	0.041	0.052	0.140	0.629	0.015	0.006
test err.	0.797	0.040	0.356	0.200	0.150	0.121	0.153	0.632	0.030	0.036
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.770	0.021	0.198	0.105	0.042	0.041	0.139	0.628	0.015	0.004
test err.	0.796	0.041	0.363	0.204	0.152	0.105	0.152	0.630	0.029	0.030
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.761	0.020	0.204	0.108	0.039	0.039	0.139	0.628	0.015	0.004
test err.	0.789	0.043	0.361	0.206	0.152	0.102	0.151	0.629	0.028	0.027
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.757	0.018	0.208	0.107	0.036	0.037	0.139	0.627	0.015	0.003
test err.	0.787	0.043	0.364	0.202	0.149	**0.101**	**0.151**	**0.628**	0.028	0.026

(b) Second subset of datasets

FURIA — Random-greedy feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.133	0.205	0.063	0.019	0.034	0.084	0.010	0.229	0.074	0.004	0.362
test err.	0.161	0.246	0.121	0.043	0.247	0.098	0.054	0.322	0.174	0.058	0.438
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.131	0.203	0.057	0.018	0.032	0.078	0.007	0.227	0.067	0.003	0.362
test err.	0.160	0.245	0.114	0.040	0.255	0.091	0.046	0.313	0.169	0.050	0.447
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.132	0.204	0.055	0.017	0.025	0.075	0.006	0.228	0.065	0.004	0.358
test err.	0.160	**0.244**	0.113	0.039	0.245	**0.088**	0.044	0.316	0.167	0.048	0.446
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.135	0.205	0.053	0.018	0.022	0.079	0.006	0.223	0.062	0.004	0.361
test err.	0.163	0.246	0.110	**0.039**	0.250	0.091	**0.041**	0.311	0.165	0.056	0.448

respectively in terms of a summarized matrix showing the number of wins, ties, and loses obtained for the three feature selection algorithms for each ensemble size is presented in Tables from 22, 26, and 30 to 33.

We will do three types of analyses taking into account the test errors obtained. In the first analysis, we will compare the performance of the three different

Table 13. Results for FURIA MCSs with Random subspace feature selection. Large feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.024	0.163	0.129	0.035	0.115	0.142	0.629	0.015	0.005
test err.	0.804	0.046	0.341	0.229	0.161	0.195	0.157	0.631	0.029	0.031
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.760	0.022	0.142	0.108	0.029	0.057	0.140	0.628	0.014	0.003
test err.	0.792	0.042	0.320	0.206	0.152	0.127	0.153	0.628	0.028	0.022
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.759	0.021	0.142	0.110	0.033	0.051	0.139	0.628	0.014	0.002
test err.	0.793	0.037	0.324	0.204	0.147	0.119	0.152	0.628	0.028	0.018
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.754	0.018	0.146	0.107	0.029	0.042	0.140	0.628	0.014	0.002
test err.	**0.786**	**0.037**	**0.316**	0.206	0.147	0.105	0.153	0.628	**0.028**	**0.015**

(b) Second subset of datasets

FURIA — Random subspace feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.123	0.207	0.063	0.015	0.030	0.086	0.011	0.234	0.161	0.007	0.426
test err.	0.155	0.251	0.125	0.048	0.233	0.099	0.060	0.318	0.225	0.059	0.503
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.122	0.213	0.058	0.013	0.013	0.077	0.007	0.222	0.131	0.007	0.380
test err.	0.153	0.256	0.116	0.042	0.214	0.089	0.047	0.315	0.201	0.061	0.454
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.120	0.213	0.055	0.015	0.010	0.074	0.005	0.220	0.119	0.005	0.370
test err.	**0.153**	0.254	0.114	0.046	0.206	0.089	0.044	0.316	0.190	0.057	0.438
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.123	0.207	0.052	0.015	0.005	0.075	0.005	0.217	0.102	0.002	0.364
test err.	0.156	0.253	**0.110**	0.044	**0.198**	0.090	0.041	**0.310**	0.180	0.054	**0.432**

feature selection algorithms, in the second analysis we will compare the different sizes (Small, Medium, and Large) for the feature selection subsets, and finally we will benchmark the FURIA-based fuzzy MCS derived by the best previous feature selection approach against the single FURIA-based fuzzy classifier.

Table 14. Comparison of results for each of the feature selection approaches for Large feature subset size of FURIA-based fuzzy MCSs generated with feature selection only in the form of a summarized matrix.

	Random-greedy vs. Random		
# Classif.	W	T	L
3	14	0	7
5	12	1	8
7	9	3	9
10	6	4	11
Overall	41	8	35

Table 15. Comparison of results for each of the feature selection approaches for all feature subset sizes of FURIA-based fuzzy MCSs generated with feature selection only in the form of a summarized matrix.

	Random-greedy vs. Random		
# Classif.	W	T	L
3	44	1	18
5	36	5	22
7	33	9	21
10	26	9	28
Overall	139	24	89

Table 16. Comparison of results for each of the feature subset sizes of FURIA-based fuzzy MCSs generated with feature selection only in the form of a summarized matrix.

	Small			Medium			Large		
# Classif.	W	T	L	W	T	L	W	T	L
3	2	1	39	5	1	36	34	0	8
5	5	1	36	5	1	36	31	1	10
7	3	2	37	2	2	38	34	1	7
10	6	3	33	2	3	37	31	3	8
Overall	16	7	145	14	7	147	130	5	33

Table 17. Average results for each of the feature selection approaches of FURIA-based fuzzy MCSs generated with feature selection only.

F.S. approach		3 Cl.	5 Cl.	7 Cl.	10 Cl.	Global
Random-greedy	avg.	**0.232**	**0.225**	**0.222**	**0.221**	**0.225**
	std. dev.	0.199	0.200	0.200	0.200	0.200
Random	avg.	0.249	0.234	0.229	0.223	0.234
	std. dev.	0.202	0.200	0.202	0.201	0.201

Table 18. Comparison of results for the Random-greedy feature selection approach for Large feature subset size of FURIA-based fuzzy MCSs generated with feature selection only compared with single FURIA in the form of a summarized matrix.

	Random-greedy vs. Single		
# Classif.	W	T	L
3	15	0	6
5	16	2	3
7	17	2	2
10	16	1	4
Overall	64	5	15

Table 19. FURIA-based fuzzy MCSs for small ensemble sizes with bagging and Greedy feature selection. Small feature subsets.

