Departamento de Informática

UNIVERSIDAD DE OVIEDO

Tesis Doctoral

Design of Fuzzy Rule-based Ensembles using FURIA, Diversity Induction and Evolutionary Algorithms

Krzysztof Trawiński

Universidad de Oviedo

Design of Fuzzy Rule-based Ensembles using FURIA, Diversity Induction and Evolutionary Algorithms

Memoria que presenta

Krzysztof Trawiński

Para optar al grado de Doctor por la Universidad de Oviedo

Octubre de 2013

Directores

Oscar Cordón García
Arnaud Quirin
Luciano Sánchez Ramos

Departamento de Informática

Acknowledgement

First of all, I would like to point out that this work is supported by the Spanish *Ministerio de Economía y Competividad* under the project TIN2012-38525-C02-01.

I would like to truthfully express gratitude to my supervisors, Óscar Cordón, Arnaud Quirin and Luciano Sánchez, for simply letting me work with them and write this PhD dissertation. Their kindness and friendship taughed me that there are people with a great value. Their motivation, work ethics and passion for the research were really inspiring and helped me to finish this work. Many thanks for supporting me, I know I was not a good student all the time.

Then, I would like to thank all the "AFE" members for having a nice time during the working hours in our "cave".

Moreover, I would like to thank all the former and current members of the European Centre for Soft Computing. I am not able to list all your names here due to the lack of space, but those who knows me and became my friends are in this list.

Besides, I neeed to express gratitude to my climbing team. Thanks to you I spent many hours in the beautiful mountains of Asturias.

I am very greatful to Maria for showing me a big support and great conversations during the "coke-breaks" in Mieres.

I cannot forget about Laura, who was my main reason for coming to Spain. I wish you would finish your PhD dissertation soon.

Finally, I would like to thank all my family. Especially, I am very grateful to my parents, Helena and Bogdan, and my siblings, Karolina and Grzegorz. Thanks for the big support and accepting me as I am. Without you probably I would not be here.

Contents

Resumen

Los sistemas basados en reglas difusas han demostrado una alta capacidad de extracción y representación del conocimiento a la hora de modelar problemas de clasificación complejos y no lineales. Sin embargo, cuando se aplican a conjuntos de datos de alta complejidad, es decir con un gran número de variables y/o ejemplos, sufren la denominada "maldición de las dimensiones" (curse of dimensionality). Los métodos de combinación de clasificadores han demostrado ser una buena técnica para afrontar este tipo de problemas.

En esta tesis doctoral se propone un marco global basado en el enfoque de los métodos de combinación de clasificadores que permite a los sistemas basados en reglas difusas manejar conjuntos de datos de alta complejidad evitando el problema anterior. Para conseguir afrontar este objetivo, el marco de trabajo propuesto incorpora distintos métodos de combinación de clasificadores y considera algoritmos evolutivos para diseñar métodos de combinación de clasificadores basados en reglas difusas. Su estructura se basa en dos etapas: 1) Diseño de métodos de combinación de clasificadores basados en reglas difusas a partir de enfoques clásicos y avanzados, y 2) Diseño de nuevos métodos de selección y fusión de clasificadores base usando algoritmos evolutivos. Este enfoque permite diseñar varios métodos específicos de combinación de clasificadores basados en reglas difusas que permiten la mejora de la precisión en los resultados y la obtención de un buen equilibrio entre precisión y complejidad. Se han realizado experimentos exhaustivos con varios conjuntos de datos de alta complejidad (en lo que respecta al número de atributos y al número de ejemplos) procedentes de los repositorios UCI y KEEL que han demostrado el buen comportamiento de los métodos propuestos.

Además, se ha aplicado con éxito uno de los diseños concretos de combinación de clasificadores basados en reglas difusas a un problema real consistente en la localización en interiores utilizando topología WiFi. Esta tarea se corresponde con un problema de clasificación de alta dimensionalidad cuando se trata de un entorno complejo, que presenta la dificultad adicional de la incertidumbre asociada debido a la naturaleza de las señales WiFi.

Abstract

Fuzzy rule-based systems have shown a high capability of knowledge extraction and representation when modeling complex, non-linear classification problems. However, they suffer from the so-called curse of dimensionality when applied to high complexity datasets, which consist of a large number of variables and/or examples. Classifier ensembles have shown to be a good approach to deal with this kinds of problems.

In this PhD dissertation, we propose a classifier ensemble-based global framework allowing fuzzy rule-based systems to deal with high dimensional datasets avoiding the curse of dimensionality. Having this goal in mind, the proposed framework incorporates several classifier ensemble methodologies as well as evolutionary algorithms to design fuzzy rule-based classifier ensembles. The proposed framework follows a two-stage structure: 1) fuzzy rule-based classifier ensemble design from classical and advanced classifier ensemble design approaches, and 2) novel designs of evolutionary component classifier combination. By using our methodology, different fuzzy rule-based classifier ensembles can be designed dealing with several aspects such as the improvement of the performance in terms of accuracy and the obtaining a good accuracy-complexity trade-off. Exhaustive experiments carried out over several UCI and KEEL datasets with high complexity (considering both the number of attributes as well as the number of examples) have shown the good performance of the proposed classifier ensemble-based global framework.

Besides, one of the specific fuzzy rule-based classifier ensemble design approaches obtained from the proposed framework has been successfully applied to a real-world problem. It consists of topology-based WiFi indoor localization, which turns into a high dimensional classification problem when dealing with a complex environment. The complexity of this task is also characterized by the huge inherent uncertainty coming from the nature of WiFi signals.

Part I. Report

1 Statement

The aim of this section is to bring the main aspects of the current doctoral dissertation into context. Firstly, a global introduction to the general topic is provided. Then, a list of open problems being the rationale of the work to be developed is presented. The objectives to be achieved during the development of the PhD dissertation are later shown. Finally, the global structure of the dissertation is reported.

1.1 Introduction

Classifier ensembles (CEs), also called multiclassification systems, are machine learning tools capable to obtain better performance than a single classifier when dealing with complex classification problems [Kun04]. These kinds of systems are especially useful when the number of dimensions or the size of the data are really large. The most common base classifiers are decision trees [Ho98] and neural networks [OM99]. More recently, the use of fuzzy classifiers has also been considered [CRSH07, PK06, CQS08, CQ10].

Meanwhile, fuzzy rule-based classification systems (FRBCSs) have shown a high capability of knowledge extraction and representation when modeling complex, non-linear classification problems. To do so, they consider soft boundaries obtained through the use of a collection of fuzzy rules that could be understood by a human being [Kun00, INN05]. Interpretability of fuzzy systems is a characteristic that definitely favors these types of models, as it is often a need to understand the behavior of the given model [CCHE03, AMGR09, AM10].

FRBCSs, however, have one significant drawback. The main difficulty appears when it comes to deal with a dataset consisting of a high number of variables and/or examples. In such a case FRBCS suffers from the so-called *curse of dimensionality* [INN05]. It occurs due to the exponential increase of the number of rules and the number of antecedents within a rule with the growth of the number of inputs of the FRBCS. This issue also causes a scalability problem in terms of the required run time and the memory consumption and of course makes the FRBCS loose its capacity to be interpreted by a human being.

The main objective of the current PhD dissertation is to propose a CE-based global framework allowing FRBCSs to deal with high dimensional datasets avoiding the curse of dimensionality. With this aim, this framework will incorporate several CE design methodologies as well as evolutionary algorithms to generate fuzzy rule-based classifier ensembles (FRBCEs). The proposed framework follows a two-stage structure: 1) FRBCE design from classical and advanced CE design approaches, and 2) Novel methods for evolutionary component classifier combination. These FRBCEs will consider several aspects such as improving the performance in terms of accuracy and obtaining a good complexity-accuracy trade-off.

1.2 Justification

As already mentioned, the main disadvantage of FRBCSs is the well known curse of dimensionality. This phenomenon is always present when applying such systems to problems involving a large number of variables [INN05]. It occurs due to the exponential increase of the number of rules and the number of antecedents within a rule with the growth of the number of inputs of the FRBCS. Dealing with this issue when designing FRBCSs is the main motivation for the current PhD dissertation.

Furthermore, we can provide some secondary open problems related to the latter in the following points:

- No global methodology for FRBCS design has been proposed that might be applied with any FRBCS design method in order to deal with the curse of dimensionality while obtaining a good accuracy.
- The literature regarding FRBCEs is not so extensive. CEs usually consider classical machine learning algorithms as a base classifier, such as decision trees or neural networks. However, fuzzy classifiers have proven to be competitive with other kinds of pattern recognition approaches [HH09, HH10]. Hence, we aim to design FRBCEs being competitive or even outperforming classical CEs in terms of accuracy.
- CEs can significantly increase their size when trying to improve their accuracy. Especially, that is the case when dealing with high dimensional datasets. Thus, looking for the most appropriate complexity-accuracy trade-off in CEs has become a crucial topic in the literature. Multiobjective optimization (MO) [CH83] could be a good approach to deal with this problem as often complexity and accuracy are conflicting objectives to be optimized. In addition, the influence of the classifier diversity on the final CE performance is still not clear [TPC05, RG05, KW03]. Thus, using a measure of this kind as one of the optimization criteria may lead to a performance improvement.
- Up to our knowledge, no work has been done regarding the "interpretability" of CEs. These systems are complex and difficult to analyze. Thus, having some linguistic insight of the ensemble's operation mode is a very challenging task that can also lead to improved CE designs.

1.3 Objectives

The main objective of this PhD dissertation is to propose and evaluate a global framework for FRBCE design dealing with high dimensional and complex datasets. Specifically, this overall goal can be divided into the following specific objectives:

- To propose a methodology to design FRBCEs using classical CE methods (i.e. bagging [Bre96] and feature selection [Ho98]) as well as advanced techniques based on diversity induction (i.e. random oracles (ROs) [KR07, RK07]). We aim to apply the Fuzzy Unordered Rules Induction Algorithm (FURIA) [HH09, HH10] to derive FRBCSs to be considered as base classifiers in the CE.

- To integrate the abovementioned approaches with a multibojective overproduce-and-choose strategy (OCS) for component classifier selection. We aim to exploit a state-of-the-art evolutionary multiobjective algorithm, namely NSGA-II [DPAM02], to perform component classifier selection and improve the designed FURIA-based FRBCEs. Our idea is to propose several multicriteria fitness functions based on three different families of optimization criteria: accuracy, complexity, and diversity [TPC05]. Thanks to the multiobjective approach, a Pareto set of FRBCE designs could be obtained with different trade-offs among the selected objectives.

- To propose a fuzzy system-based mechanism to combine the component classifiers with the aim of improving the performance of FRBCEs, following an approach globally called stacking [Wol92] in the existing literature. We aim to design a novel fuzzy linguistic combination method to perform joint classifier fusion and classifier selection at class level. By using a linguistic FRBCS as a combination method, the resulting CE would show a hierarchical structure and its operation would be transparent to the user.

- To assess the proposed FRBCE design methods and compare them with state-of-the-art approaches. We aim to analyze all of our FRBCE designs considering performance in terms of both accuracy and complexity. For that purpose, we will use standard, complex, and high dimensional datasets from the UCI machine learning [BM98] and the KEEL [AFFL+11] repositories.

- To validate the FRBCE design methodology on a high dimensional real-world problem. We aim to solve a topology-based WiFi indoor localization problem [AOS+09] by means of one of the FRBCEs designed to show the actual efficacy of our methods.

1.4 Structure

This PhD dissertation is divided into two parts. The first one is dedicated to the statement of the problem, the revision of the current state of the art, the development of our FRBCE design methodology, the discussion of the results obtained, and the presentation of the lines for future work. The second one collects the scientific publications obtained as a result of the study developed.

Part I is organized as follows. After the introduction to the problem, we present a set of open issues justifying this PhD dissertation, as well as its main objectives. Then, in Section 2 we review the basic concepts about CEs, fuzzy CEs, and the algorithms that will be used as a base for our proposal. In Section 3 we introduce the proposed global framework for designing FRBCEs, describing each method in details, while Section 4 discusses the results obtained. Finally, Section 5 shows the future research lines raised from our research work.

The work developed to achieve the stated objectives is described in the five scientific publications composing Part II of this PhD dissertation:

- K. Trawiński, O. Cordón, and A. Quirin. On Designing Fuzzy Rule-based Multiclassification Systems by Combining FURIA with Bagging and Feature Selection, International Journal of Uncertainty Fuzziness, and Knowledge-based Systems, vol. 19, no 4, pp. 589-633, 2011. DOI: 10.1142/S0218488511007155. Impact factor: 1.781. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 31/111. Q2.

- K. Trawiński, O. Cordón, and A. Quirin. A Study on the Use of Multiobjetive Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers, International

Journal of Computational Intelligence Systems, vol. 4, no 2, pp. 231-253, 2012. DOI: 10.1080/18756891.2012.685272

- K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez. A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers, IEEE Transactions of Fuzzy Systems, vol. 21, no 5, pp. 950-965, 2013, 2013. DOI: 10.1109/TFUZZ.2012.2236844. Impact Factor 2012: 5.484. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 1/115. Q1.

- K. Trawiński, O. Cordón, L. Sánchez, and A. Quirin. Multiobjective Genetic Classifier Selection for Random Oracles Fuzzy Rule-Based Multiclassifiers: How Beneficial is the Additional Diversity?, Knowledge-based Systems, In press, 2013. DOI: 10.1016/j.knosys.2013.08.006. Impact factor 2012: 4.104. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 6/115. Q1.

- K. Trawiński, J. M. Alonso, and N. Hernandez. A Multiclassifier Approach for Topology-based WiFi Indoor Localization, Soft Computing, vol. 17, no 10, pp. 1817-1831, 2013. DOI 10.1007/s00500-013-1019-5. Impact factor 2012: 1.124. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 63/115. Q3.

2 State of the Art

In this section we briefly review CEs and fuzzy CEs. We also recall the basic aspects of FURIA, a novel and good performing fuzzy rule-based classifier, which will be used as the component base classifier. Finally, we briefly describe genetic fuzzy systems, which will be a fundamental tool for development of the component fuzzy classifier combination method presented in this PhD dissertation.

2.1 Classifier Ensembles

CE design is mainly based on two stages [DS79]: the learning of the component classifiers and the combination mechanism for the individual decisions provided by them into the global CE output. Since a CE is the result of the combination of the outputs of a group of individually trained classifiers, the accuracy of the finally derived CE relies on the performance and the proper integration of these two tasks. The best possible situation for an ensemble is that where the individual classifiers are both accurate and fully complementary, in the sense that they make their errors on different parts of the problem space [OM99]. Hence, CEs rely for their effectiveness on the "instability" of the base learning algorithm.

On the one hand, the correct definition of the set of base classifiers is fundamental to the overall performance of CEs. Different approaches have been thus proposed to succeed on generating diverse component classifiers with uncorrelated errors such as data resampling techniques (mainly, bagging [Bre96] and boosting [Sch90]), specific diversity induction mechanisms (feature selection [Ho98], diversity measures [TPC05], use of different parameterizations of the learning algorithm, use of different learning models, etc.), or combination between the latter two families, as the well known random forests approach [Bre01].

On the other hand, the research area of combination methods is also very active due to the influential role of this CE component. It does not only consider the issue of aggregating the results provided by all the initial set of component classifiers derived from the first learning stage to compute the final output (what is usually called *classifier fusion* [XKS92, WKB97]). It also involves either locally selecting the best single classifier which will be taken into account to provide a decision for each specific input pattern (static or dynamic classifier selection [GR01]) or globally selecting the subgroup of classifiers which will be considered for every input pattern (overproduce-and-choose strategy [PY96]). Besides, hybrid strategies between the two groups have also been introduced [Kun04]. In any case, the determination of the optimal size of the ensemble is an important issue for obtaining both the best possible accuracy in the test data set without overfitting it, and a good accuracy-complexity trade-off [HLnS13].

2.1.1 Classifier Ensemble Design Methodologies

A CE 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 [Kun04]. 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 [Die00, BHBK07], although they can be used with any other type of classifiers (the use of neural networks is also very extended, see for example [OM99]).

There are different ways to design a CE. 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* [Bre96], they are independently learnt from resampled training sets ("bags"), which are randomly selected with replacement from the original training data set. *Boosting* methods [Sch90] 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 [1].

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 [Zho05]. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features [TPC05, XKS92]. *Random subspace* (RS) [Ho98], where each feature subset is randomly generated, is one of the most representative methods of this kind.

Further, ROs is a CE approach achieving good performance while having several interesting features (a comprehensive study is presented in [KR07, RK07]). It is based on the use of a miniensemble replacing the component base classifier which is composed of a pair of subclassifiers with a RO (a random function, e.g. a random hyperplane) choosing between the two of them (dynamic classifier selection), when an instance is presented in the input. During the training phase, RO splits a dataset into two parts and feeds each subclassifier with the data from each half-space, while during the classification phase it decides which subclassifier makes the final decision to be further used at the ensemble level.

Finally, there are some advanced proposals that can be considered as a combination of the two groups, such as *random forests* [Bre01] and more recently *rotation forest* [RKA06] and *fuzzy random forest* [BCGDV10].

The interested reader is referred to [BHBK07, OM99] for two reviews for the case of decision tree (both) and neural network ensembles (the latter), including exhaustive experimental studies.

[1] Both acquired some derivatives e.g. wagging [BK99], AdaBoost.M1 [FS97], and AdaBoost.M2 [SS99], respectively.