(a) First subset of datasets

FURIA — Greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.735	0.030	0.197	0.136	0.044	0.254	0.162	0.532	0.018	0.061
test err.	0.790	0.051	0.375	0.209	0.155	0.300	0.176	0.662	0.034	0.113

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.697	0.027	0.170	0.124	0.045	0.243	0.158	0.529	0.018	0.051
test err.	0.764	0.049	0.360	0.196	0.157	0.290	0.174	0.653	0.034	0.103

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.690	0.025	0.140	0.107	0.050	0.239	0.156	0.532	0.018	0.047
test err.	0.764	0.047	0.337	0.187	0.155	0.286	0.171	0.645	0.033	0.101

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.684	0.025	0.136	0.098	0.047	0.235	0.158	0.538	0.018	0.044
test err.	0.759	0.047	0.337	0.191	0.152	0.281	0.171	0.637	0.033	0.099

(b) Second subset of datasets

FURIA — Greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.176	0.160	0.128	0.019	0.084	0.125	0.095	0.208	0.165	0.028	0.417
test err.	0.204	0.250	0.155	0.044	0.268	0.142	0.166	0.346	0.237	0.074	0.519

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.172	0.157	0.123	0.016	0.059	0.122	0.085	0.194	0.156	0.014	0.408
test err.	0.203	0.244	0.151	0.040	0.249	0.140	0.156	0.340	0.230	0.058	0.515

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.168	0.158	0.121	0.015	0.045	0.122	0.079	0.187	0.154	0.009	0.400
test err.	0.200	0.241	0.149	0.038	0.251	0.139	0.150	0.340	0.227	0.055	0.509

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.168	0.156	0.118	0.014	0.038	0.121	0.079	0.183	0.153	0.003	0.392
test err.	0.199	0.241	0.147	0.037	0.253	0.140	0.149	0.329	0.228	0.045	0.511

5.5.1. *Feature selection approaches*

In our first analysis, we are comparing the three different feature selection algorithms among them.

Looking at all Small, Medium, and Large feature subsets (Tables 22, 26, and 30) it can be noticed the same conclusion. The three different feature selection approaches perform quite similarly.

Table 20. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random–greedy feature selection. Small feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.779	0.028	0.162	0.116	0.042	0.227	0.156	0.588	0.018	0.065
test err.	0.804	0.045	0.371	0.202	0.152	0.278	0.171	0.650	0.034	0.110
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.026	0.139	0.103	0.038	0.151	0.154	0.587	0.018	0.053
test err.	0.803	0.044	0.351	0.195	0.147	0.203	0.169	0.640	0.034	0.092
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.778	0.023	0.131	0.096	0.035	0.137	0.154	0.594	0.017	0.048
test err.	0.802	0.042	0.345	0.189	0.143	0.188	0.168	0.633	0.033	0.085
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.784	0.024	0.123	0.100	0.032	0.120	0.157	0.605	0.019	0.048
test err.	0.806	0.043	0.334	0.197	**0.143**	0.167	0.171	0.630	0.035	0.088

(b) Second subset of datasets

FURIA — Random-greedy feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.175	0.165	0.142	0.033	0.080	0.139	0.118	0.195	0.241	0.005	0.391
test err.	0.204	0.253	0.166	0.069	0.283	0.149	0.175	0.337	0.271	0.065	0.494
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.171	0.163	0.129	0.028	0.058	0.133	0.075	0.172	0.206	0.008	0.387
test err.	0.202	0.245	0.150	0.065	0.281	0.143	0.124	0.325	0.240	0.055	0.490
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.166	0.160	0.123	0.022	0.049	0.132	0.071	0.159	0.188	0.006	0.369
test err.	0.197	0.245	0.143	0.059	0.265	0.144	0.119	0.318	0.219	0.059	0.483
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.166	0.151	0.118	0.022	0.028	0.135	0.057	0.151	0.158	0.004	0.358
test err.	0.197	0.240	0.138	0.060	0.254	0.145	0.100	0.321	0.196	0.055	0.476

In view of the results obtained it is rather hard to point out one of the solutions. Table 31 summarizes the obtained results in the form of a summarized matrix showing the number of wins, ties, and loses for three feature selection algorithms for each ensemble size. In view of the overall results, collected in the best row of the table, we can maybe highlight the performance of the Greedy feature

Table 21. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random subspace feature selection. Small feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.792	0.028	0.210	0.125	0.056	0.384	0.160	0.620	0.021	0.053
test err.	0.815	0.050	0.418	0.226	0.175	0.431	0.176	0.649	0.037	0.096
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.792	0.027	0.169	0.111	0.046	0.272	0.153	0.618	0.018	0.029
test err.	0.814	0.046	0.361	0.223	0.160	0.323	0.167	0.639	0.035	0.062
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.787	0.026	0.161	0.096	0.044	0.233	0.148	0.620	0.017	0.021
test err.	0.809	0.041	0.358	0.210	0.156	0.283	0.163	0.632	0.035	0.047
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.783	0.026	0.134	0.101	0.038	0.200	0.181	0.621	0.018	0.020
test err.	0.808	0.041	0.346	0.203	0.149	0.251	0.195	**0.630**	0.036	0.045

(b) Second subset of datasets

FURIA — Random subspace feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.159	0.194	0.174	0.047	0.073	0.180	0.141	0.188	0.354	0.011	0.472
test err.	0.191	0.270	0.200	0.106	0.309	0.187	0.203	0.367	0.385	0.064	0.554
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.157	0.179	0.129	0.028	0.053	0.170	0.097	0.165	0.343	0.005	0.405
test err.	0.187	0.262	0.149	0.075	0.270	0.177	0.151	0.348	0.370	0.054	0.495
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.148	0.180	0.119	0.026	0.028	0.160	0.082	0.163	0.321	0.003	0.409
test err.	0.180	0.266	0.140	0.075	0.250	0.164	0.132	0.334	0.349	0.040	0.506
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.152	0.165	0.116	0.018	0.013	0.165	0.061	0.143	0.247	0.003	0.412
test err.	0.183	0.261	0.135	0.066	0.258	0.171	0.108	0.322	0.275	0.042	0.500

selection approach to generate FURIA-based fuzzy MCSs when combined with bagging. Nevertheless, the results are still not so conclusive. We will try to draw more categorical conclusions in the next subsection.

Table 22. Comparison of results for each of the feature selection approaches for Small feature subset size of FURIA-based fuzzy MCSs generated with bagging and feature selection in the form of a summarized matrix.

	Greedy			Random-greedy			Random		
# Classif.	W	T	L	W	T	L	W	T	L
3	9	1	11	7	1	13	4	0	17
5	6	1	14	7	1	13	7	0	14
7	6	1	14	6	1	14	8	0	13
10	7	1	13	7	2	12	5	1	15
Overall	28	4	52	27	5	52	24	1	59

5.5.2. *Feature selection subset sizes*

In our second analysis, we are comparing different sizes (Small, Medium, and Large) for feature selection subsets. From results reported in Table 32, it can be noticed that the use of Large feature subsets for generating FURIA-based fuzzy MCSs considering both bagging and feature selection allows us to significantly outperform the other feature subset sizes.

Considering a comparison between these three feature selection approaches for Large feature subsets (i.e., recalling Table 30) it is still ambiguous to determine which approach is the best one. Although the Greedy feature selection approach seems to obtain the best performance, this conclusion is deceptive as it is strongly biased by the combinations with the smallest number of classifiers, which are the worst performing ones overall.

Because of all the latter facts, let us examine the best overall results for all the combinations (Tables 19 to 21, 23 to 25, 27 to 29, best result for each dataset shown in boldface). Both FURIA-based fuzzy MCSs considering bagging with Random-greedy feature selection and bagging with Random subspace feature selection obtained the best overall performance in 6 out of 21 cases (+4 ties), whereas FURIA-based fuzzy MCSs considering bagging and Greedy feature selection does so 4 times (+3 ties). In view of this analysis and that developed in the previous subsection, from now on we will take into account only the Random-greedy feature selection with Large feature subsets, when dealing with FURIA-based fuzzy MCSs with bagging and feature selection.

This conclusion is confirmed in Table 33 presenting average and standard deviation values computed for each of the feature selection approaches for different ensemble sizes. It can be seen that FURIA-based fuzzy MCSs based on Random-greedy outperform FURIA-based fuzzy MCSs based on the other two feature selection approaches in all the cases. Considering the global average and standard deviation values, which are presented in the last column of the table, Random-greedy also presented advantage over the other two approaches, although the differences with respect to the Greedy method are not very significant.

Table 23. FURIA-based fuzzy MCSs for small ensemble sizes with bagging and Greedy feature selection. Medium feature subsets.

(a) First subset of datasets

FURIA — Greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.690	0.023	0.184	0.112	0.045	0.117	0.144	0.505	0.017	0.025
test err.	0.779	0.045	0.366	0.190	0.167	0.192	0.167	0.660	0.034	0.067

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.665	0.021	0.150	0.093	0.042	0.103	0.143	0.493	0.015	0.019
test err.	0.763	0.045	0.338	0.185	0.158	0.180	0.164	0.652	0.034	0.062

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.659	0.021	0.132	0.088	0.039	0.097	0.141	0.495	0.014	0.017
test err.	0.760	0.043	0.331	0.181	0.152	0.175	0.162	0.642	0.033	0.059

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.659	0.021	0.132	0.088	0.039	0.097	0.141	0.495	0.014	0.017
test err.	0.760	0.043	0.331	0.181	0.152	0.175	0.162	0.642	0.033	0.059

(b) Second subset of datasets

FURIA — Greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.130	0.137	0.068	0.018	0.073	0.093	0.029	0.148	0.079	0.027	0.332
test err.	0.167	0.256	0.127	0.045	0.263	0.113	0.080	0.329	0.187	0.079	0.482

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.125	0.133	0.062	0.015	0.054	0.091	0.021	0.126	0.067	0.008	0.307
test err.	0.166	0.245	0.122	0.042	0.251	0.110	0.074	0.321	0.180	0.062	0.472

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.120	0.136	0.058	0.014	0.047	0.088	0.018	0.110	0.061	0.009	0.302
test err.	0.163	0.244	0.118	0.039	0.252	0.109	0.070	0.321	0.177	0.059	0.471

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.121	0.124	0.055	0.013	0.032	0.090	0.016	0.103	0.057	0.003	0.302
test err.	0.162	0.243	0.117	0.038	0.260	0.109	0.068	0.317	0.175	0.055	0.466

Table 24. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random–greedy feature selection. Medium feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.786	0.019	0.163	0.095	0.040	0.089	0.144	0.496	0.016	0.014
test err.	0.812	0.045	0.380	0.203	0.161	0.164	0.166	0.661	0.032	0.050

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.785	0.017	0.126	0.082	0.042	0.067	0.142	0.478	0.015	0.011
test err.	0.810	0.044	0.364	0.198	0.155	0.139	0.164	0.650	0.032	0.045

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.779	0.014	0.121	0.077	0.039	0.052	0.141	0.486	0.014	0.013
test err.	0.805	0.042	0.340	0.194	0.148	0.121	0.163	0.642	0.031	0.046

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.017	0.112	0.079	0.031	0.044	0.142	0.486	0.015	0.012
test err.	0.807	0.043	0.321	0.185	0.146	0.112	0.163	0.636	0.031	0.045

(b) Second subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.133	0.135	0.083	0.023	0.073	0.102	0.029	0.148	0.118	0.008	0.325
test err.	0.169	0.251	0.129	0.053	0.260	0.114	0.087	0.325	0.187	0.059	0.465

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.128	0.127	0.077	0.025	0.054	0.098	0.022	0.120	0.105	0.005	0.304
test err.	0.166	0.246	0.123	0.058	0.256	0.111	0.069	0.312	0.176	0.063	0.459

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.123	0.125	0.075	0.019	0.032	0.097	0.016	0.104	0.095	0.004	0.292
test err.	0.162	0.247	0.119	0.044	0.245	0.110	0.058	0.304	0.169	0.062	0.453

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.126	0.120	0.073	0.016	0.024	0.099	0.013	0.098	0.089	0.004	0.292
test err.	0.163	0.240	0.117	0.044	0.245	0.111	0.051	0.309	0.164	0.065	0.453

Table 25. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random subspace feature selection. Medium feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.804	0.027	0.187	0.093	0.039	0.233	0.134	0.529	0.016	0.010
test err.	0.825	0.048	0.379	0.222	0.166	0.324	0.154	0.657	0.035	0.046

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.787	0.023	0.138	0.076	0.032	0.121	0.134	0.517	0.015	0.006
test err.	0.810	0.044	0.339	0.221	0.157	0.210	0.152	0.645	0.034	0.030

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.783	0.021	0.126	0.058	0.032	0.096	0.127	0.519	0.015	0.004
test err.	0.807	**0.039**	0.333	0.215	0.151	0.185	0.146	0.640	0.033	0.023

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.767	0.022	0.101	0.056	0.028	0.073	0.139	0.514	0.015	0.003
test err.	0.795	0.041	0.330	0.203	0.148	0.159	0.156	0.634	0.033	0.021

(b) Second subset of datasets

FURIA — Random subspace feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.122	0.162	0.088	0.026	0.071	0.133	0.027	0.141	0.268	0.004	0.469
test err.	0.166	0.277	0.132	0.074	0.274	0.144	0.085	0.321	0.311	0.060	0.555

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.117	0.155	0.077	0.018	0.040	0.103	0.019	0.110	0.204	0.004	0.396
test err.	0.163	0.253	0.122	0.066	0.243	0.112	0.069	0.311	0.247	0.049	0.494

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.107	0.148	0.074	0.016	0.025	0.101	0.015	0.091	0.185	0.003	0.371
test err.	0.155	0.255	0.119	0.056	0.235	0.110	0.059	0.299	0.226	0.044	0.483

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.111	0.139	0.072	0.013	0.020	0.101	0.012	0.080	0.171	0.002	0.328
test err.	0.155	0.248	0.117	0.055	0.216	0.110	0.053	0.289	0.215	**0.036**	0.456

Table 26. Comparison of results for each of the feature selection approaches for Medium feature subset size of FURIA-based fuzzy MCSs generated with bagging and feature selection in the form of a summarized matrix.

	Greedy			Random-greedy			Random		
# Classif.	W	T	L	W	T	L	W	T	L
3	7	2	12	7	2	12	5	0	16
5	6	1	14	5	2	14	7	3	11
7	7	0	14	6	0	15	8	0	13
10	3	2	16	8	2	11	8	1	12
Overall	23	5	56	26	6	52	28	4	52

5.5.3. *Benchmarking against the single FURIA-based fuzzy classifier*

In our third analysis, we are comparing the FURIA-based fuzzy MCSs derived by the best previous feature selection approach combined with bagging against the single FURIA-based fuzzy classifier. In view of Table 34, it can be noticed that overall, FURIA-based fuzzy MCSs generated from Bagging and Random-greedy feature selection outperform the single classifier in 70 out of 84 cases (+3 ties), an intermediate number between those of the other two variants analyzed in the previous Secs. 5.3 (bagging only, 76) and 5.4 (feature selection only, 64).