2.1.2 Classifier Ensemble Combination Methods

Two main approaches arise in the literature for the combination of the outputs provided by a previously generated set of individual classifiers into a single CE output [WKB97]: *classifier fusion* and *classifier selection*.

Classifier fusion relies on the assumption that all ensemble members make independent errors. Thus, combining the decisions of the ensemble members may lead to increasing the overall performance of the system. Majority voting, sum, product, maximum and minimum are examples of functions used to combine their decisions [KHDM98]. However, these family of methods carry the drawback that there is no guarantee that a particular ensemble generation technique will achieve the error independence. Thus, it could happen that the combination of the component classifiers' decisions does not improve the final classification performance. That is the reason for the extended use of weighted majority voting, which allows to weight the contribution of each individual classifier to the final decision according to its "classification competence" using coefficients of importance [BC94, LS97]. There are many different kinds of strategies to determine these combination weights, with genetic algorithms (GAs) [Gol89] being extensively used [Kun01a, KMH06].

Alternatively, classifier selection is based on the fact that not all the individual classifiers but only a subset of them will influence on the final decision for each input pattern. There are different families within this group according to the locality/globality nature and the timing of this decision within the CE learning process pipeline. On the one hand, a general family of classifier selection methods assumes that each individual classifier is an expert in some local regions of the problem space [ZWY04], thereby avoiding the error independence assumption. In this approach, the accuracy of each classifier surrounding the region of the feature space where the unknown pattern to be classified is located is previously estimated, and the best one is selected to classify that specific pattern. Two categories of classifier selection techniques exist: static and dynamic [WKB97, GR01]. In the former, regions of competence are defined during the training phase, while in the latter, they are defined during the classification phase taking into account the characteristics of the sample to be classified. Dos Santos et al. compile an extensive list of dynamic classifier selection methods [DSM08], reporting their main characteristics. Nevertheless, there is a drawback to both selection strategies: when the local expert does not classify the test pattern correctly, there is no way to avoid the misclassification.

On the other hand, there is another family of static classifier selection methods based on the assumption that candidate classifiers could be redundant because of the difficulty found by the base learning method to generate actually uncorrelated individual classifiers. In [ZWT02], Zhou et al. formally showed that finding the most relevant subset of classifiers is more effective in terms of performance than combining all the available classifiers (i.e, than direct classifier fusion). These methods are grouped under the name of overproduce-and-choose strategy (OCS) [PY96] (also known as test-and-select methodology [SS00]). They are based on the fact that a large set of candidate classifiers is generated and then selected to extract the best performing subset (removing duplicates and poor-performing candidate classifiers) which composes the final CE used to classify the whole test set. Diversity measures, accuracy, and ensemble size are frequently employed as search criteria to determine the selected component classifiers and GAs are commonly used for that task [RG05, DSM06, CQ10]. Consequently, OCS methods determine the optimal ensemble size by considering a trade-off between accuracy and complexity. However, OCS could be subject to overfitting, as a fixed subset of classifiers defined using a training/optimization data set may not be well adapted for the classification of every pattern in the test set [DSM08].

In order to overcome the problems of each family, hybrid methods between the latter families have been proposed. That is the case of [DSM08] where a dynamic OCS procedure is introduced combining a GA for static classifier selection and a dynamic local selection into a two-level selection phase. On the other hand, Gabrys and Ruta [GR06] developed a multidimensional GA to optimize two weight-based models, in which the weights are assigned to each classifier or to each class. Besides, in [DVA09], the authors proposed a GA selecting the *votes* of each classifier in an ensemble for its reliability to classify each class, instead of discarding the classifiers at a whole.

2.1.3 Classifier Ensemble Fuzzy Combination Methods

Fuzzy set theory has been extensively and successfully considered for CE combination, especially classifier fusion. As mentioned in several papers like [AC03], the latter is a consequence of the different advantages the use of fuzzy aggregation operators present, mainly their capability to model the imprecision and uncertainty involved in the CE combination process. The use of fuzzy connectives to combine the outputs of the component classifiers of an ensemble was first proposed in [CK95]. Since then, two different groups of fuzzy operators have been considered in the specialized literature [Kun01a]:

1. *the classical simple fuzzy aggregation operators*, such as minimum, maximum, simple average, or product.
2. *more advanced fuzzy operators*, including the fuzzy integral [Gra95], the BADD defuzzification strategy [FY89], Zimmermann's compensatory operator [ZZ80], and other fuzzy combination operators specifically designed for this task, such as the decision templates model [KBD01].

Some studies have developed experimental comparisons of the performance of different CEs considering the latter fuzzy connectives as fusion combination operator [VLM+99, Kun02]. Additionally, in [Kun03] their accuracy was compared to that of seven of the usual crisp (i.e., non-fuzzy) aggregation operators when considered as combination operators for Boosting CEs. The conclusions drawn from that experimentation were that fuzzy combination methods outperformed non-fuzzy ones, and that decision templates based on Euclidean distance and fuzzy integral were the best methods overall.

Besides, some other works have extended the scope of the latter ones. In [AC03], the authors focused on analyzing the influence of the choice of the ensemble members (i.e., the impact of the ensemble sizes and the type of base classifiers considered) in the accuracy of the combination methods considered. They concluded that fuzzy methods delivered higher accuracy and lower dependency to the choice of the ensemble members than non-fuzzy methods. On the other hand, Bulacio et al. [BGTM10] introduced a hybrid classifier selection-fusion strategy, considering Sugeno's fuzzy integral [Sug74] as combination method and a greedy heuristic for the ensemble member selection.

Lu and Yamaoka [LY97] introduced a fuzzy combination method specifically designed for a hybrid ensemble of three classifiers which shows the novel characteristic of allowing the user to incorporate human expert knowledge on the bias of the component classifiers. This is done by means of an additional refinement module based on a fuzzy rule-based system (FRBS) comprised by Mamdani-type fuzzy rules. In this way, Lu and Yamaoka's fuzzy combination method does not make use of fuzzy rules but of a complex fuzzy reasoning process where the following components are considered: a linguistic partition for the ensemble members' outputs, a fuzzy aggregation of their membership degrees and a defuzzification method to modify them, and a new (crisp) aggregation for each class in order to take the final CE decision corresponding to the largest aggregated class membership value.

As said, the latter procedure can be complemented by expert-defined fuzzy rules to adjust the importance of the decisions taken for each class according to the nature of the component classifiers. Hence, an FRBS is used as a refinement module for the fuzzy combination method decisions. Nevertheless, this strategy shows several problems such as its specificity to the consideration of a simple three-CE, its highly complex structure composed of two different nature fuzzy reasoning modules, the need of manually defining the fuzzy rules in the refinement module [2], and the impossibility to perform classifier selection (which of course is not required in the simple ensemble structure considered).

Finally, an interesting and very recent approach for generating FRBSs that combine ensembles is presented in [TG13]. The authors use a context-free grammar within a hybrid genetic programming using a multi-population model to evolve the fuzzy rule base and the composition of the ensembles over time.

2.2 FURIA

Fuzzy Unordered Rules Induction Algorithm (FURIA) [HH09, HH10] is an extension of the state-of-the-art rule learning algorithm called RIPPER [Coh95], considering the derivation of simple and comprehensible fuzzy rule bases, and introducing some new features. FURIA provides three different extensions of RIPPER:

- It takes an advantage of fuzzy rules instead of crisp ones. Fuzzy rules of FURIA are composed of a class C_j and a certainty degree CD_j in the consequent. The final form of a rule is the following:

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

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

  $$CD_j = \frac{2\frac{D_T^{C_j}}{D_T} + \sum_{x \in D_T^{C_j}} \mu_r^{C_j}(x)}{2 + \sum_{x \in D_T} \mu_r^{C_j}(x)} \tag{I.1}$$

  where D_T and $D_T^{C_j}$ stands for the training set and a subset of the training set belonging to the class C_j respectively. 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. Hence, the fuzzy reasoning method used is the so-called voting-based method [INM99, CdJH99].

- It uses unordered rule sets instead of rule lists. This change omits a bias caused by the default class rule, which is applied whenever there is an uncovered example detected.

- It proposes a novel rule stretching method in order to manage uncovered examples. The unordered rule set introduces one crucial drawback, there might appear a case when a given example is not covered. Then, to deal with such situation, one rule is generalized by removing its antecedents. The information measure is proposed to verify which rule to "stretch".

The interested reader is referred to [HH09] for a full description of FURIA.

[2] This could be feasible when using a very small number of component classifiers –only three– but not with dealing with a more usual larger number. In fact, the FRBSs considered in their experimentation are only composed of a single rule with three inputs as well as the authors mention they were not able to incorporate expert knowledge to the Bayesian component classifier.

2.3 Genetic Fuzzy Systems

Fuzzy systems, which are based on fuzzy logic, became popular in the research community, since they have ability to deal with complex, non-linear problems being too difficult for the classical methods [YF94]. Besides, its capability of knowledge extraction and representation allowed them to become human-comprehensible to some extent (more than classical black-box models) [CCHE03, AMGR09].

The lack of automatic extraction processes in fuzzy systems attracted the attention of the computational intelligence community to incorporate learning capabilities to these kinds of systems. In consequence, a hybridization of fuzzy systems and GAs became one of the most popular approaches in this field [CHHM01, CGH$^+$04, Her08, Cor11]. In general, genetic fuzzy systems (GFSs) are fuzzy systems enhanced by a learning procedure coming from evolutionary computation, i.e. considering any evolutionary algorithm (EA).

FRBSs, which are based on fuzzy "IF-THEN" rules, constitute one of the most important areas of fuzzy logic applications. Designing FRBSs might be seen as a search problem in a solution space of different candidate models by encoding the model into the chromosome, as GAs [Gol89] are well known optimization algorithms capable of searching among large spaces with the aim of finding optimal (usually nearly optimal) solutions.

The generic coding of GAs provides them with a large flexibility to define which parameters/components of FRBS are to be designed [Her08]. For example, the simplest case would be a parameter optimization of the fuzzy membership functions. The complete fuzzy rule base can also be learned. This capability allowed the field of GFSs to grow over two decades and to still be one of the most important topics in computational intelligence.

In the current PhD dissertation, we will relay on the GFS paradigm to define some of the proposed FRBCE designs.

2.4 Related Work on Fuzzy Classifier Ensembles and Fuzzy Fusion Methods

Focusing on fuzzy CEs, only a few contributions for bagging fuzzy classifiers have been proposed considering fuzzy neural networks (together with feature selection) [TH06], neuro-fuzzy systems [CRSH07], and fuzzy decision trees [BCGDV10, Mar09] as component classifier structures.

Especially worth mentioning is the contribution of Bonissone et al. [BCGDV10]. This approach hybridizes Breiman's idea of random forests [Bre01] with fuzzy decision trees [Jan98]. Such resulting fuzzy random forest combines characteristics of CEs 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.

Some advanced contributions based on GFSs should also be remarked. On the one hand, an FRBCS ensemble design technique is proposed in [ACdJH07] considering some niching GA to develop feature selection in order 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 [NI06, NI07]. 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 [IN06], where an evolutionary multiobjective optimization (EMO) algorithm [CLV07] generated a Pareto set of FRBCSs with different accuracy-complexity trade-offs to be combined into an ensemble.

3 Development

This section proposes the global framework for FRBCEs design that has been developed in this PhD dissertation to deal with high dimensional and complex datasets. In addition, it also describes the use of the general framework to create some specific FRBCE generation methods obtaining good performance in terms of accuracy and complexity.

3.1 Proposed Methodology

The main objective of this dissertation is to enable FRBCSs to deal with high dimensional datasets by means of different CE approaches. Thus, we sketched a global framework allowing for several FRBCEs specific designs. This framework is composed of two stages (see Fig. 1). The first one, called “component fuzzy classifier design from classical ML approaches”, includes the use of FURIA to derive the component classifiers considering classical CE design approaches (see Sec. 2.1.1) such as:

- *Static* approaches. From this family we incorporate bagging, feature selection, and the combination of bagging and feature selection. Thanks to the intrinsic parallelism of bagging, the design procedures will also be time efficient.
- *Dynamic* approaches. From this family we employ the combination of bagging and ROs, since ROs induce an additional diversity to the base classifiers, the accuracy of the final FRBCEs is thus improved.

In [HLnS13], a study to determine the size of a parallel ensemble (e.g. bagging) by estimating the minimum number of classifiers that are required to obtain stable aggregate predictions was shown. The conclusion drawn was that the optimal ensemble size is very sensitive to the particular classification problem considered. Thus, the second stage of our framework, called “Evolutionary component classifier combination”, is related to post-processing of the generated ensemble by means of EAs to perform component classifier combination. All the approaches used consider classifier selection and some of them also combine it with classifier fusion.

Of course, the second stage follows the approaches from the first stage. This is indicated by red arrows in the figure, showing exactly which approach is used for the FRBCE design (Stage 1) together with its corresponding evolutionary post-processing (Stage 2). A dashed red arrow points out a proposal that, although is included in the general framework, was not developed in the current PhD dissertation and is left for future works.

The second stage includes the following evolutionary component classifier selection designs:

- *Classifier selection.* Within this family, we opted for an EMO OCS strategy (see Sec. 2.1.2), using the state-of-the-art NSGA-II algorithm [DPAM02], in order to obtain a good accuracy-complexity trade-off.
- *Classifier selection and fusion.* As a combination method joining both families, classifier selection and classifier fusion, we proposed the use of a novel GFS, which not only improves the accuracy while reducing the complexity of the FRBCE but also allows us to benefit from the key advantage of fuzzy systems, i.e., their interpretability.

The different specific FRBCE design methods obtained from the general methodology are introduced and tested in Sections 3.2 and 3.3. Finally, to validate our framework we successfully applied one of the static FRBCEs to a real-world problem, which consists of topology-based indoor localization (bottom-left corner in the figure). Section 3.4 presents the latter development.

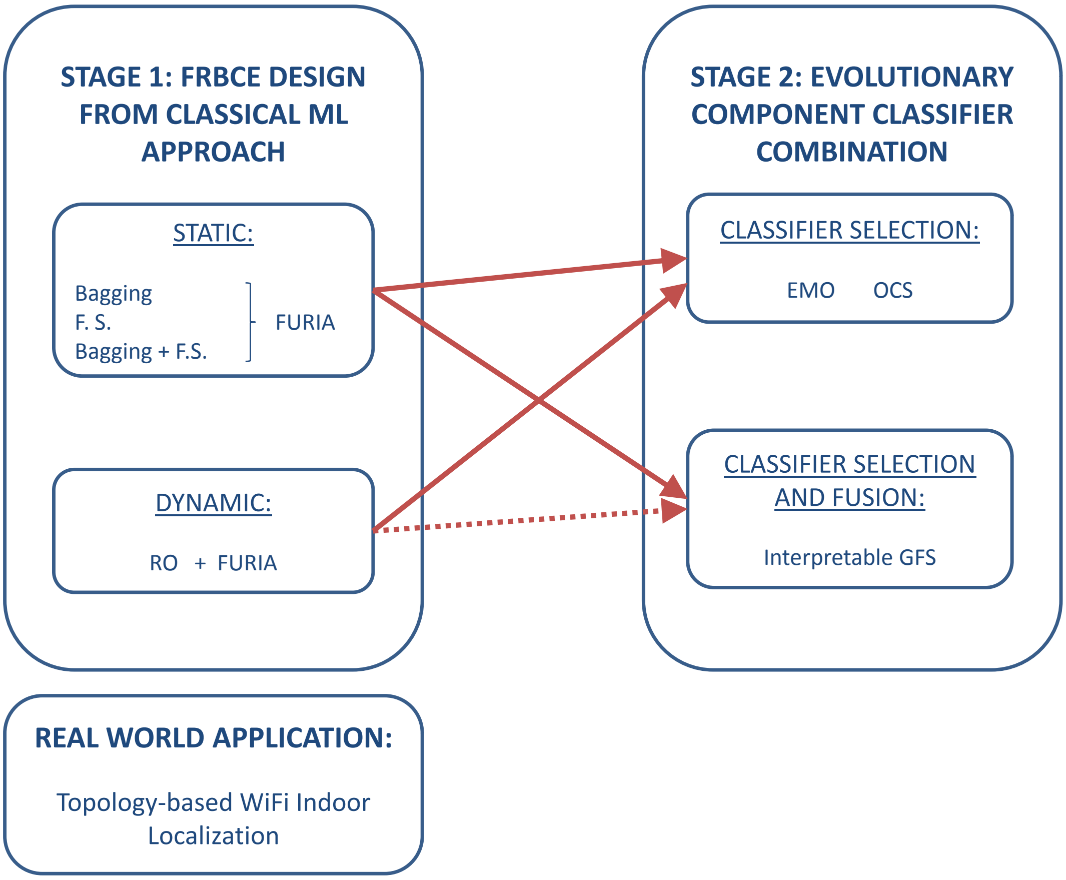


Figure 1: The proposed framework is composed of several FRBCEs design methodologies embedded into two stages: 1) FRBCE design from classical ML approaches, and 2) evolutionary component classifier combination. It also presents an application of the static FRBCEs to a real-world problem, involving topology-based indoor localization.