5.6. ***Final comparison of FURIA-based fuzzy MCSs***

This subsection presents a joint comparison of all the FURIA-based fuzzy MCSs variants proposed. The main aim of this contribution is to obtain FURIA-based fuzzy MCSs which, apart from improving the accuracy of the single FURIA-based fuzzy classifier, are able to be competitive with the state-of-the-art MCSs when dealing with high dimensional datasets. In principle, it seems that the best choice is a combination between bagging and a feature selection algorithm to obtain well-performing FURIA-based fuzzy MCS, as it should induce a high amount of diversity into the base classifiers.[34,44] In order to test that assumption we will compare this FURIA-based fuzzy MCS approach, that from now on will be called the reference approach, against the remaining variants resulting from FURIA-based fuzzy MCS generation methodology, i.e. the use of bagging and feature selection in isolation.

In addition, in order to test the performance of our approach, we compare it with two state-of-the-art algorithms: C4.5 decision tree[38] MCSs generated from bagging,[17] and random forests.[8] Moreover, we compare it against an application of the fuzzy MCS design approach with other, less powerful, fuzzy classifier derivation method.[13,14] For that we choose Ishibuchi's fuzzy classification rule generation method.[26]

We will therefore develop two different types of analyses, a first one comparing the different proposed approaches to generate FURIA-based fuzzy MCSs in order to determine the best performing one, and a second one comparing the best choices of FURIA-based fuzzy MCSs with C4.5 decision tree ensembles, random forests, and Ishibuchi-based fuzzy MCSs.

Table 27. FURIA-based fuzzy MCSs for small ensemble sizes with bagging and Greedy feature selection. Large feature subsets.

(a) First subset of datasets

FURIA — Greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.657	0.020	0.163	0.095	0.039	0.053	0.115	0.511	0.015	0.015
test err.	0.769	0.051	0.360	0.199	0.161	0.124	0.140	0.664	0.031	0.049

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.642	0.018	0.123	0.090	0.040	0.039	0.114	0.499	0.014	0.011
test err.	0.762	0.047	0.348	0.196	0.157	0.111	0.139	0.654	0.031	0.044

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.625	0.017	0.116	0.073	0.038	0.034	0.114	0.502	0.014	0.009
test err.	0.756	0.044	0.337	0.187	0.153	0.104	0.139	0.643	0.030	0.043

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.622	0.017	0.114	0.074	0.035	0.029	0.116	0.501	0.014	0.008
test err.	**0.753**	0.045	0.335	0.184	0.147	0.100	0.140	0.639	0.030	0.041

(b) Second subset of datasets

FURIA — Greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.135	0.049	0.020	0.070	0.072	0.017	0.123	0.058	0.022	0.278
test err.	0.141	0.254	0.121	0.045	0.248	0.091	0.053	0.318	0.169	0.070	0.432

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.124	0.043	0.017	0.051	0.069	0.012	0.099	0.048	0.009	0.270
test err.	0.137	0.246	0.115	0.043	0.263	0.087	0.048	0.309	0.162	0.057	0.423

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.117	0.040	0.015	0.037	0.067	0.010	0.080	0.044	0.007	0.263
test err.	0.136	0.243	0.112	0.039	0.239	**0.084**	0.046	0.303	0.157	0.052	0.418

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.086	0.109	0.038	0.014	0.026	0.068	0.009	0.071	0.039	0.002	0.257
test err.	0.138	0.240	0.111	0.039	0.242	0.087	0.045	0.300	**0.156**	0.049	**0.416**

Table 28. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random–greedy feature selection. Large feature subsets.

(a) First subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.016	0.155	0.094	0.041	0.054	0.114	0.446	0.015	0.006
test err.	0.797	0.043	0.368	0.200	0.152	0.128	0.141	0.666	0.031	0.036

5 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.770	0.016	0.132	0.084	0.044	0.037	0.114	0.423	0.014	0.004
test err.	0.796	0.044	0.348	0.198	0.147	0.108	0.139	0.656	0.030	0.030

7 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.761	0.014	0.122	0.079	0.040	0.031	0.113	0.421	0.014	0.004
test err.	0.789	0.042	0.344	**0.179**	0.146	0.102	**0.138**	0.646	0.030	0.027

10 classifiers

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.757	0.014	0.106	0.077	0.035	0.026	0.115	0.410	0.014	0.003
test err.	0.787	0.043	0.334	0.187	0.145	0.**096**	0.139	0.640	0.030	0.026

(b) Second subset of datasets

FURIA — Random-greedy feature selection

3 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.088	0.123	0.063	0.020	0.072	0.084	0.010	0.125	0.074	0.004	0.286
test err.	0.141	0.256	0.121	0.046	0.255	0.098	0.054	0.318	0.174	0.058	0.436

5 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.087	0.122	0.057	0.015	0.056	0.078	0.007	0.098	0.067	0.003	0.269
test err.	0.139	0.245	0.114	0.042	0.250	0.091	0.046	0.310	0.169	0.050	0.431

7 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.084	0.115	0.055	0.013	0.032	0.075	0.006	0.081	0.065	0.004	0.263
test err.	0.138	0.246	0.113	0.040	0.241	0.088	0.044	0.300	0.167	0.048	0.425

10 classifiers

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.086	0.109	0.053	0.014	0.027	0.079	0.006	0.072	0.062	0.004	0.249
test err.	0.139	**0.235**	0.110	**0.037**	0.246	0.091	**0.041**	0.292	0.165	0.056	0.423

Table 29. FURIA-based fuzzy MCSs for small ensemble sizes with with bagging and Random subspace feature selection. Large feature subsets.

(a) First subset of datasets

FURIA — Random subspace feature selection										
3 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.772	0.017	0.139	0.080	0.043	0.102	0.120	0.444	0.016	0.005
test err.	0.804	0.043	0.375	0.202	0.165	0.202	0.145	0.665	0.033	0.031
5 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.760	0.017	0.099	0.073	0.041	0.047	0.115	0.417	0.015	0.003
test err.	0.792	0.040	0.339	0.199	0.158	0.132	0.139	0.656	0.031	0.022
7 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.759	0.015	0.087	0.066	0.035	0.034	0.114	0.420	0.014	0.002
test err.	0.793	0.040	**0.318**	0.195	0.157	0.116	0.139	0.644	0.030	0.018
10 classifiers										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.754	0.015	0.075	0.062	0.026	0.025	0.117	0.410	0.015	0.002
test err.	0.786	0.041	0.319	0.191	0.147	0.103	0.140	0.638	**0.030**	**0.015**

(b) Second subset of datasets

FURIA — Random subspace feature selection											
3 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.093	0.134	0.063	0.019	0.070	0.086	0.011	0.117	0.161	0.007	0.365
test err.	0.146	0.269	0.125	0.050	0.258	0.099	0.060	0.312	0.225	0.059	0.479
5 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.086	0.126	0.058	0.014	0.037	0.077	0.007	0.101	0.131	0.007	0.314
test err.	0.139	0.257	0.116	0.045	0.231	0.089	0.047	0.298	0.201	0.061	0.446
7 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.084	0.117	0.055	0.015	0.018	0.074	0.005	0.083	0.119	0.005	0.305
test err.	0.138	0.253	0.114	0.051	**0.214**	0.089	0.044	0.290	0.190	0.057	0.436
10 classifiers											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.095	0.052	0.011	0.016	0.075	0.005	0.076	0.102	0.002	0.290
test err.	**0.136**	0.253	**0.110**	0.047	0.216	0.090	0.041	**0.284**	0.180	0.054	0.431

For our first analysis, we benchmark the average and standard deviation values as well as the best individual results for each dataset computed for the results obtained by the reference approach against all FURIA-based fuzzy MCS variants presented above.