3.2 Stage 1: Design of the Component Classifiers via Diversity Induction

3.2.1 Static Approach: Bagging, feature selection, and Bagging with feature selection

In [PD07, Ste05] it was shown that a combination between bagging and feature selection composed a general design procedure usually leading to good CE designs, regardless the classifier structure considered. Hence, we decided to follow that approach by integrating FURIA into a framework of that kind. Our aim was to combine the diversity induced by the CE design methods and the robustness of the FURIA method (see Sec. 2.2) in order to derive good performance FURIA-based FRBCEs for high dimensional problems [TCQ11a, TCQ11b]. We also tried 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 FRBCEs.

We considered three different types of feature selection algorithms: random subspace [Ho98], mutual information-based feature selection (MIFS) [Bat94], and the random-greedy feature selection based on MIFS and the GRASP approach [FR95].

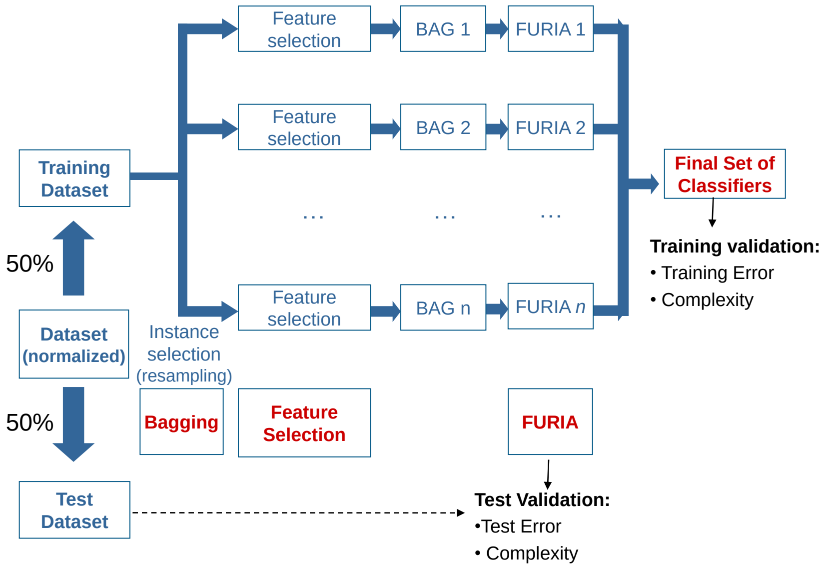


Figure 2: Our initial framework: after the instance and the feature selection procedures, the component fuzzy classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based combination method.

The term *bagging* is an acronym of bootstrap aggregation and refers to the first successful method to generate CEs proposed in the literature [Bre96]. This approach was originally designed for decision tree-based classifiers, however it can be 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 been individually 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 more effective when dealing with unstable classifiers (the so-called "weak learners"), what means a small change in the training set can cause a significant change in the final model. In addition, it is recommended when the given dataset is composed of a small amount of examples. Furthermore, bagging enables a parallel and independent learning of the learners in the ensemble.

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 [Bat94] is based on a forward greedy search using the mutual information measure [SW49], 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:

$$I(C, F) = \sum_{c,f} P(c, f) \log \frac{P(c, f)}{P(c)P(f)} \tag{I.2}$$

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.

Table I.1: Data sets considered

abbrev.	Dataset	#attr.	#examples	#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

The random-greedy variant is an approach where the feature subset 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.

Table I.2: Average and standard deviation values for the different FURIA-based CE 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

FURIA-based FRBCEs are designed as follows. 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. Let us emphasize that FURIA already incorporates an internal feature selection algorithm, being one of the features inherently owned from the RIPPER algorithm. The whole procedure is graphically presented in Fig. 2, which presents FRBCE design approaches tested such as combination between bagging and feature selection, as well as bagging and feature selection separately.

An exhaustive study was developed comparing all the variants proposed. We selected 21

Table I.3: Results for the best choices of each different approach for FURIA-based fuzzy CE for each dataset

FURIA single classifier - All features																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra err.	0.781	0.023	0.336	0.141	0.041	0.038	0.143	0.633	0.018	0.003	0.132	0.193	0.042	0.008	0.154	0.043	0.007	0.331	0.043	0.004	0.433
test err.	0.805	0.049	0.377	0.227	0.163	0.123	0.157	0.683	0.033	0.027	0.160	0.245	0.122	0.042	0.298	0.070	0.055	0.364	0.187	0.056	0.441

FURIA-based CEs obtained from bagging only.																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra err.	0.570	0.010	0.096	0.052	0.031	0.016	0.110	0.246	0.015	0.002	0.084	0.075	0.025	0.006	0.018	0.028	0.004	0.051	0.017	0.006	0.223
test err.	0.755	0.044	**0.313**	**0.178**	0.152	**0.091**	**0.136**	0.641	0.030	0.017	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.	10	7	7	7	10	10	7	10	10	10	7	10	10	10	10	10	10	10	10	10	10

FURIA-based CEs obtained from feature selection only.																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra err.	0.754	0.018	0.146	0.113	0.050	0.037	0.139	0.627	0.014	0.002	0.120	0.204	0.052	0.018	0.005	0.075	0.006	0.217	0.089	0.002	0.364
test err.	0.786	**0.037**	0.316	0.185	**0.134**	0.101	0.151	**0.628**	**0.028**	**0.015**	0.153	0.244	0.110	0.039	**0.198**	0.088	0.088	0.310	0.164	**0.036**	0.432
feat. sel.	R	R	R	RG	RG	RG	RG	RG	R	R	R	RG	R	RG	R	RG	RG	R	RG	RG	R
feat. s.s.	L	L	L	M	S	L	L	L	L	L	L	L	L	L	L	L	L	L	M	M	L
nr of cl.	10	10	10	7	7	10	10	10	10	10	7	7	10	10	10	7	10	10	10	10	10

FURIA-based CEs obtained from bagging and feature selection.																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
tra err.	0.622	0.021	0.087	0.079	0.032	0.026	0.113	0.621	0.015	0.020	0.085	0.109	0.052	0.014	0.018	0.067	0.005	0.076	0.039	0.020	0.257
test err.	**0.753**	0.039	0.318	0.179	0.143	0.096	0.138	0.630	0.030	**0.015**	**0.136**	**0.235**	0.110	0.037	0.214	0.084	0.041	0.284	0.156	**0.036**	0.416
feat. sel.	G	R	R	RG	RG	RG	RG	R	R	R	R	RG	R	RG	R	G	RG	R	G	R	G
feat. s.s.	L	M	L	L	S	L	L	S	L	L	L	L	L	L	L	L	L	L	L	M	L
nr of cl.	10	7	7	7	10	7	10	10	10	10	10	10	10	10	7	7	10	10	10	10	10

Table I.4: A comparison of the best choice for different approaches for FURIA-based fuzzy CEs against the best choice of bagging C4.5 CEs, random forests, and Ishibuchi-based fuzzy CEs

FURIA-based CEs																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.753	**0.037**	0.313	**0.178**	0.134	0.091	0.136	**0.628**	**0.028**	**0.015**	0.136	**0.235**	0.105	0.035	**0.198**	0.061	**0.036**	0.276	**0.156**	**0.036**	**0.408**
feat. sel.	G	R	-	-	RG	-	-	RG	R	R	R	RG	-	-	R	-	-	-	-	RG	-
feat. s.s.	L	L	-	-	S	-	-	L	L	L	L	L	-	-	L	-	-	-	-	M	-
nr of cl.	10	10	7	7	7	10	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10

C4.5 ensembles with bagging																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.772	0.043	0.306	0.194	0.149	0.103	**0.134**	0.697	0.030	0.028	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	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

random forests																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.777	0.041	**0.282**	0.211	0.140	**0.080**	**0.134**	0.695	0.031	0.016	**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.	7	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Ishibuchi-based fuzzy CEs																					
	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	**0.751**	0.056	0.379	0.213	**0.129**	0.420	0.202	0.629	0.075	0.062	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	7	10	7	3	7	10	3	7	7	10	10	10	7	3	7	10	7
feat. sel.	R	R	G	R	RG	RG	R	R	RG	R	G	G	RG	RG	RG	G	RG	RG	RG	G	G

datasets from the UCI machine learning repository [BM98] with different characteristics concerning the number of examples, features, and classes (see Table I.1). For validation we used Dietterich's 5×2-fold cross-validation (5×2-cv) [Die98]. Three different feature subsets of different sizes (Small "S", Medium "M", and Large "L") were tested for the FURIA-based fuzzy CEs using the three different feature selection algorithms (Greedy "G", Random-greedy "RG", and Random subspace "R"). A small number of component fuzzy classifiers (up to 10) was considered in this study. Finally, the best choices of FURIA-based FRBCEs were compared to two state-of-the-art CE algorithms such as bagging decision trees and random forests, as well as with the use of the same methodology combined with a different fuzzy classifier generation method, Ishibuchi-based fuzzy CE [INN05].

We only report here the most representative results that we have obtained for this part of the study. The reader is referred to Sec. 1 in Part II of this PhD dissertation for the complete experiments and analysis of the results. Table I.2 benchmarks average and standard deviation values computed for each of the FURIA-based fuzzy CEs 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, where the last column provides global statistics for each of the approaches. The best result of each approach for each dataset regardless the parameter choice is presented in Table I.3, which consists of statistics (5x2-cv training and testing errors) and algorithm parameters (feature selection algorithm "feat. sel.", feature subset size "feat. s. s.", and number of classifiers "nr of cl.") for each of the twenty one datasets. The best accuracy obtained for the given dataset is emphasized in bold font. Finally, Table I.4 presents the final comparison of the best choices of FURIA-based fuzzy CEs with bagging C4.5 CEs and random forests, as well as with the use of the same methodology combined with a different fuzzy classifier generation method, Ishibuchi-based fuzzy CE. For each algorithm, we only show the best obtained result in terms of accuracy (5×2-cv test error values) for each dataset and highlight the best values in boldface.

The main obtained conclusion is that FURIA-based fuzzy CEs perform better when only combined with bagging (no additional feature selection is required) and that results are promising in comparison with state-of-the-art classical CEs.

3.2.2 Dynamic Approach: Bagging with Random Oracles

This section introduces the use of ROs [KR07, RK07] within the bagging CE framework to derive FURIA-based FRBCEs. Our idea is that, thanks to the additional diversity introduced by ROs into the base classifiers, the obtained FRBCEs are able to achieve an outstanding performance in terms of accuracy [TCQS13b, TCQS13a, TCQ13].

An RO (see Sec. 2.1.1) is a structured classifier, also defined as a "mini-ensemble", encapsulating the base classifier of the CE. It is composed of two subclassifiers and an oracle that decides which one to use in each case. Basically, the oracle is a random function whose objective is to randomly split the dataset into two subsets by dividing the feature space into two regions. Each of the two generated regions (together with the corresponding data subset) is assigned to one classifier. Any shape for the decision surface of the function can be applied as far as it divides the training set into two subsets at random.

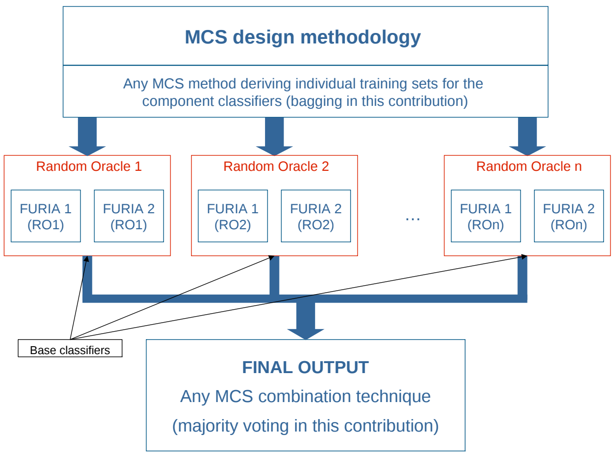


Figure 3: The RO-based framework: after obtaining bootstrapped replicas, the individual component classifiers are derived by RO composed of an oracle and two FURIA-based subclassifiers. The final output is taken by means of the majority voting, an inherent feature of bagging.

Let us emphasize that during the classification phase, the oracle commits an internal dynamic classifier selection, that is to say it decides which subclassifier makes the final decision for the given example to be further used at the ensemble level (classifier fusion). Thus, this CE method belongs to the *dynamic* family [GR01, DSM08] (see Sec. 2.1.2).

The RO approach owns several interesting features, making it quite unique among the existing CE solutions:

- It is a generic approach composing a framework in which ROs embed only the base classifier.

Table I.5: Datasets considered

Dataset	#ex.	#attr.	(R/I/N)	cmpl.	#classes
abalone	4178	8	(7/0/1)	3.3	28
bioassay_688red	27190	153	(27/126/0)	416.0	2
coil2000	9822	85	(0/85/0)	83.5	2
gas_sensor	13910	128	(128/0/0)	178.0	7
isolet	7797	617	(617/0/0)	481.1	26
letter	20000	16	(0/16/0)	32.0	26
magic	19020	10	(10/0/0)	19.0	2
marketing	6876	13	(0/13/0)	8.9	9
mfeat_fac	2000	216	(0/216/0)	43.2	10
mfeat_fou	2000	76	(76/0/0)	15.2	10
mfeat_kar	2000	64	(64/0/0)	12.8	10
mfeat_zer	2000	47	(47/0/0)	9.4	10
musk2	6598	166	(0/166/0)	109.5	2
optdigits	5620	64	(0/64/0)	36.0	10
pblocks	5474	10	(4/6/0)	5.5	5
pendigits	10992	16	(0/16/0)	17.6	10
ring_norm	7400	20	(20/0/0)	14.8	2
sat	6436	36	(0/36/0)	23.2	6
segment	2310	19	(19/0/0)	4.4	7
sensor_read_24	5456	24	(24/0/0)	13.1	4
shuttle	58000	9	(0/9/0)	52.2	7
spambase	4602	57	(57/0/0)	26.2	2
steel_faults	1941	27	(11/16/0)	5.2	7
texture	5500	40	(40/0/0)	22.0	11
thyroid	7200	21	(6/15/0)	15.1	3
two_norm	7400	20	(20/0/0)	14.8	2
waveform_noise	5000	40	(40/0/0)	20.0	3
waveform	5000	21	(21/0/0)	10.5	3
wquality_white	4898	11	(11/0/0)	5.4	7

Thus, it allows a design choice at two different levels: i) any CE strategy can be applied; ii) any classifier learning algorithm can be used. Apart from that, it can be used as the CE generation method on its own.

- It induces an additional diversity through the randomness coming from the nature of ROs. Generating a set of diverse base classifiers was shown to be fundamental for the CE overall performance [OM99, KW03]. Let us emphasize that ROs are applied separately to each of the base classifiers and no training of the oracle is recommended, as it will strongly diminish the desired diversity.

- It embeds the two most common and complementary CE combination methods, i.e. *classifier fusion* and *(dynamic) classifier selection*.

- A wide study has been carried out over several CE generation approaches in order to analyze the influence of ROs on these methods [KR07, RK07]. C4.5 decision trees [Qui93] (in [KR07]) and Naïve Bayes [DP97] (in [RK07]) were the base classifiers used. All the CE approaches took an advantage of the ROs, outperforming the original CEs in terms of accuracy. Especially, the highest accuracy improvement was obtained by random subspace and bagging according to [KR07].

In particular, we considered two versions of ROs: random linear oracle (RLO) [KR07, RK07] and random spherical oracle (RSO) [RK07]. The former uses a randomly generated hyperplane to divide the feature space, while the latter does so using a hypersphere. The global framework of this proposal, namely RO-based bagging FRBCE approach is presented in Fig. 3.

We selected 29 datasets with different characteristics concerning a high number of examples (see Table I.5), features, and classes from the UCI machine learning [BM98] and KEEL [AFFL+11] repositories. For validation, 5×2-cv was used. We studied the performance of both RO-based bagging FRBCEs in comparison with bagging FRBCEs considering both accuracy and complexity. Then, the best performing FRBCEs were compared against state-of-the-art RO-based bagging CEs. By doing so, we wanted to show that RO-based bagging FRBCEs are competitive against the state-of-the-art RO-based bagging CEs using C4.5 [KR07, RK07] and Naïve Bayes [RK07] as the base classifiers, when dealing with high dimensional datasets, thanks to the use of the FURIA algorithm.

For an illustrative purpose, we include several tables in this section, reporting the most significant results obtained (the whole experimentation can be checked in Sec. 4 in Part II of this PhD dissertation). Table I.6 collects the test errors for for the three FRBCEs considered: bagging FRBCEs, RLO-based bagging FRBCEs, and RSO-based bagging FRBCEs, considering an equivalent complexity (see Sec. 4). The best result for a given dataset is presented in bold font. The average "Avg." and standard deviation "Std. Dev." values over the 29 datasets are reported at the bottom of the table. Tables I.7 and I.8 show the statistical tests carried out for the results obtained in the previous table. Furthermore, the same study considering complexity is presented in Tables I.9, I.10, and I.11. Finally, Table I.12 presents the test results achieved by RSO-based bagging FRBCEs and RSO-based bagging CEs using C4.5 and NB, as well as random forests over the 29 selected datasets. Tables I.13 and I.14 show the statistical tests carried out for the results obtained in the previous table. The competitive performance of our proposal can be observed.

Table I.6: A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging FRBCEs (100 classifiers) in terms of accuracy. FURIA serves as the base classifier in the three approaches

Dataset	**BAG Test err.**	**BAG+RLO Test err.**	**BAG+RSO Test err.**
abalone	0.7455	0.7452	**0.7480**
	0.0090	**0.0090**	**0.0090**
coil2000	0.0602	**0.0601**	**0.0601**
gas_sensor	0.0086	0.0079	**0.0078**
isolet	0.0774	**0.0691**	0.0700
letter	0.0778	**0.0742**	0.0743
magic	0.1325	0.1314	**0.1299**
marketing	0.6749	0.6673	**0.6671**
mfeat_fac	0.0547	0.0434	**0.0431**
mfeat_fou	0.1992	0.1941	**0.1925**
mfeat_kar	0.0825	**0.0699**	0.0709
mfeat_zer	0.2231	**0.2169**	0.2181
musk2	0.0338	0.0328	**0.0320**
optdigits	0.0324	0.0283	**0.0282**
pblocks	**0.0335**	0.0353	0.0338
pendigits	0.0155	0.0137	**0.0132**
ring_norm	0.0432	0.0438	**0.0315**
sat	0.1013	0.1008	**0.1001**
segment	0.0309	0.0303	**0.0295**
sensor_read_24	**0.0222**	0.0227	0.0233
shuttle	**0.0008**	0.0009	0.0009
spambase	0.0663	0.0651	**0.0639**
steel_faults	0.2371	0.2367	**0.2361**
texture	0.0288	0.0278	**0.0274**
thyroid	**0.0212**	0.0215	0.0218
two_norm	0.0316	**0.0271**	0.0276
waveform_noise	0.1480	0.1461	**0.1457**
waveform	0.1480	**0.1451**	0.1453
wquality_white	0.3908	0.3840	**0.3803**
Avg.	0.1286	0.1259	**0.1252**
Std. Dev.	0.1833	0.1825	0.1829

Table I.7: Average Rankings of the Friedman's test

Algorithm	Ranking
FURIA+BAG+RSO	1.552
FURIA+BAG+RLO	1.828
FURIA+BAG	2.621

Table I.8: The adjusted p-values of Shaffer test for the pair-wise comparisons (FURIA is the base classifier in every case)

Comparison	**p-value**
BAG+RSO vs BAG	**+(1.41e-4)**
BAG+RLO vs BAG	**+(0.002)**
BAG+RSO vs BAG+RLO	=(0.293)

Table I.9: A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging CEs (100 classifiers) in terms of complexity (number of rules). FURIA serves as the base classifier in the three approaches

Dataset	**BAG # Rules**	**BAG+RLO # Rules**	**BAG+RSO # Rules**
abalone	**8369.0**	8696.7	9382.8
bioassay_688red	5526.9	**4642.8**	4780.8
coil2000	4331.9	**3804.1**	4002.1
gas_sensor	8628.3	**7091.3**	7310.7
isolet	12215.7	**10523.6**	10828.5
letter	47109.1	**39410.5**	40972.9
magic	13770.8	**13143.0**	14556.9
marketing	**6418.5**	7252.0	7429.1
mfeat_fac	3479.9	**3050.2**	3110.3
mfeat_fou	5483.5	**4711.4**	4886.9
mfeat_kar	4953.3	**4448.4**	4581.0
mfeat_zer	5028.3	**4349.9**	4549.2
musk2	4332.2	**3581.1**	3582.7
optdigits	7167.3	**6352.4**	6511.1
pblocks	3201.7	2877.9	**2816.4**
pendigits	8788.6	**7348.0**	7491.6
ring_norm	7308.9	6205.7	**5961.4**
sat	8454.4	**6956.2**	7109.5
segment	2546.3	**2201.6**	2378.7
sensor_read_24	3430.8	**3340.4**	3428.3
shuttle	1826.2	**1723.8**	1737.5
spambase	3612.9	**3281.9**	4181.1
steel_faults	5467.3	**4799.0**	4857.0
texture	6537.2	**5305.7**	5542.8
thyroid	3299.5	**2831.7**	2959.8
two_norm	6147.5	**4973.3**	5307.8
waveform_noise	7932.6	**6729.9**	6850.6
waveform	8303.0	**7017.3**	7115.0
wquality_white	13429.3	**12134.0**	12564.4
Avg.	7831.1	6854.6	7130.6
Std. Dev.	8144.6	6857.3	7156.8
Avg. (Without Letter)	6428.3	5691.9	5921.9
Std. Dev. (Without Letter)	3100.2	2847.3	3030.4

3.3 Stage 2: Evolutionary Component Classifier Combination

3.3.1 Evolutionary Multiobjective Overproduce-and-Choose static classifier selection

In this section, we describe our proposal of an EMO method defining an OCS strategy for the component classifier selection [TQC12]. Our goal is to obtain a good accuracy-complexity trade-off

Table I.10: Average Rankings of the Friedman's test

Algorithm	Ranking
FURIA+BAG+RLO	1.138
FURIA+BAG+RSO	2.069
FURIA+BAG	2.793

Table I.11: The adjusted p-values of Shaffer test for the pair-wise comparisons (FURIA is the base classifier in every case)

Comparison	**p-value**
BAG+RLO vs BAG	**+(8.77e-10)**
BAG+RSO vs BAG	**+(0.006)**
BAG+RLO vs BAG+RSO	=**+(3.92e-4)**

Table I.12: A comparison of RSO-based bagging CEs using FURIA, C4.5 and NB, as well as random forests in terms of accuracy

Dataset	**FURIA Test err.**	**C4.5 Test err.**	**NB Test err.**	**RF Test err.**
abalone	**0.7480**	0.7681	0.7619	0.7536
bioassay_688red	**0.0090**	**0.0090**	0.0152	**0.0090**
coil2000	0.0601	0.0615	0.1847	**0.0597**
gas_sensor	**0.0078**	0.0089	0.2939	0.0092
isolet	**0.0700**	0.0788	0.1246	0.0766
letter	0.0743	**0.0615**	0.2927	0.0701
magic	0.1299	**0.1255**	0.2391	0.1314
marketing	0.6671	0.6735	0.6864	**0.6624**
mfeat_fac	**0.0431**	0.0498	0.0659	0.0475
mfeat_fou	0.1925	0.1902	0.2221	**0.1858**
mfeat_kar	0.0709	0.0818	**0.0593**	0.0597
mfeat_zer	**0.2181**	0.2273	0.2464	0.2330
musk2	0.0320	**0.0271**	0.1107	0.0375
optdigits	0.0282	**0.0276**	0.0709	0.0277
pblocks	0.0338	**0.0327**	0.0706	0.0332
pendigits	**0.0132**	0.0150	0.0864	0.0162
ring_norm	**0.0315**	0.0376	0.0199	0.0587
sat	0.1001	**0.0950**	0.1720	0.1027
segment	**0.0295**	0.0328	0.1180	0.0350
sensor_read_24	0.0233	0.0234	0.3710	**0.0224**
shuttle	**0.0009**	**0.0009**	0.0143	**0.0009**
spambase	0.0639	0.0651	0.1788	**0.0625**
steel_faults	0.2361	**0.2263**	0.3441	0.2517
texture	**0.0274**	0.0334	0.1384	0.0383
thyroid	**0.0218**	0.0222	0.0381	0.0221
two_norm	0.0276	0.0280	**0.0219**	0.0389
waveform_noise	**0.1457**	0.1643	0.1668	0.1556
waveform	**0.1453**	0.1588	0.1534	0.1587
wquality_white	0.3803	**0.3688**	0.5230	0.3864
Avg.	**0.1252**	0.1274	0.1997	0.1292
Std. Dev.	0.1829	0.1852	0.1890	0.1830

Table I.13: Average Rankings of the Friedman's test

Algorithm	Ranking
FURIA+BAG+RSO	1.793
C4.5+BAG+RSO	2.276
RF	2.345
NB+BAG+RSO	3.586

Table I.14: The adjusted p-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE (using FURIA) is the control method

Comparison	**p-value**
FURIA+BAG+RSO vs C4.5+BAG+RSO	=(0.207)
FURIA+BAG+RSO vs RF	=(0.207)
FURIA+BAG+RSO vs NB+BAG+RSO	+**(3.69e-7)**

in the FURIA-based FRBCEs when dealing with high dimensional problems. That is, we aim to obtain FRBCEs with a low number of base classifiers, which jointly keep a good accuracy, even better than that of the full original FRBVE in many cases. Thus, we have selected the state-of-the-art NSGA-II EMO algorithm [DPAM02] in order to generate good quality Pareto set approximations.

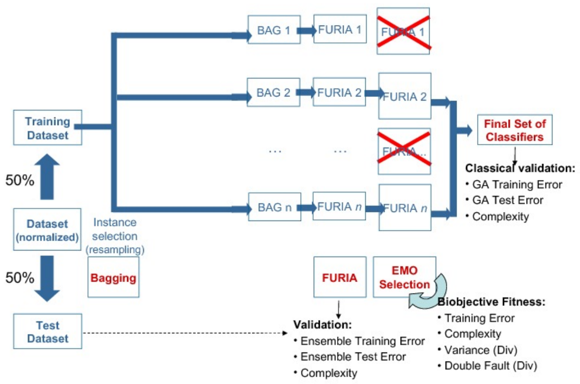


Figure 4: The EMO OCS framework: after the instance and the feature selection procedures, the component classifiers are derived by the FURIA learning method. Then, the EMO OCS stage takes place by means of NSGA-II. Finally the output is obtained using a voting-based method.

Table I.15: The five fitness function proposed for the EMO OCS method

1st obj.	2nd obj.
TE	Complx
TE	θ
TE	δ
θ	Complx
δ	Complx

NSGA-II is based on a Pareto dominance depth approach, where the population is divided into several fronts and the depth of each front shows to which front an individual belongs to. A pseudo-dominance rank being assigned to each individual, which is equal to the front number, is the metric used for the selection of an individual.

We have used a standard binary coding in such a way that a binary digit/gene is assigned to each classifier. When the variable takes value 1, it means that the current component classifier belongs to the final ensemble, while when the variable is equal to 0, that classifier is discarded. This approach provides a low operation cost, which leads to a high speed of the algorithm.

Five different biobjective fitness functions combining the three existing kinds of optimization criteria (accuracy, complexity, and diversity, see Sec. 2.1.2) are proposed in order to study the best setting. Fig. 4 shows the final structure of the FURIA-based fuzzy MCS design methodology

Table I.16: Datasets considered

Data set	#examples	#attr.	#classes
abalone	4178	7	28
breast	700	9	2
glass	214	9	7
heart	270	13	2
ionosphere	352	34	2
magic	19020	10	2
optdigits	5620	64	10
pblocks	5474	10	5
pendigits	10992	16	10
phoneme	5404	5	2
pima	768	8	2
sat	6436	36	6
segment	2310	19	7
sonar	208	60	2
spambase	4602	57	2
texture	5500	40	11
waveform	5000	40	3
wine	178	13	3
vehicle	846	18	4
yeast	1484	8	10

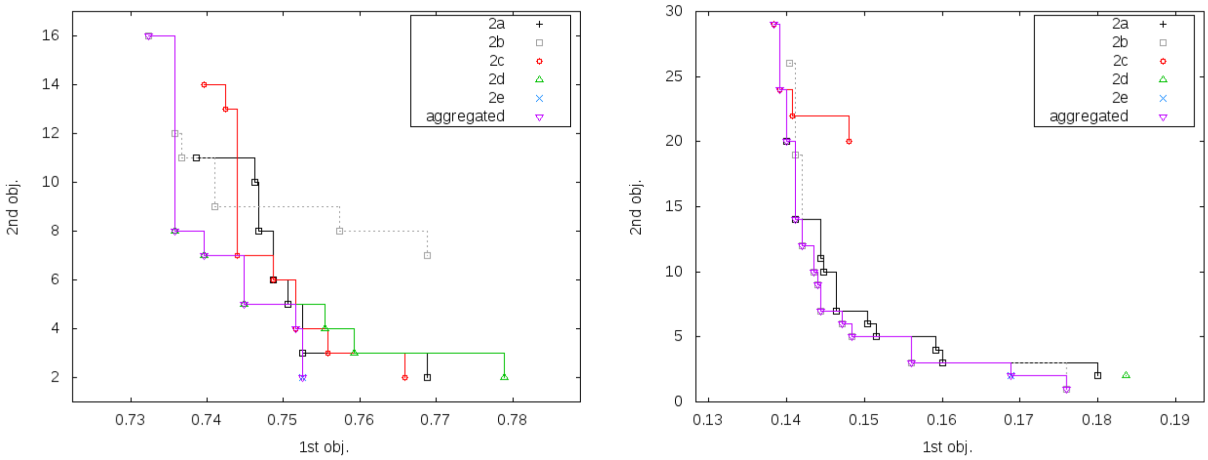
including the OCS stage. We use the following measures: the training error (accuracy), the number of classifiers (complexity), and the difficulty measure θ and the double fault δ (diversity) [KW03, TPC05, RG05]. Table I.15 presents the five combinations proposed.

The initial fuzzy CEs are based on applying a bagging approach with the FURIA method as described in Section 3.2.1. Each FRBCE so generated is composed of 50 weak learners.

Table I.17: Comparison of Pareto fronts using the HVR measure

	2a	2b	2c	2d	2e
aba	**0.9973**	0.5126	**0.9973**	0.9961	0.9962
bre	0.6632	**0.9955**	0.3321	0.6627	0.6644
gla	0.8455	**0.9867**	0.8314	0.8376	0.8469
hea	0.6582	**0.9858**	0.5915	0.6564	0.6625
ion	0.9437	**0.9796**	0.5294	0.9416	0.9464
mag	0.9323	**0.9988**	0.9324	0.9300	0.9307
opt	**0.9952**	0.3335	0.3335	**0.9952**	**0.9952**
pbl	0.8555	**0.9983**	0.8555	0.8547	0.8553
pen	0.9609	**0.9992**	0.4307	0.9580	0.9587
pho	0.9267	**0.9978**	0.9266	0.9224	0.9241
pim	0.8700	**0.9944**	0.8700	0.8650	0.8730
sat	0.9554	**0.9988**	0.1738	0.9510	0.9528
seg	0.9483	**0.9982**	0.3295	0.9452	0.9472
son	0.6544	**0.9797**	0.3927	0.6492	0.6597
spa	0.9071	**0.9978**	0.1542	0.9047	0.9060
tex	0.9587	**0.9983**	0.3518	0.9525	0.9542
veh	0.8523	**0.9940**	0.8520	0.8459	0.8521
wav	0.9638	**0.9984**	0.2068	0.9554	0.9585
win	0.9240	**0.9893**	0.1066	0.9213	0.9265
yea	0.9315	**0.9947**	0.9311	0.9256	0.9301
avg.	0.8450	**0.8920**	0.5299	0.8415	0.8870
dev.	0.2202	0.2682	0.3263	0.2194	0.1058

We carried out an experiment comparing all five biobjective fitness functions. We have selected 20 datasets from the UCI machine learning repository with different characteristics concerning the number of examples, features, and classes (see Table I.16). To compare the Pareto front approximations of the global learning objectives (i.e. CE test accuracy and complexity), we considered two of the usual kinds of multiobjective metrics, namely hypervolume ratio (HVR) [CLV07] and C-measure [ZT99], respectively. We also analyzed single solutions extracted from the obtained Pareto front approximations.



(a) (b)

Figure 5: The Pareto front approximations obtained for two datasets using the five fitness functions: (a) abalone, and (b) waveform. Objective 1 stands for test error and objective 2 for complexity. The pseudo-optimal Pareto front is also drawn for reference

Table I.18: A comparison of the NSGA-II FURIA-based fuzzy CEs against static FURIA-based CE

NSGA-II combined with FURIA-based CEs.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	**0.741**	**0.037**	0.283	**0.170**	**0.126**	**0.132**	**0.625**	**0.027**	**0.014**	**0.125**	**0.231**	**0.101**	**0.027**	**0.188**	**0.056**	**0.028**	**0.255**	**0.146**	**0.018**	**0.396**
fit. f.	2b	2b	2c	2b	2c	2a	2b	2c	2c	2c	2e	2b	2c	2e	2b	2c	2b	2c	2c	2b
# cl.	18.6	2.7	5.5	2	18.7	5.6	26	4.8	21.8	9	2	14.6	17.6	2	6.8	23.2	7.5	18.7	18.7	7.1
FURIA-based CE algorithms. Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.753	**0.037**	0.313	0.178	0.134	0.136	0.628	0.028	0.015	0.136	0.235	0.105	0.035	0.198	0.061	0.036	0.276	0.156	0.036	0.408
# cl.	10	10	7	7	7	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10
FURIA-based CE algorithms. Ensemble size 50.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.748	0.041	0.287	0.182	0.145	0.135	0.630	0.028	0.016	0.135	0.241	0.102	0.034	0.226	0.059	0.031	0.275	0.149	0.035	0.400
C4.5 ensembles with bagging. Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.772	0.043	0.306	0.194	0.149	0.134	0.697	0.03	0.028	0.131	0.253	0.112	0.042	0.247	0.067	0.051	0.289	0.193	0.097	0.415
# cl.	10	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Random forests. Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.777	0.041	**0.282**	0.211	0.14	0.134	0.695	0.031	0.016	0.119	0.264	0.104	0.034	0.239	0.06	0.04	0.269	0.185	0.048	0.438
# cl.	7	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

We show some representative results for this study in the current section (the remainder can be referred to Sec. 2 in Part II of this PhD dissertation). In Table I.17, we present the HVR metric results. For illustration purposes, the aggregated Pareto fronts are represented graphically for the abalone and waveform datasets in Figures 5a and 5b, which allows an easy visual comparison of the performance of the different EMO OCS-based FRBCE variants. Finally, Table I.18 benchmarks the performance of the EMO OCS proposal in terms of individual results.. FURIA-based fuzzy CEs are comprised by 7 or 10 classifiers, the small ensemble sizes providing the best results in our previous contribution [TCQ11b] (see Section 3.2.1), and with 50 classifiers, the initial structure of the EMO-selected fuzzy CEs. We also compare them with two state-of-the-art algorithms, random forests [Bre01] and bagging C4.5 CEs [Qui93], comprised by 7 or 10 classifiers [TCQ11b]. An accurate performance with a small number of classifiers is obtained.