Table 30. Comparison of results for each of the feature selection approaches for Large feature subset size of FURIA-based fuzzy MCSs generated with bagging and feature selection in the form of a summarized matrix.

	Greedy			Random-greedy			Random		
# Classif.	W	T	L	W	T	L	W	T	L
3	13	3	5	2	4	15	2	1	18
5	7	1	13	8	1	12	5	1	15
7	9	1	11	5	2	14	5	2	14
10	6	1	14	5	3	13	7	3	11
Overall	35	6	43	20	10	54	19	7	58

Table 31. Comparison of results for each of the feature selection approaches for all feature subset sizes of FURIA-based fuzzy MCSs generated with bagging and feature selection in the form of a summarized matrix.

	Greedy			Random-greedy			Random		
# Classif.	W	T	L	W	T	L	W	T	L
3	29	6	28	16	7	40	11	1	51
5	19	3	41	20	4	39	19	3	41
7	22	2	39	17	3	43	21	2	40
10	16	4	43	20	7	36	20	5	38
Overall	86	15	151	73	21	158	71	11	170

Table 32. Comparison of results for each of the feature subset sizes of FURIA-based fuzzy MCSs generated with bagging and feature selection in the form of a summarized matrix.

	Small			Medium			Large		
# Classif.	W	T	L	W	T	L	W	T	L
3	5	1	57	4	0	59	53	1	9
5	6	4	53	7	2	54	45	5	13
7	7	1	55	8	1	54	47	1	15
10	7	4	52	7	4	52	44	4	15
Overall	25	10	217	26	7	219	189	7	56

Table 33. Average results for each of the feature selection approaches of FURIA-based fuzzy MCSs generated with bagging and feature selection.

Bag. + F.S.		3 Cl.	5 Cl.	7 Cl.	10 Cl.	Global
Greedy	avg.	**0.231**	0.224	0.220	0.218	0.223
	std. dev.	0.197	0.194	0.193	0.192	0.194
Random-gredy	avg.	**0.231**	**0.223**	**0.217**	**0.214**	**0.221**
	std. dev.	0.200	0.199	0.198	0.198	0.199
Random	avg.	0.253	0.231	0.224	0.218	0.232
	std. dev.	0.206	0.201	0.200	0.198	0.201

Table 34. Comparison of results for the Random-greedy feature selection approach for Large feature subset size of FURIA-based fuzzy MCSs generated with feature selection only compared with single FURIA in the form of a summarized matrix.

	Random-greedy vs. Single		
# Classif.	W	T	L
3	15	0	6
5	17	1	3
7	18	1	2
10	19	1	1
Overall	69	3	12

Table 35. Average and standard deviation values for the different FURIA-based MCS approaches over all the considered datasets.

		3 Cl.	5 Cl.	7 Cl.	10 Cl.	Global
Bagging	avg.	**0.210**	**0.201**	**0.198**	**0.197**	**0.202**
	std. dev.	0.204	0.200	0.198	0.197	0.196
Feat. sel.	avg.	0.240	0.229	0.225	0.222	0.229
	std. dev.	0.200	0.199	0.200	0.199	0.199
Bag. + Feat. sel.	avg.	0.238	0.226	0.220	0.217	0.225
	std. dev.	0.200	0.197	0.196	0.195	0.197

Firstly, we are comparing average and standard deviation values computed for each of the FURIA-based fuzzy MCSs considering all the parameters selected for the different ensemble sizes. These two values constitute a measure of the average performance of the different variants over all considered datasets. Table 35 collects these results where the last column provides global statistics for each of the approaches. Considering all the ensemble sizes and also the global average values, bagging FURIA-based fuzzy MCSs significantly outperform the other two approaches. From this comparison, it seems that the use of the bagging approach in isolation is the best choice. As it has been already mentioned, this could be due to the internal feature selection provided by FURIA. In that case, inducing diversity by an external feature selection is not a good option, since it decreases the information provided to the classifier.

Secondly, in order to compare FURIA-based approaches, we gather the best result of each approach for each dataset independently of the parameter choice such as number of classifiers, feature subset size, and feature selection method. The results are presented in Table 36, which consists of statistics (5 × 2-cv training and testing errors) and algorithm parameters (feature selection algorithm — feat. sel., feature subset size — feat. subset. size, number of classifiers — nr of cl.) for each of the twenty one datasets. The three feature selection algorithms are considered, Greedy — G, Random-greedy — RG, and Random subspace — R, where the feature subset size may be Small — S, Medium — M, and Large — L. The best accuracy obtained for each given dataset is emphasized in bold font.

Table 36. Results for the best choices of each different approach for FURIA-based fuzzy MCS for each dataset.

(a) First subset of datasets

FURIA single classifier — All features

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.781	0.023	0.336	0.141	0.041	0.038	0.143	0.633	0.018	0.003
test err.	0.805	0.049	0.377	0.227	0.163	0.123	0.157	0.683	0.033	0.027

FURIA-based MCSs obtained from bagging only.

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.570	0.010	0.096	0.052	0.031	0.016	0.110	0.246	0.015	0.002
test err.	0.755	0.044	**0.313**	**0.178**	0.152	**0.091**	**0.136**	0.641	0.030	0.017
nr of cl.	10	7	7	7	10	10	7	10	10	10

FURIA-based MCSs obtained from feature selection only.

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.754	0.018	0.146	0.113	0.050	0.037	0.139	0.627	0.014	0.002
test err.	0.786	**0.037**	0.316	0.185	**0.134**	0.101	0.151	**0.628**	**0.028**	**0.015**
feat. sel.	R	R	R	RG	RG	RG	RG	RG	R	R
feat. sub. size	L	L	L	M	S	L	L	L	L	L
nr of cl.	10	10	10	7	7	10	10	10	10	10

FURIA-based MCSs obtained from bagging and feature selection.

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.622	0.021	0.087	0.079	0.032	0.026	0.113	0.621	0.015	0.020
test err.	**0.753**	0.039	0.318	0.179	0.143	0.096	0.138	0.630	0.030	**0.015**
feat. sel.	G	R	R	RG	RG	RG	RG	R	R	R
feat. sub. size	L	M	L	L	S	L	L	S	L	L
nr of cl.	10	7	7	7	10	10	7	10	10	10

(b) Second subset of datasets

FURIA single classifier — All features

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.132	0.193	0.042	0.008	0.154	0.043	0.007	0.331	0.043	0.004	0.433
test err.	0.160	0.245	0.122	0.042	0.298	0.070	0.055	0.364	0.187	0.056	0.441

FURIA-based MCSs obtained from bagging only.

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.084	0.075	0.025	0.006	0.018	0.028	0.004	0.051	0.017	0.006	0.223
test err.	0.138	0.246	**0.105**	**0.035**	0.230	**0.061**	**0.036**	**0.276**	**0.156**	0.060	**0.408**
nr of cl.	7	10	10	10	10	10	10	10	10	10	10

FURIA-based MCSs obtained from feature selection only.