3.3.2 Joint Classifier Selection and Fusion via an Interpretable Genetic Fuzzy System

The aim of the current section is to present a fuzzy linguistic rule-based classification system playing the role of CE combination method (a FRBCS-CM) [SCQT10, TCSQ13]. Our design fulfills several requirements, namely: i) showing a human-understandable structure; ii) being able to deal with high dimensional problems avoiding the curse of dimensionality; iii) having the chance to be automatically learned from training data; and iv) being able to perform both classifier fusion and selection in order to derive low complexity FRBCEs with a good accuracy-complexity trade-off (see Fig. 6) [3].

Using the novel FRBCS-CM together with a FRBCE, we have the additional advantage of handling a two-level hierarchical structure composed of the individual classifiers in the first level and the FRBCS-CM in the second. These kinds of hierarchical structures [Tor02, GF95, Yag98, CHZ03] are well known in the area as they allow fuzzy systems to properly deal with high-dimensional problems while maintaining their descriptive power, especially when considering the single-winner rule fuzzy reasoning method [CdJH99, INN05] in the component fuzzy classifiers as done in our case.

One step further, using it in combination with a bagging strategy as done in this proposal, we can also benefit from some collateral advantages for the overall design of the FRBCE: a) the simplicity of the implicit parallelism of bagging, which allows for an easy parallel implementation; and b) the problem partitioning due to the internal feature selection at the component classifier level and the classifier selection capability of the fuzzy linguistic combination method, resulting in a tractable dimension for learning fuzzy rules for each individual classifier and for achieving a compact FRBCE. These characteristics make the fuzzy ensemble using the FRBCS-CM specially able to deal with the curse of dimensionality.

Our approach might thus be assigned to the stacking (or stacked generalization) group [Wol92], which after bagging and boosting is probably the most popular approach in the literature. Its basis lay in the definition of the meta-learner, playing a role of (advanced) CE combination method, giving a hierarchical structure of the ensemble. Its task is to gain knowledge of whether training data have been properly learned and to be able to correct badly trained base classifiers. The proposed FRBCS-CM acts as the meta-learner, by discarding the rule subsets in the base fuzzy classifiers providing incorrect decisions at individual class level and promoting the ones leading to a correct classification.

Moreover, fuzzy classification rules with a class and a certainty degree in the consequent [CdJH99, INN05] used in FRBCS-CM allows the user to get an understandable insight to the CE considered, namely bagging FURIA FRBCE. This means that this approach allows interpretability (to some extent) of such complicated system.

The proposed FRBCS-CM is built under the GFS approach (see Sec. 2.3) (in particular, being an interpretable GFS). A specific GA, which uses a sparse matrix to codify features and linguistic terms in the antecedent parts of the rules and a fitness function based on three accuracy components performs both classifier fusion and classifier selection at class level. The complexity of the final ensemble is defined by the number of terms in the sparse matrix different than zero ("nonzero value"), which is a designed parameter provided by the user as a percentage reduction.

To evaluate the performance of the FRBCS-CM in the ensembles generated, 20 popular datasets from the UCI machine learning repository have been selected with a number of features varying from a small value (i.e., 5) to a large one (i.e., 64), while the number of examples scales

[3] We should mention that the proposed combination method can be applied to any CE with the only restriction that the component classifiers must additionally provide certainty degrees associated to each class in the dataset.

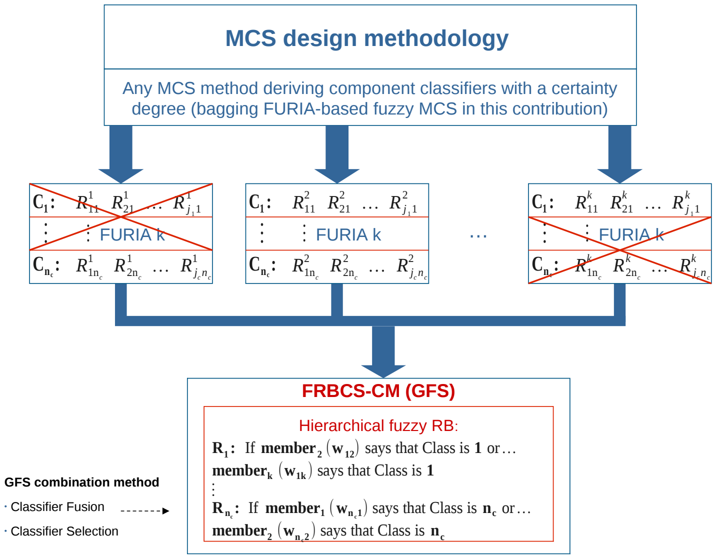


Figure 6: The FRBCS-CM framework: after the instance selection, the component classifiers are derived by the FRBCS learning method (or by any method deriving component classifiers with a certainty degree). Then, the fuzzy linguistic rule-based classification system playing the role of MCS combination method (a FRBCS-CM) selects rules with a proper behavior in order to obtain a good interpretability-accuracy trade-off. Finally, the output is obtained using the FRBCS-CM fuzzy reasoning mechanism.

Table I.19: Datasets considered

Data set	#examples	#attr.	#classes
Low dimensional:			
abalone	4178	7	28
breast	700	9	2
glass	214	9	7
heart	270	13	2
magic	19020	10	2
pblocks	5474	10	5
phoneme	5404	5	2
pima	768	8	2
wine	178	13	3
yeast	1484	8	10
High dimensional:			
ionosphere	352	34	2
letter	20000	16	26
optdigits	5620	64	10
pendigits	10992	16	10
sat	6436	36	6
segment	2310	19	7
sonar	208	60	2
spambase	4602	57	2
texture	5500	40	11
vehicle	846	18	4
waveform	5000	40	3

from 208 to 19 020 (see Table I.19). In order to compare the accuracy of the considered classifiers, we used 5×2-cv. This study was carried in a three-fold manner. Firstly, we compared bagging FRBCEs combined with our interpretable GFS performing classifier selection and fusion over bagging FRBCEs with the full ensemble using standard majority voting (MV). Secondly, we compared the novel interpretable GFS with state-of-the-art crisp and fuzzy CE combination methods, as well as with a hybrid method based on GA considering both classifier selection and classifier fusion [DVA09]

Table I.20: Accuracy of the original FRBCEs, FRBCS-CM, and the other CE combination methods in terms of test error

	fuzzy	FRBCS-CM					Greedy FS			Greedy BS			GA
Dataset	CEs	10%	25%	50%	75%	90%	MV	AVG	DT	MV	AVG	DT	Dimil.
Low dim.:													
abalone	0.7458	0.7581	0.7537	0.7493	0.7470	0.7461	0.7524	0.7582	0.7610	0.7484	0.7524	0.7511	0.7494
breast	0.0409	0.0472	0.0469	0.0452	0.0438	0.0432	0.0455	0.0418	0.0398	0.0412	0.0386	0.0372	0.0409
glass	0.2822	0.3159	0.2879	0.2832	0.2692	0.2710	0.2981	0.3271	0.3000	0.2832	0.2720	0.2776	0.3131
heart	0.1822	0.1785	0.1733	0.1719	0.1696	0.1696	0.1859	0.2015	0.1874	0.1778	0.1770	0.1674	0.1726
magic	0.1346	0.1340	0.1314	0.1309	0.1302	0.1300	0.1329	0.1328	0.1323	0.1338	0.1326	0.1298	0.1336
pblocks	0.0288	0.0285	0.0265	0.0271	0.0268	0.0261	0.0282	0.0302	0.0296	0.0286	0.0269	0.0263	0.0402
phoneme	0.1332	0.1277	0.1252	0.1261	0.1256	0.1264	0.1260	0.1232	0.1258	0.1291	0.1271	0.1248	0.1301
pima	0.2385	0.2492	0.2484	0.2411	0.2432	0.2424	0.2503	0.2516	0.2596	0.2385	0.2375	0.2414	0.2398
wine	0.0393	0.0461	0.0382	0.0303	0.0404	0.0393	0.0629	0.0551	0.0607	0.0393	0.0371	0.0360	0.0348
yeast	0.4008	0.4155	0.4054	0.3985	0.4034	0.4013	0.4116	0.4142	0.4189	0.4011	0.3978	0.4018	0.4116
Avg. Low	0.2227	0.2301	0.2237	0.2204	0.2199	0.2196	0.2294	0.2336	0.2315	0.2221	0.2199	0.2193	0.2266
High dim.:													
ionosphere	0.1459	0.1527	0.1413	0.1458	0.1430	0.1430	0.1584	0.1532	0.1646	0.1476	0.1430	0.1413	0.1464
optdigits	0.0329	0.0337	0.0327	0.0327	0.0318	0.0313	0.0367	0.0352	0.0351	0.0329	0.0284	0.0279	0.0721
pendigits	0.0156	0.0174	0.0152	0.0140	0.0140	0.0138	0.0171	0.0150	0.0162	0.0156	0.0129	0.0126	0.0160
sat	0.1021	0.1067	0.1027	0.0997	0.0986	0.1005	0.1044	0.1010	0.1005	0.1022	0.0967	0.0971	0.1040
segment	0.0336	0.0334	0.0319	0.0304	0.0316	0.0302	0.0318	0.0326	0.0336	0.0330	0.0309	0.0306	0.0345
sonar	0.2269	0.2404	0.2183	0.2077	0.2077	0.2058	0.2163	0.2337	0.2452	0.2260	0.2183	0.2163	0.2231
spambase	0.0587	0.0569	0.0559	0.0555	0.0539	0.0546	0.0576	0.0573	0.0574	0.0579	0.0554	0.0549	0.0574
texture	0.0307	0.0343	0.0312	0.0304	0.0291	0.0285	0.0343	0.0330	0.0336	0.0308	0.0268	0.0270	0.0325
vehicle	0.2726	0.2773	0.2664	0.2690	0.2664	0.2674	0.2671	0.2690	0.2693	0.2723	0.2641	0.2600	0.2721
waveform	0.1492	0.1554	0.1490	0.1503	0.1489	0.1479	0.1508	0.1535	0.1533	0.1498	0.1468	0.1472	0.1532
Avg. High	0.1068	0.1108	0.1045	0.1036	0.1025	0.1023	0.1075	0.1084	0.1109	0.1068	0.1023	0.1015	0.1111
Avg. All	0.1647	0.1704	0.1641	0.1620	0.1612	0.1609	0.1684	0.1710	0.1712	0.1644	0.1611	0.1604	0.1689

(see Secs. 2.1.2 and 2.1.3). Finally, we showed some interpretability aspects of the proposed fuzzy linguistic combination method.

For the comparison, apart from the standard MV, we selected average (AVG) [Kun04] and decision templates (DT) [KBD01] based on Euclidean distance, as crisp and fuzzy fusion methods respectively, being the best methods of each group according to Kuncheva [Kun03]. Since the proposed FRBCS-CM includes classifier selection and classifier fusion, we also applied classifier selection with the mentioned classifier fusion methods in order to make a fair comparison. To select classifiers we used two standard greedy approaches, Greedy Forward Selection (FS) and Greedy Backward Selection (BS) [RG05], which consider the abovementioned classifier fusion methods (these methods are also used to guide the search of the greedy algorithms). The hybrid method based on GA proposed in [DVA09] (GA-Dimililer) embeds both classifier selection and classifier fusion, thus we directly apply it without any modifications.

For illustrative purpose, Tables I.20 and I.21 present a comparison between FRBCS-CM (interpretable GFS) and the other CE combination methods in terms of accuracy and complexity, respectively. The whole study described can be found in Sec. 3 in Part II of this PhD dissertation. Table I.20 shows the test error obtained for MV (operating on the full original ensemble), FRBCS-CM (nonzero values= 10%, 25%, 50%, 75%, and 90%), Greedy FS with MV, AVG, and DT, Greedy BS with MV, AVG, and DT, and GA-Dimililer. Then, Table I.21 reports the total number of rules in the ensembles considering the same approaches. The comparison was conducted with respect to a similar complexity in the obtained FRBCEs. For example, FRBCS-CM with nonzero values 10% and 25% were compared to Greedy FS with MV, AVG, and DT, clearly outperforming them.

Finally, to illustrate the interpretability capabilities of FRBCS-CM, we show how it works on the wine dataset. The fuzzy rule base obtained with FRBCS-CM 10% on this dataset is presented in Figure 7.

Table I.21: Complexity of the original FRBCEs, FRBCS-CM, and the other CE combination methods in terms of the number of rules

Dataset	fuzzy CEs	FRBCS-CM 10%	25%	50%	75%	90%	Greedy FS MV	AVG	DT	Greedy BS MV	AVG	DT	GA Dimil.
Low dim.:													
abalone	3990.9	398.2	995.7	1996.9	2983.6	3578.4	1211.0	1047.6	1037.7	2711.3	3306.9	3398.7	2391.9
breast	435.2	46.1	110.9	217.0	326.2	391	33.0	25.7	24.1	415.9	426.6	427.4	221.1
glass	590.3	57.4	140.6	289.9	434.4	528	88.7	43.6	54.7	560.5	576.4	577.5	173.8
heart	466.0	49.4	120.3	235.3	352.6	421	48.9	35.7	33.4	444.6	455.7	454.7	221.1
magic	3882.1	421.0	968.3	1965.6	2969.9	3475.8	528.2	424.6	417.3	2247.8	3203.6	3319	2123.6
pblocks	1329.4	131.2	328.9	628.1	967.8	1182.2	248.2	108.9	106.1	1259	1288	1297.3	314.1
phoneme	2197.3	241.7	587.8	1132.5	1679.0	2000	493.2	381.1	339.4	1442.8	2046	2049.4	996.9
pima	1050.9	110.9	260.7	530.1	782.4	946	239.3	149.4	118.1	957	1025	1027.7	530
wine	231.4	23.7	57.9	116.4	172.7	208	9.1	6.8	6.2	222.4	226.9	226.9	71.2
yeast	2449.0	260.8	630.9	1198.4	1825.1	2198.4	511.5	389.5	434.9	1901.3	2296.7	2291.9	902.4
Avg. Low	1662.3	174.0	420.2	831.0	1249.4	1492.8	341.1	261.3	257.2	1216.3	1485.2	1507.1	794.6
High dim.:													
ionosphere	367.7	37.8	95.4	211.0	279.8	334	27.0	22.2	24.4	353.3	361.2	360.6	190.3
optdigits	3584.6	359.2	893.5	1787.7	2678.8	3227.2	652.7	428.7	423.7	3398.5	3513.8	3513.1	661.5
pendigits	4395.3	448.8	1098.1	2208.7	3299.9	3964.3	892.1	569.8	470.8	4167.2	4306.4	4307.5	1874.6
sat	4207.2	427.2	1046.9	2107.2	3128.1	3762.8	1214.0	728.7	800.6	3575.2	4006.8	4055	1431.9
segment	1175.3	130.1	290.9	593.4	876.9	1051.4	165.6	109.2	86.7	1100.5	1151.3	1151.4	414.2
sonar	319.3	32.4	80.4	162.0	240.0	288	24.4	22.9	19.8	306.4	312.1	311.9	158.8
spambase	2220.9	229.0	557.2	1115.5	1661.7	2002.6	340.7	286.1	292.8	2135.5	2152.4	2139.8	1026
texture	2912.2	300.1	716.6	1458.8	2175.0	2610.9	433.6	333.8	352.5	2759.8	2852.9	2852.8	1240.4
vehicle	1415.3	154.3	380.4	735.3	1075.3	1283	364.1	173.3	193.4	1304.7	1387.6	1380	425.7
waveform	3484.3	354.0	861.5	1749.8	2601.2	3137.6	1355.9	753.1	727.1	3125.9	3408.3	3381.1	828.9
Avg. High	2408.2	247.3	602.1	1212.9	1801.7	2166.1	547.0	342.8	339.2	2222.7	2345.3	2345.3	825.2
Avg. All	2035.2	210.7	511.1	1022.0	1525.5	1829.4	444.1	302.0	298.2	1719.5	1915.2	1926.2	809.9

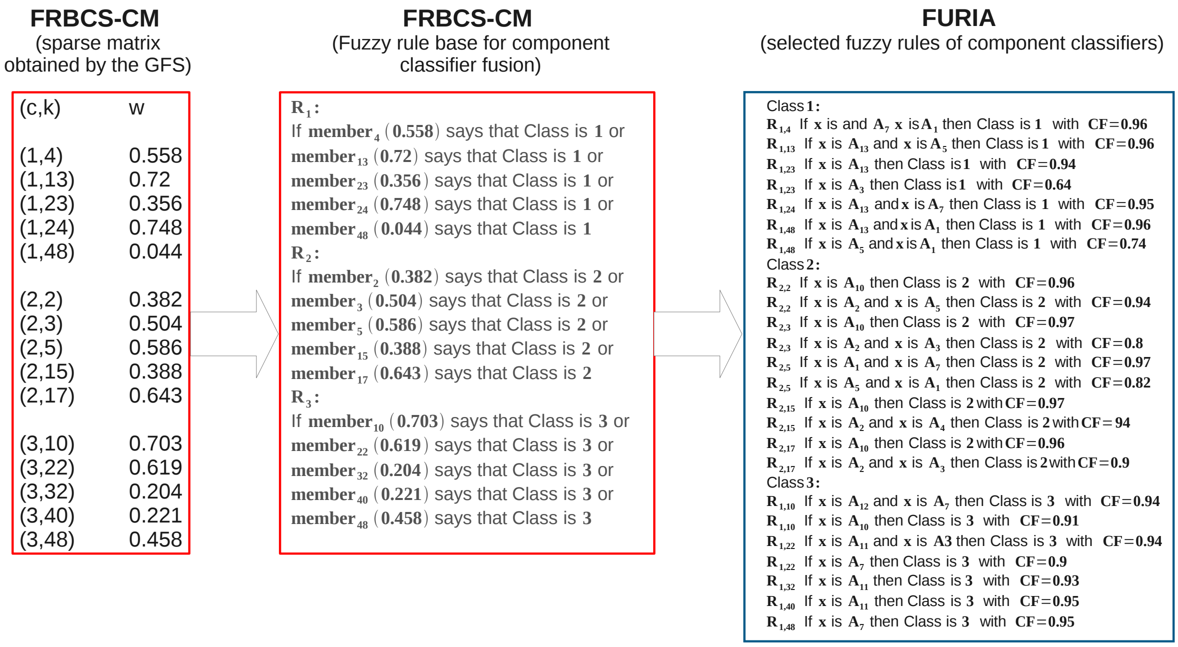


Figure 7: An example showing how FRBCS-CM selects and combines fuzzy rules of the selected component FURIA-based fuzzy classifiers. The wine dataset was used for illustration with FRBCS-CM considering 10% of the non-zero values.