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.120	0.204	0.052	0.018	0.005	0.075	0.006	0.217	0.089	0.002	0.364
test err.	0.153	0.244	0.110	0.039	**0.198**	0.088	0.041	0.310	0.164	**0.036**	0.432
feat. sel.	R	RG	R	RG	R	RG	RG	R	RG	R	R
feat. sub. size	L	L	L	L	L	L	L	L	M	M	L
nr of cl.	7	7	10	10	10	7	10	10	10	10	10

FURIA-based MCSs obtained from bagging and feature selection.

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.109	0.052	0.014	0.018	0.067	0.005	0.076	0.039	0.020	0.257
test err.	**0.136**	**0.235**	0.110	0.037	0.214	0.084	0.041	0.284	0.156	**0.036**	0.416
feat. sel.	R	RG	R	RG	R	G	RG	R	G	R	G
feat. sub. size	L	L	L	L	L	L	L	L	L	M	L
nr of cl.	10	10	10	10	7	7	10	10	10	10	10

In view of those results, FURIA-based MCSs obtained from bagging obtain the best global result in 10 out of 21 cases (+1 tie), placing FURIA-based MCSs obtained from feature selection in the second place with 5 out of 21 best results (+2 ties). Finally, FURIA-based MCSs obtained from bagging and feature selection outperformed the other two FURIA-based approaches in only 3 out of 21 cases (+3 ties).

Hence, it seems that FURIA-based MCSs obtained from bagging is the best choice, especially when dealing with high dimensional datasets such like letter, magic, sat, segment, spambase, texture, and waveform. However, it is difficult to say that only bagging FURIA-based MCSs deals well with high dimensional datasets, since FURIA-based MCSs obtained from feature selection obtains the best results for optdigits and pendigits. FURIA-based MCSs obtained from joint bagging and feature selection, which was originally considered as the reference approach, turned out to be a rather secondary choice performing well only with a few datasets (e.g. pendigits, and waveform).

Finally, let us develop here a comparison between FURIA-based fuzzy MCSs and the single FURIA classifier. It can be noticed that, in every case, FURIA-based MCSs overcome the single classifier. Besides, each of the three variants does so in 19 out of 21 cases.

In our second analysis, we are comparing the best choices of FURIA-based fuzzy MCSs with two state-of-the-art algorithms, bagging C4.5 MCSs and random forests, as well as with the use of the same methodology combined with a different fuzzy classifier generation method, Ishibuchi-based fuzzy MCS. The obtained results are presented in Table 37, which consists of 5 × 2-cv train and test error values. In all algorithms, we only consider the best obtained result in terms of accuracy for each dataset and highlighted the best values in boldface.

The following conclusions arise comparing FURIA-based fuzzy MCSs to C4.5 MCSs, random forests, and Ishibuchi-based fuzzy MCS: our approach outperforms the other algorithms in 11 out of 21 cases, while random forests obtains the best result in the 7 cases (+1 tie). Ishibuchi-based fuzzy MCSs obtain the best result twice, while C4.5 MCSs only obtains one tie. Note that our approach shows the best performance in 5 out of 10 high dimensional datasets (sonar, optdigits, pendigits, texture, waveform).

We were also interested in answering the question: would another evaluation metric change the latter conclusion? To do so, Table 38 shows a comparison of the same four methods based on the AUC metric values. In view of those results, similar observations are found. FURIA-based fuzzy MCSs outperform the other algorithms in 9 out of 21 cases (+2 ties), while random forests obtains the best result in the 7 cases (+2 tie). C4.5 MCSs achieve the best result four times, while Ishibuchi-based fuzzy MCSs do so only once.

Table 37. A comparison of the best choice for different approaches for FURIA-based fuzzy MCSs against the best choice of bagging C4.5 MCSs, random forests, and Ishibuchi-based fuzzy MCSs.

(a) First subset of datasets

FURIA-based MCSs										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.622	0.018	0.096	0.052	0.050	0.016	0.110	0.627	0.014	0.002
test err.	0.753	**0.037**	0.313	**0.178**	0.134	0.091	0.136	**0.628**	**0.028**	**0.015**
feat sel.	G	R	—	—	RG	—	—	RG	R	R
feat. sub. size	L	L	—	—	S	—	—	L	L	L
nr of cl.	10	10	7	7	7	10	7	10	10	10

C4.5 ensembles with bagging										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
tra. err.	0.118	0.017	0.075	0.053	0.021	0.018	0.052	0.105	0.012	0.005
test err.	0.772	0.043	0.306	0.194	0.149	0.103	**0.134**	0.697	0.030	0.028
nr of cl.	10	7	10	10	10	10	10	10	10	10

random forests										
tra. err.	0.002	0.001	0.001	0.001	0.001	0.000	0.003	0.003	0.002	0.000
test err.	0.777	0.041	**0.282**	0.211	0.140	**0.080**	**0.134**	0.695	0.031	0.016
nr of cl.	7	7	10	10	10	10	10	10	10	10

Ishibuchi-based fuzzy MCSs										
tra. err.	0.732	0.010	0.279	0.093	0.047	0.411	0.199	0.612	0.073	0.054
test err.	**0.751**	0.056	0.379	0.213	**0.129**	0.420	0.202	0.629	0.075	0.062
nr of cl.	3	7	7	10	7	10	7	3	7	10
feat. sel.	R	R	G	R	RG	RG	R	R	RG	R

(b) Second subset of datasets

FURIA-based MCSs											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.085	0.109	0.025	0.006	0.005	0.028	0.004	0.051	0.017	0.002	0.223
test err.	0.136	**0.235**	0.105	0.035	**0.198**	0.061	**0.036**	0.276	**0.156**	**0.036**	**0.408**
feat sel.	R	RG	—	—	R	—	—	—	—	RG	—
feat. sub. size	L	L	—	—	L	—	—	—	—	M	—
nr of cl.	10	10	10	10	10	10	10	10	10	10	10

C4.5 ensembles with bagging											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra. err.	0.044	0.056	0.021	0.009	0.024	0.025	0.007	0.047	0.015	0.020	0.119
test err.	0.131	0.253	0.112	0.042	0.247	0.067	0.051	0.289	0.193	0.097	0.415
nr of cl.	10	10	10	10	10	10	10	10	10	10	10

random forests											
tra. err.	0.001	0.003	0.002	0.001	0.002	0.001	0.000	0.002	0.001	0.000	0.005
test err.	**0.119**	0.264	**0.104**	**0.034**	0.239	**0.060**	0.040	**0.269**	0.185	0.048	0.438
nr of cl.	10	10	10	10	10	10	10	10	10	10	10
feat. sel.	R	R	G	R	RG	RG	R	R	RG	R	

Ishibuchi-based fuzzy MCSs											
tra. err.	0.197	0.181	0.172	0.163	0.065	0.221	0.248	0.335	0.166	0.021	0.442
test err.	0.208	0.238	0.175	0.166	0.245	0.223	0.256	0.398	0.181	0.056	0.482
nr of cl.	3	7	7	10	0	10	7	3	7	10	7
feat. sel.	G	G	RG	RG	RG	G	RG	RG	RG	G	G

Table 38. A comparison of the best choice for different approaches for FURIA-based fuzzy MCSs against the best choice of bagging C4.5 MCSs, random forests, and Ishibuchi-based fuzzy MCSs in terms of *AUC*.