3.3.3 Evolutionary Multiobjective Overproduce-and-Choose dynamic classifier selection

This section presents an OCS strategy for the classifier selection of our *dynamic* FRBCEs, the RO-based bagging FRBCEs (see Section 3.2.2). On the one hand, the main aim is again to refine the accuracy-complexity trade-off in the RO-based bagging FRBCEs when dealing with high dimensional and complex classification problems. On the other hand, an other interesting objective is to study whether the additional diversity induced by ROs is beneficial for the EMO OCS-based FRBCEs. Thus, we have again chosen the state-of-the-art NSGA-II EMO algorithm in order to generate good quality Pareto set approximations. In this approach, we propose a specific design customized to the RO characteristics.

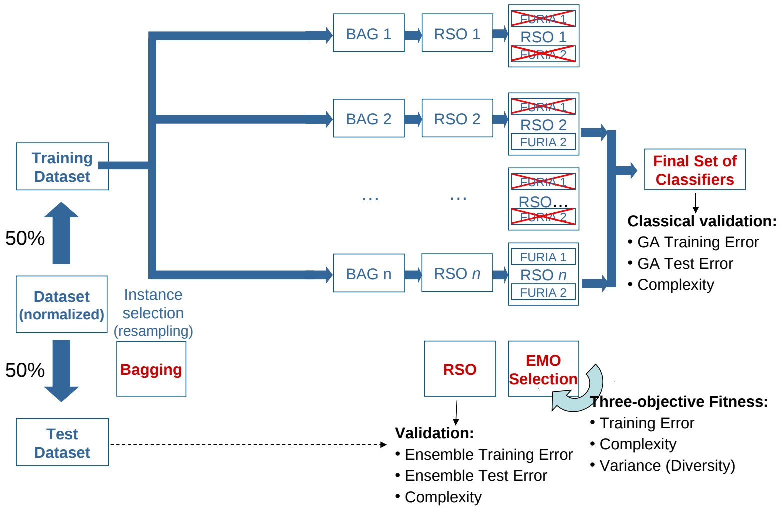


Figure 8: The dynamic EMO OCS framework: after obtaining bootstrapped replicas, the component classifiers are derived by the specific RO method (either RLO or RSO) using FURIA as subclassifiers. Then, the OCS takes place by means of NSGA-II with a three-objective fitness function providing a Pareto set of simplified FRBCEs. In every case, the output is obtained using a voting-based method.

In this study [TCQS13a], we take one step further and use a three-objective fitness function combining the three existing kinds of optimization criteria for classifier selection: accuracy, complexity, and diversity. We use the following measures: an advanced accuracy measure based on three different components (accuracy), the total number of fuzzy rules in the ensemble (complexity), and the difficulty measure θ (diversity) [TPC05, RG05, KW03]. Notice that, in order to make a fair comparison, we consider the final complexity in terms of the total number of rules instead of the total number of classifiers, since RO-based classifiers produce twice as much classifiers and usually they are less complex than a standard base classifier.

Table I.22: The different variants resulting from the three EMO approaches used for the classifier selection

abbreviation	base classifier	CE methodology	OCS strategy	mut. type
BAS-BAG	FURIA	bagging	standard NSGA-II	standard
BAS-RLO	RLO (2×FURIA+oracle)	bagging+RLO	standard NSGA-II	standard
ADV-RLO	RLO (2×FURIA+oracle)	bagging+RLO	specific RO NSGA-II	standard
ADV-BI-RLO	RLO (2×FURIA+oracle)	bagging+RLO	specific RO NSGA-II	biased
BAS-RSO	RSO (2×FURIA+oracle)	bagging+RSO	standard NSGA-II	standard
ADV-RSO	RSO (2×FURIA+oracle)	bagging+RSO	specific RO NSGA-II	standard
ADV-BI-RSO	RSO (2×FURIA+oracle)	bagging+RSO	specific RO NSGA-II	biased

RO offers a tremendous advantage over a standard component classifier since each classifier can be independently selected within each pair component. Because of that, our classifier selection is done at the level of the component classifiers and not at the whole pair of classifiers. A specific coding scheme, which permits that none, one, or both FURIA fuzzy subclassifiers can be selected, is introduced. We also develop a reparation operator, whose objective is to correct the unfeasible solutions. Fig. 8 shows the final structure of the RO-based bagging FRBCE design methodology including the OCS stage.

We have considered two different mutation operator settings. The first one is the standard bit-flip mutation, while the second is the bit-flip mutation with biased probabilities proposed in [INN05]. The aim of the latter is to positively bias the complexity reduction in the classifier selection process.

Table I.23: Comparison of Pareto fronts using the HVR measure

	BAS-BAG	BAS-RLO	ADV-RLO	ADV-BI-RLO	BAS-RSO	ADV-RSO	ADV-BI-RSO
aba	0.8248	0.8594	0.6399	**0.8878**	0.8378	0.7305	0.8500
bio	0.8343	0.9073	0.8059	**0.9825**	0.9115	0.9118	0.9678
coi	0.6929	0.7419	0.5687	**0.7548**	0.7251	0.6497	0.6477
gas	0.8590	0.9404	0.6876	**0.9771**	0.9382	0.8435	0.9642
iso	0.8611	0.9118	0.7661	**0.9534**	0.9074	0.8571	0.9155
let	0.9127	0.9477	0.7961	0.9726	0.9626	0.8945	**0.9727**
mag	0.7970	0.8423	0.6444	**0.9061**	0.8433	0.8119	0.8737
mar	0.7214	0.8217	0.6569	**0.8689**	0.8170	0.7994	0.8225
mfa	0.8874	0.9463	0.7886	**0.9763**	0.9439	0.8717	0.9600
mfo	0.8373	0.8838	0.7145	**0.9322**	0.8809	0.8040	0.8931
mka	0.8661	0.9227	0.7643	**0.9631**	0.9091	0.8418	0.9211
mze	0.8041	0.8650	0.6498	**0.9183**	0.8560	0.7702	0.8660
mus	0.7112	0.8098	0.6161	**0.8779**	0.8172	0.7071	0.8122
opt	0.8721	0.9316	0.7662	**0.9669**	0.9322	0.8411	0.9415
pbl	0.7487	0.7794	0.6038	0.7231	0.8052	0.7764	**0.8421**
pen	0.8617	0.9375	0.6873	**0.9752**	0.9419	0.8106	0.9609
rin	0.8187	0.8526	0.6878	0.8803	0.9221	0.8954	**0.9222**
sat	0.8436	0.9219	0.7196	**0.9613**	0.9284	0.8296	0.9468
seg	0.8551	0.9081	0.7621	**0.9358**	0.9080	0.8172	0.8417
sen	0.8597	0.9234	0.6630	**0.9644**	0.9228	0.8043	0.9503
shu	0.9347	0.9192	0.7051	0.9645	0.9176	0.7858	**0.9661**
spa	0.8196	0.8932	0.6805	**0.9343**	0.8690	0.8535	0.9109
ste	0.8206	0.8836	0.6620	**0.9264**	0.8877	0.7998	0.9053
tex	0.8713	0.9308	0.7769	**0.9614**	0.9288	0.8388	0.9444
thy	0.8368	0.9084	0.6804	**0.9560**	0.9025	0.8303	0.9487
two	0.8774	0.9558	0.7478	**0.9814**	0.9392	0.8880	0.9565
wan	0.8566	0.8881	0.7335	**0.9397**	0.8873	0.8400	0.8890
wav	0.8426	0.9033	0.7163	**0.9300**	0.8989	0.8367	0.9192
wqu	0.7914	0.8567	0.6973	**0.9098**	0.8724	0.8119	0.8881
avg.	0.8317	0.8894	0.7031	**0.9269**	0.8901	0.8191	0.9035
dev.	0.0562	0.0522	0.0608	0.0618	0.0524	0.0564	0.0681

We compared the proposed NSGA-II for RLO- and RSO-based bagging FRBCEs classifier selection with the standard NSGA-II using two different approaches from the first stage, namely RLO- and RSO-based bagging FRBCEs as well as bagging FRBCEs. Table I.22 summarizes the seven resulting EMO OCS-based FRBCE design approaches.

We conducted exhaustive experiments considering 29 datasets with different characteristics concerning a high number of examples, features, and classes from the UCI [BM98] machine learning and KEEL [AFFL$^+$11] repositories (see Table I.5 in Sec. 3.2.2). For validation we used 5×2-cv. To compare the Pareto front approximations of the global learning objectives (i.e. CE test accuracy and complexity) we considered the most common multiobjective metric, HVR [CLV07]. We also analyzed single solutions extracted from the obtained Pareto front approximations. We compared the three EMO variants in order to check whether the additional diversity induced by the RO is beneficial to the performance of the final FRBCE selected by NSGA-II.

To give a brief view to the results obtained, we present the most representative ones as

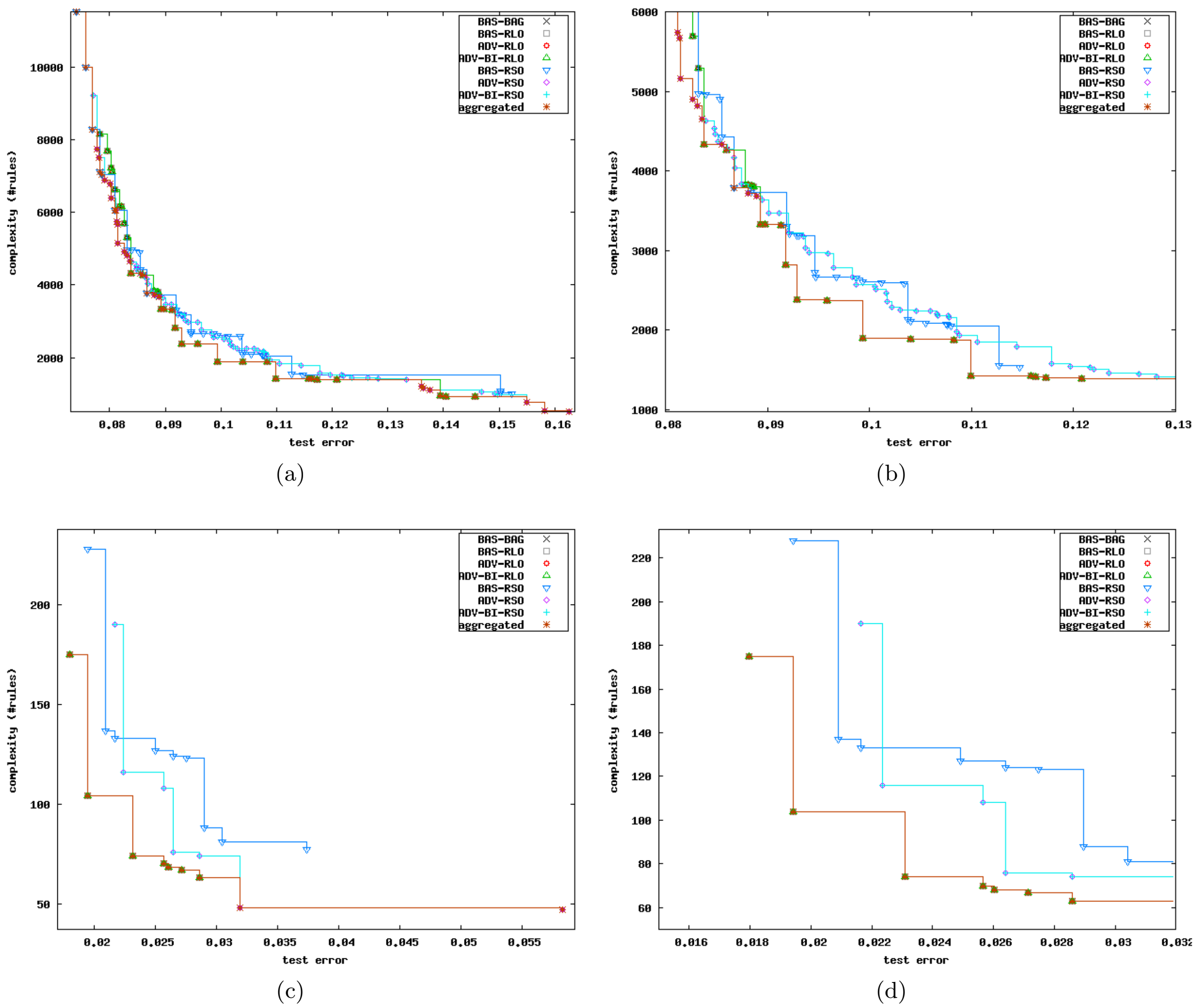


Figure 9: Graphical representations of the Pareto front approximations obtained from the three EMO approaches for two datasets: (a) letter, (b) letter (zoom), (c) sensor_read_24, and (d) sensor_read_24 (zoom). Objective 1 stands for test error and objective 2 for complexity in terms of the number of rules. The pseudo-optimal Pareto front is also drawn for reference.

Table I.24: A comparison of the averaged performance of the four single solutions selected from the obtained Pareto sets

	Best complx				Best diversity				Best train				Best trade-off (tra-div-cmpl)			
	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst
avg. BAS-BAG	445	0.1392	0.0694	0.1399	558	0.1371	0.0695	0.1396	2003	0.2173	0.0506	0.1306	647	0.1554	0.0640	0.1317
BAS-RLO	202	0.0874	0.0983	0.1736	303	0.0842	0.0962	0.1726	1748	0.2176	0.0436	0.1301	404	0.1151	0.0667	0.1381
ADV-RLO	1058	0.0203	0.0525	0.1314	3044	0.0165	0.0476	0.1268	2663	0.0172	0.0435	0.1267	2078	0.0172	0.0473	0.1235
ADV-BI-RLO	90	0.0390	0.0916	0.1680	1606	0.0163	0.0474	0.1271	1269	0.0173	0.0426	0.1276	480	0.0185	0.0489	0.1264
BAS-RSO	205	0.0904	0.0926	0.1688	308	0.0874	0.0905	0.1682	1883	0.2243	0.0420	0.1292	402	0.1201	0.0646	0.1384
ADV-RSO	587	0.2633	0.0523	0.1307	778	0.1818	0.0514	0.1290	2164	0.6111	0.0409	0.1271	836	0.2228	0.0501	0.1248
ADV-BI-RSO	115	0.1180	0.1417	0.2114	670	0.0542	0.0628	0.1446	1463	0.3331	0.0392	0.1308	414	0.0624	0.0635	0.1380
dev. BAS-BAG	512	0.0987	0.1500	0.1858	687	0.0987	0.1495	0.1852	3505	0.1539	0.1390	0.1832	731	0.1088	0.1478	0.1833
BAS-RLO	222	0.0668	0.1511	0.1925	496	0.0640	0.1418	0.1881	2976	0.1565	0.1213	0.1821	541	0.0845	0.1396	0.1832
ADV-RLO	1267	0.0230	0.1319	0.1845	4234	0.0203	0.1274	0.1831	3508	0.0210	0.1206	0.1828	2745	0.0209	0.1276	0.1812
ADV-BI-RLO	96	0.0395	0.1535	0.1928	2218	0.0202	0.1272	0.1830	1645	0.0211	0.1196	0.1829	542	0.0221	0.1309	0.1818
BAS-RSO	205	0.0688	0.1472	0.1906	499	0.0662	0.1366	0.1867	3461	0.1622	0.1194	0.1820	454	0.0878	0.1372	0.1837
ADV-RSO	686	0.1195	0.1303	0.1845	927	0.0871	0.1307	0.1842	3269	0.3958	0.1156	0.1829	989	0.1020	0.1288	0.1818
ADV-BI-RSO	120	0.0665	0.1462	0.1870	764	0.0483	0.1287	0.1833	2235	0.2822	0.1135	0.1839	519	0.0517	0.1375	0.1838