(a) First subset of datasets

FURIA-based MCSs										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
train AUC	0.693	0.993	0.895	0.927	0.999	0.991	0.802	0.724	0.950	0.999
test AUC	**0.548**	**0.970**	0.770	**0.785**	**0.875**	0.951	0.772	0.476	0.874	**0.991**
feat. sub. size	—	L	L	—	M	—	—	—	L	L
nr of cl.	7	7	7	7	7	10	5	3	7	10
feat. sel.	—	R	R	—	RG	—	—	—	R	R
approach	bag	FS	B.+FS	bag	FS	bag	bag	bag	FS	B.+FS

C4.5 ensembles with bagging										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
train AUC	0.866	0.990	0.925	0.931	0.986	0.990	0.912	0.757	0.948	0.997
test AUC	0.545	0.955	**0.771**	0.782	0.831	0.945	0.797	0.485	**0.876**	0.984
nr of cl.	10	7	7	7	7	10	7	3	3	7

random forests										
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
train AUC	0.999	0.998	0.998	0.962	0.997	1.000	0.994	0.930	0.994	1.000
test AUC	0.543	0.962	**0.771**	0.756	0.822	**0.957**	**0.809**	0.488	0.868	**0.991**
nr of cl.	10	5	7	3	5	10	7	3	10	10

Ishibuchi-based fuzzy MCSs										
AUC train	0.475	0.953	0.766	0.905	0.967	0.833	0.929	0.837	0.855	0.976
AUC test	0.487	0.939	0.684	0.783	0.822	0.737	0.779	**0.516**	0.683	0.965
nr of cl.	3	7	7	10	7	10	7	3	7	10
feat. sel.	R	R	G	R	RG	RG	R	R	RG	R

(b) Second subset of datasets

FURIA-based MCSs											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
train AUC	0.872	0.917	0.975	0.996	0.994	0.960	0.998	0.955	0.934	0.998	0.838
test AUC	0.804	**0.877**	0.907	**0.978**	**0.874**	0.915	**0.982**	0.765	**0.909**	**0.970**	0.705
feat. sub. size	L	S	—	—	M	—	—	—	S	M	
nr of cl.	7	7	10	10	7	7	10	10	10	10	10
feat. sel.	G	R	—	—	R	—	—	—	RG	R	
approach	B.+FS	FS	—	—	FS	—	—	—	FS	B.+FS	

C4.5 ensembles with bagging											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
train AUC	0.942	0.950	0.981	0.994	0.969	0.968	0.996	0.952	0.986	0.981	0.874
test AUC	0.833	0.748	0.906	0.973	0.752	**0.924**	0.973	0.761	0.866	0.903	**0.732**
nr of cl.	7	7	10	10	7	7	10	7	7	10	10

random forests											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
train AUC	0.997	0.997	0.999	1.000	0.998	0.998	1.000	0.998	0.998	1.000	0.996
test AUC	**0.843**	0.744	**0.912**	**0.978**	0.778	**0.924**	0.980	**0.775**	0.871	0.954	0.701
nr of cl.	7	7	10	10	7	7	10	10	7	10	10

Ishibuchi-based fuzzy MCSs											
	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
AUC train	0.801	0.567	0.912	0.847	0.790	0.997	0.820	0.827	0.688	0.974	0.693
AUC test	0.749	0.708	0.847	0.871	0.744	0.624	0.859	0.737	0.791	0.960	0.674
nr of cl.	7	7	7	10	10	5	10	10	7	10	7
feat. sel.	G	G	RG	RG	RG	G	RG	RG	RG	G	G

5.7. *Overall conclusions*

From the results obtained in the developed experiments we may conclude that the design of FURIA-based fuzzy MCSs is a competitive approach with respect to the classical state-of-the-art MCS design methods. Note that the same fuzzy MCS design methodology with a poor fuzzy classifier generation method does not provide good results. Hence, further research in this topic could lead to a promising methodology to design accurate fuzzy MCSs.

Basically, the global insights of our proposal are:

- A framework based on a quick and accurate fuzzy classification rule learning algorithm, namely FURIA, can be competitive if not better than two state-of-the-art machine learning classifier ensembles.
- The proposed FURIA-based fuzzy MCSs are *accurate* and can be directly applied on high dimensional datasets, high in terms of large number of attributes, number of instances, and/or number of classes, thanks to the fact we use FURIA as a component classifier.
- Due to the application of bagging to the MCSs, we obtained an approach being able to run the classifiers in parallel, thus being *time efficient.*
- FURIA-based fuzzy MCSs with bagging clearly outperform FURIA-based fuzzy MCSs with feature selection and FURIA-based fuzzy MCSs with bagging and feature selection. Thus, it is the recommended MCSs combination method.
- From the feature selection approaches Random-greedy turned out to be the best approach. This conclusion is not so clear, though. Notice that, considering FURIA-based fuzzy MCSs with bagging and feature selection average results for Greedy feature selection are not much worst than the ones with Random-greedy feature selection.
- Overall, it can be noticed that the larger the number of classifiers forming the fuzzy MCS, the lower the test error. Mostly, MCSs composed of 10 classifiers obtain the lowest test error, although in some cases MCSs composed of 7 classifiers outperformed the ones composed of 10.

6. Concluding Remarks

In this study, we proposed a methodology in which a bagging approach together with a feature selection technique is used to train FURIA-based fuzzy classifiers in order to obtain a fuzzy rule-based MCS. We used a single winner-based method on top of the base classifiers. This design allows our system to be both efficient by its inherent parallelism and accurate by the high quality of the base classifier when dealing with high dimensional datasets.

We tested FURIA-based fuzzy MCSs with bagging, feature selection, and the combination of both of them. By using the abovementioned techniques, we aimed to obtain fuzzy MCSs dealing with high dimensional data.

We have conducted comprehensive experiments over 21 datasets taken from the UCI machine learning repository. It turned out that FURIA-based fuzzy MCSs was the best performing approach from all the methods considered. Moreover, we showed that the obtained results are promising and provide a performance advantage in comparison with two state-of-the-art algorithms.

One of the next steps we will consider in the short future is to develop classifier selection using evolutionary multiobjective optimization algorithms to look for an optimal size of the ensemble. This MCS design approach, called overproduce-and-choose strategy (OCS)[35,40] is based on the generation of a large number of component classifiers and of the subsequent selection of the subset of them best cooperating. By doing so, the performance of FURIA-based fuzzy MCSs could be improved, while decreasing the number of classifiers in the ensemble, thus obtaining different trade-offs between accuracy and complexity.[47] The other extension to follow is to study alternative fuzzy reasoning methods to combine the results of the individual members of the ensemble, trying to combine classifiers in a dynamic manner,[40] in a way that a classifier or a set of them is responsible just for a particular data region.