Table I.25: Average Rankings of the Friedman's test

Algorithm	Ranking
ADV-RLO	1.603
ADV-RSO	2.138
ADV-BI-RLO	3.345
BAS-BAG	3.707
BAS-RSO	5.603
BAS-RLO	5.638
ADV-BI-RSO	5.966

Table I.26: The adjusted p-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE is the control method (FURIA is the base classifier in every case)

Comparison	**p-value**
ADV-RLO vs ADV-BI-RSO	**8.89e-014**
ADV-RLO vs BAS-RLO	**5.73e-012**
ADV-RLO vs BAS-RSO	**7.11e-012**
ADV-RLO vs BAS-BAG	**0.0006**
ADV-RLO vs ADV-BI-RLO	**0.0043**
ADV-RLO vs ADV-RSO	0.3461

Table I.27: A comparison of RLO-based bagging CEs using FURIA, C4.5 and NB, as well as random forests in terms of accuracy

Dataset	**ADV-RLO Test err.**	**FURIA+BAG+RLO Test err.**	**C4.5+BAG+RLO Test err.**	**RF Test err.**
abalone	**0.7425**	0.7452	0.7666	0.7536
bioassay_688red	0.0091	**0.0090**	**0.0090**	**0.0090**
coil2000	0.0603	0.0601	0.0612	**0.0597**
gas_sensor	**0.0075**	0.0079	0.0097	0.0092
isolet	0.0693	**0.0691**	0.0803	0.0766
letter	0.0852	0.0742	**0.1559**	0.0701
magic	0.1285	0.1314	**0.1254**	0.1314
marketing	**0.6620**	0.6673	0.6728	0.6624
mfeat_fac	**0.0427**	0.0434	0.0484	0.0475
mfeat_fou	**0.1825**	0.1941	0.1932	0.1858
mfeat_kar	0.0655	0.0699	0.0766	**0.0597**
mfeat_zer	**0.2108**	0.2169	0.2285	0.2330
musk2	0.0297	0.0328	**0.0271**	0.0375
optdigits1	**0.0270**	0.0283	0.0290	0.0277
pblocks	0.0336	0.0353	**0.0333**	0.0332
pendigits	**0.0127**	0.0137	0.0155	0.0162
ring_norm	**0.0409**	0.0438	0.0558	0.0587
sat	0.0980	0.1008	**0.0953**	0.1027
segment	**0.0263**	0.0303	0.0338	0.0350
sensor_read_24	**0.0218**	0.0227	0.0228	0.0224
shuttle	**0.0006**	0.0009	0.0009	0.0009
spambase	**0.0604**	0.0651	0.0650	0.0625
steel_faults	0.2293	0.2367	**0.2265**	0.2517
texture	**0.0262**	0.0278	0.0348	0.0383
thyroid	0.0218	**0.0215**	0.0222	0.0221
two_norm	**0.0260**	0.0271	0.0266	0.0389
waveform	**0.1422**	0.1461	0.1630	0.1556
waveform1	**0.1430**	0.1451	0.1599	0.1587
wquality_white	0.3762	0.3840	**0.3714**	0.3864
avg.	**0.1235**	0.1259	0.1314	0.1292
dev.	0.1812	0.1825	0.1844	0.1830

follows (the whole study is reported in Sec. 4 of Part II). Table I.23 shows the results using the HVR metric, while the average and standard deviation values for the four different solutions selected from each Pareto front approximation in the 29 problems are shown in Table I.24. Statistical tests for those results are presented in Tables I.25 and I.26. Besides, the aggregated Pareto fronts for the letter and sensor_read_24 datasets are represented graphically in Figure 9, which allows an easy visual comparison of the performance of the different EMO OCS-based FRBCEs variants. Finally, Table I.27 reports a final comparison, where the ADV-RLO variant is compared to RO-based bagging FRBCEs (full original ensemble) as well as to the classical RO-based bagging CE approaches using C4.5 and random forests. The statistical tests carried out for the results obtained in the previous table are shown in Tables I.28 and I.29.

From this study two main conclusions can be emphasized. Firstly, the competitive performance in terms of accuracy obtained by the proposed NSGA-II cand be observed. Secondly, the additional diversity induced by the ROs to the base classifier is beneficial for the final performance of the FRBCEs designed.

Table I.28: Average Rankings of the Friedman's test

Algorithm	Ranking
ADV-RLO	1.586
FURIA+BAG+RLO	2.534
RF	2.879
C4.5+BAG+RLO	3.000

Table I.29: The adjusted p-values of Holm test for the pair-wise comparisons where RLO-based bagging FRBCE (using FURIA) is the control method

Comparison	**p-value**
ADV-RLO vs C4.5+BAG+RLO	**9.13e-005**
ADV-RLO vs RF	**2.73e-004**
ADV-RLO vs FURIA+BAG+RLO	**0.0051**

3.4 Topology-based WiFi Indoor Localization - a Real World Application

People localization is required for many novel applications like for instance proactive care for the elders or people suffering degenerative dementia such as Alzheimer's disease. In this section, we introduce a system for people localization in indoor environments. It is based on a topology-based WiFi signal strength fingerprint approach. Accordingly, it is a robust, cheap, ubiquitous and nonintrusive system which does require neither the installation of extra hardware nor prior knowledge about the structure of the environment under consideration. The localization task thus turns into a high dimensional classification task. The well-known curse of dimensionality critically emerges when dealing with complex environments like the current one. Therefore, the core of the proposed framework [TAH13] is a FRBCE considering fuzzy logic to deal with the huge uncertainty that is characteristic of WiFi signals, and based on the classical methodologies for CE design as bagging and random subspace.

The main goal is to obtain a scalable and accurate localization system which can estimate the closest reference location to the actual user location using received signal strength in a relatively short time. To do so, we considered a generic CE approach, using both classical and fuzzy base classifiers. Two different methodologies, bagging [Bre96] and bagging combined with random subspace [PD07], are exploited to design the final CE-based localization system (as it is detailed in Section 3.2.1). First, the base classifiers are learnt off-line from a fingerprint database previously generated, and then the CE-based framework is run on-line.

A flow of our design is as follows. The training set is submitted to an instance selection procedure, and (optionally) to a feature selection procedure, in order to provide individual training sets (bags) to train the base classifiers (in off-line mode).

The combination of classifier members within the ensemble (on-line mode) is made by the so-called classifier fusion method [WKB97], which aggregates the results provided by the set of component classifiers to calculate the final output, assuming that all classifiers are trained over the entire feature space. The Decision Profile (DP) represents the outputs of all the classifiers in the ensemble [Kun01b, Kun04]:

$$DP(\mathbf{x}) = \begin{pmatrix} D_1(\mathbf{x}) \\ \vdots \\ D_L(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} d_{1,1}(\mathbf{x}) & \cdots & d_{1,c}(\mathbf{x}) \\ \vdots & & \vdots \\ d_{L,1}(\mathbf{x}) & \cdots & d_{L,c}(\mathbf{x}) \end{pmatrix} \tag{I.3}$$

where c is the number of classes; L is the number of classifiers; and $d_{i,j}(\mathbf{x})$ are the confidence degrees for the classes given an example $\mathbf{x}$. Considering L classifiers, the combined output is usually computed by an algebraic function [KHDM98, Kun02] such as maximum, minimum, product, mean, median, etc.

To deal with the inherent noise that characterizes the WiFi signal in indoor environments, we propose an elaborated framework encapsulating a CE in order to improve the robustness of the whole system. A global schema of the proposed framework is made up of the three following phases:

- **Phase1 - Classification process of each classifier component.**

  In this phase the classification task of each CE components is carried out. Each classifier for each instance from 1 to N outputs confidence degrees d_{ij}^m for each class. Thus, the N matrices, namely DPs, are generated to be provided as the input required for the Phase 2.

- **Phase2 - Filtering (Aggregation 1).**

  The filtering phase takes place at the classifier output level. The confidence degrees d_{ij}^m of N instances are aggregated for each classifier d_{ij}^*. The aggregation is done by means of one of the (algebraic) functions mentioned above. Then, the aggregated DP of the CE is provided as an output. Notice that, the filtering follows the "moving average" fashion, in every step the DP of the next example is included in the aggregation of DPs, while excluding the first DP appeared in the given period of time.

- **Phase3 - Classifier fusion (Aggregation 2).**

  In the last phase, a second aggregation is performed. The aggregated DP is combined by means of one of the abovementioned algebraic functions (mean, median, etc.). As a result, the outputs of all the individual classifiers d_{ij}^* are merged into one final decision c'.

It is worth noting that, the framework described above is only applicable for the on-line execution mode of our WiFi location system, while the core of the CE is trained in an off-line mode, starting from a fingerprint database previously generated.

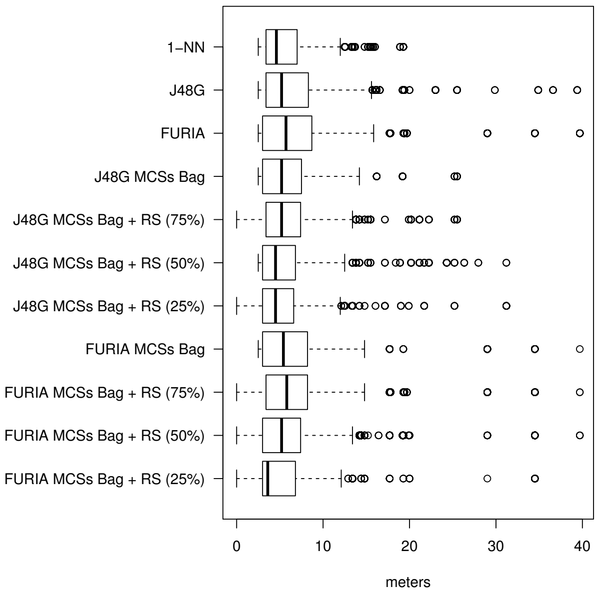
Our design (considering both J48G [Web99] CEs and FURIA FRBCEs) is composed of 10 classifiers (because in some preliminary trials on Scenario 1, we observed that considering a larger number of classifiers was not yielding a significant increase of accuracy for the analyzed problem), while RS selects a subset of features containing 25%, 50%, or 75% of the initial feature set. It is compared with the classical k-Nearest Neighbor (k-NN) classification technique [CH67].

In addition, we chose mean as the most common aggregation method in both stages (Aggr1 and Aggr2). The second aggregation stage (Aggr2) only takes place in the case of the designed CEs, where the 10 individual classifier outputs are fused. The final decision is done using the maximum activation degree. Notice that, with the structure of basic k-NN method Aggr2 makes no sense.

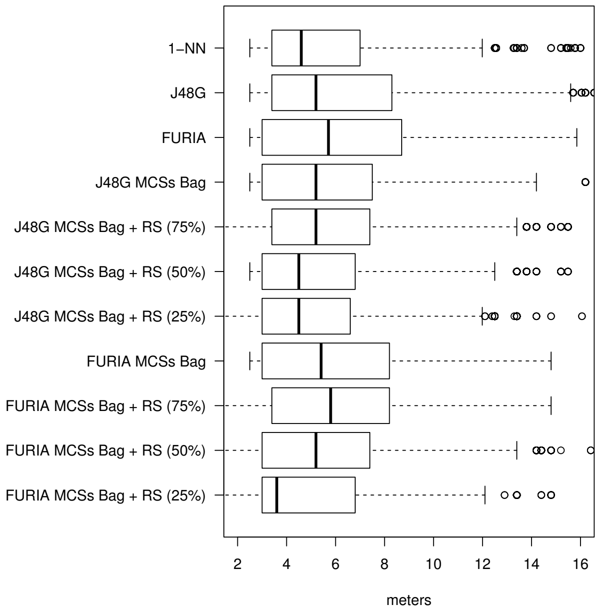
In this approach, we took an advantage of the time-dependent characteristics of the data, namely the signals of APs being obtained in a consecutive order for each location. In our scenario, the user stops for a few seconds to acquire several consecutive WiFi measures with the aim of getting better estimation of its current position, which is the global task of the classification process. In order to avoid any loss or distortion of the data, for the identification of a given position we consider a consecutive block of samples.

Table I.30: Accuracy results for different classification and aggregation methods in Scenario 2 (UAH environment)

Algorithm		N1	N4	N7	N10
1-NN		0.490	0.501	0.523	0.536
J48G		0.566	0.578	0.586	0.589
FURIA		0.534	0.564	0.581	0.595
J48G CEs	Bag	0.644	0.679	0.690	0.694
	Bag + RS (75%)	0.657	0.697	0.711	0.720
	Bag + RS (50%)	0.697	0.735	0.749	0.757
	Bag + RS (25%)	0.731	0.776	0.789	0.797
FURIA CEs	Bag	0.624	0.667	0.680	0.688
	Bag + RS (75%)	0.675	0.715	0.726	0.734
	Bag + RS (50%)	0.723	0.769	0.785	0.794
	Bag + RS (25%)	**0.733**	**0.790**	**0.803**	**0.809**

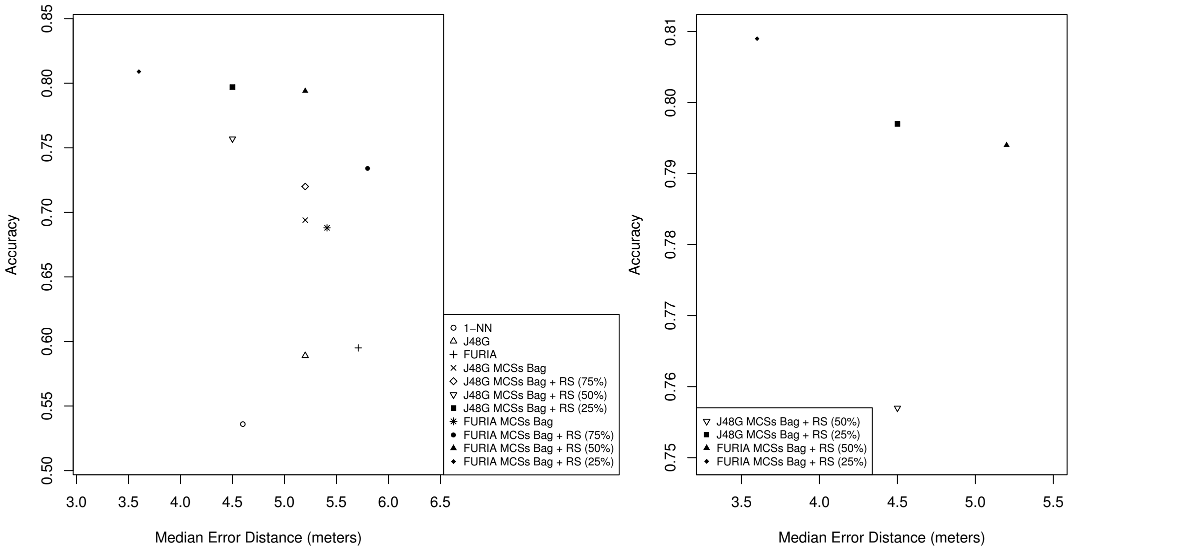


(a) Block size N10



(b) Block size N10 (zoomed)

Figure 10: Reported results (considering block size N10) in terms of error distances (in meters) for the misclassified positions in Scenario 2 (UAH test-bed environment).



(a) Accuracy vs Median Error Distance

(b) Accuracy vs Median Error Distance (zoomed)

Figure 11: Comparison of algorithms used in Scenario 2 (UAH test-bed environment). Accuracy (y-axis) versus Median Error Distance (x-axis).

Several values of the block size N were selected, i.e. 1, 4, 7, and 10 corresponding to around 1, 4, 7, and 10 seconds respectively because our WiFi acquisition frequency was 1Hz.

We have conducted a comprehensive experiment on two real test-bed environments: a simple scenario considering only one corridor at the European Centre for Soft Computing, which is composed of 39,000 instances, 4 features, and 13 classes (locations). The second scenario, which considers the second and the third floors of the Polytechnic School at the University of Alcalá, is much more realistic (also more complex) composed of 8,520 instances, 143 (134) features, and 71 classes (locations).

To give a brief view to the results obtained, we present those obtained for Scenario 2 (see Sec. 5 in Part II for the whole study). Table I.30 reports the achieved results in terms of accuracy for all the selected block sizes. Then, Figure 10 depicts a dispersion of the error distance (in meters) for each algorithm evaluated by means of boxplots (the possible outliers are represented by circles), while Figure 11 presents the median values of the error distances represented in Fig. 10 (x-axis) against the accuracy values reported in Table I.30 (y-axis), for all the eleven algorithms evaluated in the experiments. Overall, the FURIA FRBCE proposed in Sec. 3.2.1 provides the best performance.

4 Discussion of the Results Obtained

The current section summarizes the main results obtained in this PhD dissertation. The next six subsections will be devoted to analyze the main outcomes derived from the work developed.