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 Int. Conf. on Fuzzy Systems* (*FUZZ-IEEE*), London (UK), 2007, pp. 1794–1799.
2. R. E. Banfield, L. O. Hall, K. W. Bowyer and W. P. Kegelmeyer, A comparison of decision tree ensemble creation techniques, *IEEE Trans. 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 Trans. Neural Networks* **5**(4) (1994) 537–550.
4. P. P. Bonissone, J. M. Cadenas, M. C. Garrido and R. A. Díaz-Valladares, A fuzzy random forest, *International Journal of Approximate Reasoning* **51**(7) (2010) 729–747.
5. P. P. Bonissone, J. M. Cadenas, M. C. Garrido and R. A. Díaz-Valladares, A fuzzy random forest: Fundamental for design and construction, in *Proc. 12th Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems* (*IPMU'08*), Málaga (Spain), 2008, pp. 1231–1238.
6. A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition* **30** (1997) 1145–1159.
7. L. Breiman, Bagging predictors, *Machine Learning* **24**(2) (1996) 123–140.
8. L. Breiman, Random forests, *Machine Learning* **45**(1) (2001) 5–32.
9. J. Canul-Reich, L. Shoemaker, and L.O. Hall, Ensembles of fuzzy classifiers, in *IEEE Int. Conf. Fuzzy Systems* (*FUZZ-IEEE*), London (UK), 2007, pp. 1–6.
10. W. W. Cohen, Fast effective rule induction, in *Proc. Twelfth Int. Conf. Machine Learning* (Morgan Kaufmann, 1995), pp. 115–123.
11. O. Cordón, M. J. del Jesus and F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, *Int. J. Approximate Reasoning* **20** (1999) 21–45.
12. O. Cordón and A. Quirin, Comparing two genetic overproduce-and-choose strategies for fuzzy rule-based multiclassification systems generated by bagging and mutual

information-based feature selection, *Int. J. Hybrid Intelligent Systems* **7**(1) (2010) 45–64.

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 *Third Int. Workshop on Genetic and Evolving Fuzzy Systems* (*GEFS*), Witten-Bommerholz (Germany), 2008, pp. 11–16.
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 *Int. Conf. Hybrid Intelligent Systems* (*HIS*), Barcelona (Spain), 2008, pp. 549–554.
15. M. J. del Jesus, F. Hoffmann, L. J. Navascues and L. Sánchez, Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms, *IEEE Trans. Fuzzy Systems* **12**(3) (2004) 296–308.
16. T. G. Dietterich, Approximate statistical test for comparing supervised classification learning algorithms, *Neural Computation* **10**(7) (1998) 1895–1923.
17. T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, *Machine Learning* **40**(2) (2000) 139–157.
18. T. A. Feo and M. G. C. Resende, Greedy randomized adaptive search procedures, *J. Global Optimization* **6** (1995) 109–133.
19. J. Gentle, W. Härdle and Y. Mori, Handbook of computational statistics: Concepts and methods, *Metrika* **67**(2) (2008) 245–246.
20. T. Ho, The random subspace method for constructing decision forests, *IEEE Trans. Pattern Analysis and Machine Intelligence* **20**(8) (1998) 832–844.
21. F. Hoffmann, Combining boosting and evolutionary algorithms for learning of fuzzy classification rules, *Fuzzy Sets and Systems* **141**(1) (2004) 47–58.
22. J. Huang and C. X. Ling, Using AUC and accuracy in evaluating learning algorithms, *IEEE Trans. Knowledge and Data Engineering* **17** (2005) 299–310.
23. J. C. Hühn and E. Hüllermeier, FURIA: an algorithm for unordered fuzzy rule induction, *Data Mining and Knowledge Discovery* **19**(3) (2009) 293–319.
24. J. C. Hühn and E. Hüllermeier, An analysis of the FURIA algorithm for fuzzy rule induction, in *Advances in Machine Learning I*, 2010, pp. 321–344.
25. H. Ishibuchi, T. Nakashima and T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, *Fuzzy Sets and Systems* **103**(2) (1999) 223–238.
26. H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining* (*Advanced Information Processing*) (Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2004).
27. H. Ishibuchi and Y. Nojima, Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers, *Int. J. Hybrid Intelligent Systems* **3**(3) (2006) 129–145.
28. C. Z. Janikow, Fuzzy decision trees: issues and methods, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **28**(1) (1998) 1–14.
29. L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms* (Wiley, 2004).
30. C. Marsala, Data mining with ensembles of fuzzy decision trees, in *IEEE Symp. Computational Intelligence and Data Mining*, Nashville (USA), 2009, pp. 348–354.
31. Y. Nojima and H. Ishibuchi, Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion, in *Int. Conf. Hybrid Intelligent Systems and Conference on Neuro-Computing and Evolving Intelligence*, Auckland, New Zealand, 2006.

32. Y. Nojima and H. Ishibuchi, Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design, *Int. J. Hybrid Intelligent Systems* **4**(3) (2007) 157–169.
33. D. Optiz and R. Maclin, Popular ensemble methods: an empirical study, *J. Artificial Intelligence Res.* **11** (1999) 169–198.
34. P. Panov and S. Džeroski, Combining bagging and random subspaces to create better ensembles, in *IDA'07: Proc. 7th Int. Conf. Intelligent Data Analysis*, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 118–129.
35. D. Partridge and W. B. Yates, Engineering multiversion neural-net systems, *Neural Computation* **8**(4) (1996) 869–893.
36. 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.
37. W. Pedrycz and K.C. Kwak, Boosting of granular models, *Fuzzy Sets and Systems* **157**(22) (2006) 2934–2953.
38. J. R. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993).
39. L. Sánchez and J. Otero, Boosting fuzzy rules in classification problems under single-winner inference, *Int. J. Intelligent Systems* **22**(9) (2007) 1021–1034.
40. 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.
41. R. Schapire, The strength of weak learnability, *Machine Learning* **5**(2) (1990) 197–227.
42. R. Scherer, Boosting ensemble of relational neuro-fuzzy systems, *Lecture Notes in Computer Science* **4029** (2006) 306–313.
43. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication* (University of Illlinois Press, 1949).
44. J. Stefanowski, An experimental study of methods combining multiple classifiers — diversified both by feature selection and bootstrap sampling, in K. T. Atanassov, J. Kacprzyk, M. Krawczak, and E. Szmidt (eds.), *Issues in the Representation and Processing of Uncertain and Imprecise Information*, Akademicka Oficyna Wydawnicza EXIT, Warsaw, 2005, pp. 337–354.
45. H. Takahashi and H. Honda, A new reliable cancer diagnosis method using boosted fuzzy classifier with a SWEEP operator method, *J. Chem. Eng. Jpn.* **38**(9) (2005) 763–773.
46. H. Takahashi and H. Honda, Lymphoma prognostication from expression profiling using a combination method of boosting and projective adaptive resonance theory, *J. Chem. Eng. Jpn.* **39**(7) (2006) 767–771.
47. K. Trawiński, A. Quirin and O. Cordón, A study on the use of multi-objective genetic algorithms for classifier selection in furia-based fuzzy multiclassifers, *Int. J. Computational Intelligence Systems*, in press.
48. K. Trawiński, A. Quirin and O. Cordón, Bi-criteria genetic selection of bagging fuzzy rule-based multiclassification systems, in *IFSA World Congress-EUSFLAT Conference* (*IFSA/EUSFLAT*), Lisbon (Portugal), 2009, pp. 1514–1519.
49. K. Trawiński, A. Quirin and O. Cordón, On the combination of accuracy and diversity measures for genetic selection of bagging fuzzy rule-based multiclassification systems, in *Int. Conf. Intelligent Systems Design and Applications (ISDA)*, Pisa (Italy), IEEE Computer Society, 2009, pp. 121–127.

50. 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, *Math. Biosci.* **193**(1) (2005) 79–100.
51. A. Tsymbal, M. Pechenizkiy and P. Cunningham, Diversity in search strategies for ensemble feature selection, *Information Fusion* **6**(1) (2005) 83–98.
52. 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.
53. 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.
54. Z. H. Zhou, Ensembling local learners through multimodal perturbation. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics* **35**(4) (2005) 725–735.