4.1 Static Fuzzy Rule-Based Classifier Ensembles Design from Classical Data Mining Approaches

As our aim was to obtain FRBCEs dealing with high dimensional datasets, we have proposed a methodology in which a bagging approach is used together with a feature selection technique to train FURIA-based fuzzy component classifiers for a FRBCE. We used a single winner-based classifier fusion method on top of the base classifiers. We tested FURIA-based FRBCEs with bagging, feature selection, and the combination of both of them. The main conclusions obtained are as follows:

- The proposed FURIA-based FRBCEs showed to be *accurate* and capable to directly be 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 to design the weak learners.
- The application of bagging for the CE design resulted in an approach being able to generate the classifiers in parallel, thus being *time efficient*.
- FURIA-based FRBCEs with bagging clearly outperformed FURIA-based FRBCEs with feature selection and FURIA-based FRBCEs with bagging and feature selection. Thus, it is the recommended FRBCE design approach. It seems that the feature selection capability direcly incorporated by the FURIA method makes a good combination with the bagging approach.
- A FRBCE framework based on a quick and accurate fuzzy classification rule learning algorithm, namely FURIA, proved to be competitive if not better than two state-of-the-art classical CEs such as random forests and bagging C4.5 decision trees.

This study has resulted in the following scientific publications:

- K. Trawiński, O. Cordón, and A. Quirin. A First Study on a Fuzzy Rule-Based Multiclassification System Framework Combining FURIA with Bagging and Feature Selection, In Proceedings of the World Conference on Soft Computing (WConSC), San Francisco (USA), pp. 167-175, 2011.
- K. Trawiński, O. Cordón, and A. Quirin. On Designing Fuzzy Rule-based Multiclassification Systems by Combining FURIA with Bagging and Feature Selection, International Journal of Uncertainty Fuzziness, and Knowledge-based Systems, vol. 19, no 4, pp. 589-633, 2011. DOI: 10.1142/S0218488511007155. Impact factor: 1.781. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 31/111. Q2.

4.2 Dynamic Fuzzy Rule-Based Classifier Ensembles Design from Advanced Data Mining Approaches

Taking the general FRBCE design methodology as a base, we have integrated two ROs, namely the RLO and RSO approaches, into the bagging FURIA-based FRBCEs. By doing so we aimed to improve the diversity of the FRBCEs and thus increase their accuracy thanks to the appealing characteristics of that dynamic CE design approach. The following outcomes were obtained from this study:

- Both RO-based bagging FRBCEs outperform bagging FRBCEs in terms of accuracy and complexity.

- Comparing the two different RO approaches, RSO obtains slightly higher accuracy but also a higher complexity, while RLO does the opposite (slightly lower accuracy and slightly lower complexity).
- RSO-based bagging FRBCEs outperform classical RSO-based bagging CEs using C4.5 and NB.

The work carried out resulted in a JCR journal paper and two conference articles:

- K. Trawiński, O. Cordón, and A. Quirin. Random oracles fuzzy rule-based multiclassifiers for high complexity datasets. In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Hyderabad (India), pp. 1-8, 2013.
- K. Trawiński, O. Cordón, and A. Quirin. On Applying Random Oracles to Fuzzy Rule-Based Classifier Ensembles for High Complexity Datasets. The 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), Milan (Italy), 2013. DOI: 10.2991/eusflat.2013.92.
- K. Trawiński, O. Cordón, L. Sánchez, and A. Quirin. Multiobjective Genetic Classifier Selection for Random Oracles Fuzzy Rule-Based Multiclassifiers: How Beneficial is the Additional Diversity?, Knowledge-based Systems, In press, 2013. DOI: 10.1016/j.knosys.2013.08.006. Impact factor 2012: 4.104. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 6/115. Q1.

4.3 Static Evolutionary Multi Objective Overproduce-and-Choose Strategy

We have introduced a two-stage method to design FRBCEs based on the use of bagging FURIA-based FRBCEs and an EMO-OCS method for classifier selection by means of NSGA-II. Five different biobjective fitness functions were tested, considering the three existing sets of optimization criteria for classifier selection: accuracy, complexity, and diversity. We summarize the most important conclusions obtained point by point below:

- Comparing the obtained Pareto Front approximations using the HVR metric, the fitness function composed of training error (accuracy) and variance (diversity) clearly reported the best performance. Meanwhile, the combinations of variance (diversity) with the number of classifiers (complexity) and double fault (diversity) with the number of classifiers (complexity) turned out to be deceptive.
- NSGA-II bagging FURIA-based FRBCEs showed to be competitive with the static bagging FURIA-based FRBCEs and classical CEs such as random forests and bagging C4.5 decision trees in terms of accuracy.
- In addition, the proposed method proved to be a good approach to obtain high quality, well performing ensembles with a good accuracy-complexity trade-off when dealing with high dimensional datasets.

The scientific article associated to this part are listed as follows:

- 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. 9th International Conference on Intelligent Systems Design and Applications (ISDA), Pisa (Italy), pp. 121-127, 2009.

- K. Trawiński, O. Cordón, and A. Quirin. A Study on the Use of Multiobjetive Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers, International Journal of Computational Intelligence Systems, vol. 4, no 2, pp. 231-253, 2012. DOI: 10.1080/18756891.2012.685272

4.4 Interpretable Genetic Fuzzy System for Joint Classifier Selection and Fusion

A novel CE combination method was developed based on the use of an FRBCS automatically derived by means of a GA. This new GFS-based fuzzy linguistic combination method shows very interesting characteristics, especially its transparency and its capability to jointly perform classifier fusion and selection. In addition, when combined with a FRBCE, the overall system shows a hierarchical structure (called stacking in the literature).

This study was carried in a three-fold manner. Firstly, as a preliminary analysis we compared bagging FRBCEs combined with interpretable FRBCS-CM with the whole initial FRBCEs using a greedy classifier selection algorithm abd standard MV as fusion method. Secondly, we compared the novel interpretable GFS with state-of-the-art crisp and fuzzy CE combination methods, as well as with a hybrid method based on GAs considering both classifier selection and classifier fusion [DVA09]. Finally, we showed some interpretability aspects of the proposed fuzzy linguistic combination method. This study led us to very interesting outcomes:

- Bagging FRBCEs combined with FRBCS-CM obtained good results in comparison with bagging FRBCEs with the full ensemble using standard MV. Apart from obtaining good performance in terms of accuracy, it was also very competitive in terms of complexity reduction, after the selection of the component classifiers. We noticed that the final results highly depended on the value of the parameter defining the complexity of the FRBCS-CM, which leads to different complexity-accuracy trade-offs.
- The proposal allows the user to specify the reduction of the complexity of the final CE *a priori* by selecting the desired non zero parameter value. This high flexibility, an *a priori* choice of how simple the obtained CE will be, constitutes an advantage over the compared approaches.
- The proposed fuzzy linguistic combination method provides a good degree of interpretability to the CE, making the combination method operation more transparent for the user. Furthermore, when combined with a FRBCE, the whole system takes a hierarchical fuzzy classification system structure (in the sense that the weak learners constitute individual FRBCSs becoming the input to the FRBCS-based combination method). The type of rules with a a class and a certainty degree in the consequent used in our FRBCS-CM allows the user to get an understandable insight to the CE, thus providing interpretability of such complicated system to some extent.

The developments in this research line has been published in a JCR journal paper and an international conference article:

- L. Sánchez, O. Cordón, A. Quirin, K. Trawiński, Introducing a Genetic Fuzzy Linguistic Combination Method for Bagging Fuzzy Rule-Based Multiclassification Systems. Fourth IEEE International Workshop on Genetic and Evolving Fuzzy Systems (GEFS), Mieres (Spain), pp. 75-80, 2010.

- K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez. A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers, IEEE Transactions of Fuzzy Systems, vol. 21, no 5, pp. 950-965, 2013, 2013. DOI: 10.1109/TFUZZ.2012.2236844. Impact Factor 2012: 5.484. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 1/115. Q1.

4.5 Dynamic Evolutionary Multi Objective Overproduce-and-Choose Strategy

We have designed an EMO OCS method for dynamic RO-based bagging FRBCEs. We used NSGA-II with a specific binary coding for the RO-based classifier selection. A three-objective fitness function was used composed of three different optimization criteria: accuracy (considering an advanced accuracy measure), complexity, and diversity metrics.

We have conducted exhaustive experiments comparing seven EMO variants in order to check whether the additional diversity induced by the RO is beneficial to the performance of the final FRBCE selected by the NSGA-II-based OCS method (see Table I.22 in Sec. 3.3.3 for the abbreviations of the seven EMO OCS-based FRBCE design approaches tested). We employed two RO approaches, RLO and RSO to test the new proposal. From the results obtained we drew the following conclusions:

- The results obtained with the HVR metric corroborated the initial assumption that the additional diversity provided by the RO approach is beneficial for the FRBCEs designed. The best performing approaches were based on both RLO and RSO (ADV-BI-RLO and ADV-BI-RSO) using specific NSGA-II and biased mutation. Notice that, the HVR metric considered in the comparison measures the overall quality of the Pareto front approximations obtained with respect to the two global learning goals, accuracy and complexity. Hence, the EMO OCS methods performing a stronger component classifier reduction are promoted.

- The ADV-RLO, ADV-RSO, and ADV-BI-RLO variants outperformed the standard BAS-BAG variant considering test accuracy, which showed a good behavior of the proposed approach.

- The biased mutation obtained very good results in terms of complexity, as it significantly reduced the number of rules in the final FRBCEs. Considering the best complexity it managed to significantly decrease the number of rules by more than 90% on average for both ADV-BI-RLO and ADV-BI-RSO.

- In general, the proposed NSGA-II approaches with three learning objectives derived good quality solutions, which were widely spread among the Pareto front. They reached both extents acquiring high performance for the two main learning goals: accuracy (ADV-RSO and ADV-RLO) and complexity (ADV-BI-RSO and ADV-BI-RLO).

- The best individual performance in terms of test accuracy was obtained by the ADV-RLO variant, even though it obtained quite weak Pareto front approximations. This fact is justified by the HVR metric nature, as already mentioned.

- When comparing the ADV-RLO variants with non-selected CEs, that is RLO-based bagging FRBCEs (the full original ensemble) and the classical RLO-based bagging CEs using C4.5 decision trees and Random Forests, the proposed approaches turned out to be the best performing, while strongly reducing the complexity.

The obtained results were disclosed in a paper published in a JCR journal:

- K. Trawiński, O. Cordón, L. Sánchez, and A. Quirin. Multiobjective Genetic Classifier Selection for Random Oracles Fuzzy Rule-Based Multiclassifiers: How Beneficial is the Additional Diversity?, Knowledge-based Systems, In press, 2013. DOI: 10.1016/j.knosys.2013.08.006. Impact factor 2012: 4.104. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 6/115. Q1.

4.6 A real-world application: A Topology-Based WiFi Indoor Localization Problem

The last step made in this research line concerning FRBCE design for high dimensional classification problems was to apply at least one of the proposed approaches into a real-world problem. This goal was achieved by using bagging FURIA-based FRBCEs, optionally combined with RS feature selection, to two different scenarios of a topology-based WiFi indoor localization problem: 1) a simple but highly illustrative case, and 2) a realistic high dimensional case. Its performance was tested in comparison with some classical CEs. Overall, we concluded that:

- In Scenario 1, the goal was to check the proposed framework in the context of a rather simple case as the one defined in the selected corridor of the European Centre for Soft Computing premises. Most of the evaluated algorithms were able to achieve very high accuracy. Considering different combinations of parameters, it was hard to point out a single one. Anyway, we could appreciate how the proposed framework achieved very good results for all the classifiers used. This fact is due to the inherent simplicity of the analyzed scenario.

- Considering Scenario 2, which considers the second and the third floors of the Polytechnic School at the University of Alcalá, we noticed that the reported accuracy significantly decreased in comparison with the results obtained in Scenario 1. FURIA-based FRBCEs with bagging and RS (25%) outperformed the other algorithms for this high dimensional dataset. From these facts, we could confirm the need of adopting the CE-based approach in order to properly deal with high dimensional problems arising from complex environments like this scenario. Moreover, fuzzy methods like FURIA exhibited all their potential in the context of very noisy problems where classical methods did not perform so well. This is due to the appealing characteristics of the fuzzy rules generated by FURIA.

This study contributed in a JCR journal paper and conference article:

- P. Menendez, C. Campomanes, K. Trawiński, J. M. Alonso. Topology-based indoor localization by means of WiFi fingerprinting with a computational intelligent classifier. In Proceedings of the 11th IEEE International Conference on Intelligent System Design and Applications (ISDA), Córdoba (Spain), pp. 1020-1025, 2011.

- K. Trawiński, J. M. Alonso, and N. Hernandez. A Multiclassifier Approach for Topology-based WiFi Indoor Localization, Soft Computing, vol. 17, no 10, pp. 1817-1831, 2013. DOI 10.1007/s00500-013-1019-5. Impact factor 2012: 1.124. Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 63/115. Q3.

5 Final Conclusions and Future Works

In this PhD dissertation, we have proposed a global framework for FRBCE design in order to allow FRBCSs to deal with high dimensional datasets. Our proposal is composed of different methods for component fuzzy classifier derivation, which consider several classical and recently proposed CE methodologies, as well as EAs for classifier selection and fusion. In addition, we proposed a linguistic FRBCS-CM based on GFSs, which jointly develop classifier selection and fusion, allowing interpretability of FRBCEs to some extent.

We have carried out exhaustive experiments for each specific FRBCE design derived from our framework. Furthermore, we have applied one of those designs on a topology-based WiFi indoor localization real-world problem involving a high dimensional classification task. The results obtained showed that we have reached the global goal. Besides, we also achieved the different subobjectives defined at the beginning of this PhD dissertation such as improvement of the FRBCE performance in terms of accuracy and accuracy-complexity trade-off. The good outcomes obtained are reflected in five scientific journal publications and several conference papers.

In addition, the emerged research line during the development of this PhD dissertation and the promising results obtained lead to several interesting future works that could be elaborated:

- A straightforward future work is to combine Bagging RO-based FRBCEs with the interpretable FRBCS-CM proposed. This challenging task would consist of a dynamic FRBCE in the first stage and a combined classifier selection and fusion by means of an interpretable GFS in the second stage.
- Another interesting way to follow is to incorporate an EMO algorithm, that determines the complexity of the generated FRBCEs, to the FRBCS-CM. The behavior of this combination method strongly depends on a single parameter, whose value is set by the user in advance. To avoid this problem, an EMO algorithm, e.g. the well known NSGA-II, could be applied. Therefore, the user will obtain a Pareto set of FRBCE designs with different complexity-accuracy trade-offs as the final output.
- The topology-based WiFi indoor localization is a demanding task. It turns into a high dimensional classification problem when dealing with complex environments. Furthermore, WiFi signals are characterized by a huge uncertainty. Thus, we think that applying the remaining FRBCE design methods could improve the performance of the results obtained and extend the development of this research line.
- FRBCEs could be applied to some other real-world problems. For example, an interesting field growing in the literature, where the proposed FRBCE designs fit, is imbalanced classification. The class-imbalance problem [CJA04], also named as learning with imbalanced datasets in the literature, basically characterizes a significant difference between the number of examples of one class in comparison to the number of examples from the other classes. Typically a minority class is much more difficult to be correctly classified as well as it is of special interest not to commit errors on this class. Imbalanced datasets commonly arise in applications such as risk management [HHJ06], medical diagnosis [MHZ^{+}08], and face recognition [LC07].
- There are many general CE techniques from different families such as dynamic classifier selection, classifier fusion, stacking, mixture of experts, or diversity induction methods that could be combined with FRBCSs in order to both allow FRBCSs to deal with high dimensional datasets and to improve their performance.

6 Conclusiones Finales y Trabajos Futuros

En esta tesis doctoral, se ha propuesto un marco global de diseño de FRBCEs para permitir a los FRBCSs manejar conjuntos de datos de alta dimensionalidad. Nuestra metodología se compone de diferentes métodos para generar FRBCSs, que consideran varias metodologías clásicas y recientes de diseño de CEs, así como de técnicas basadas en EAs para la selección y fusión de los clasificadores base. Además, hemos propuesto un FRBCS-CM linguístico basado en GFSs que realiza conjuntamente la selección y la fusión de clasificadores base y permite dotar de interpretabilidad a los FRBCEs en cierta medida.

Hemos desarrollado experimentos exhaustivos para cada diseño específico de FRBCE derivado de nuestro marco de trabajo. Además, hemos aplicado uno de esos diseños a un problema real, consistente en la localización en interiores utilizando topología WiFi, que se corresponde con un problema de clasificación de alta dimensionalidad.

Los resultados obtenidos demuestran que hemos tenido éxito en el objetivo global planteado en esta tesis doctoral. Además, también se han alcanzado los distintos subobjetivos definidos al inicio de la misma, como la mejora del rendimiento de los FRBCEs en términos tanto de precisión como de un equilibrio adecuado entre precisión y complejidad. Los buenos resultados obtenidos se reflejan en cinco publicaciones en revistas científicas y varios artículos en congresos.

Además, la línea de investigación que ha surgido con el desarrollo de esta tesis doctoral conduce a una serie de nuevos desarrollos interesantes que podrían ser desarrollados en un futuro próximo:

- Una extensión sencilla sería combinar FRBCEs basados en ROs con el método FRBCS-CM interpretable propuesto. Esta tarea consistiría en considerar un FRBCE dinámico en la primera etapa y una combinación de selección y fusión de clasificadores mediante un GFS interpretable en la segunda.

- Otra idea interesante sería incorporar un algoritmo EMO al FRBCS-CM para determinar la complejidad de los FRBCEs generados de forma automática. El comportamiento de este método de combinación depende en gran medida de un único parámetro, cuyo valor es fijado por el usuario con antelación. Para evitar este problema, se podría aplicar un algoritmo EMO, por ejemplo, el conocido NSGA-II. De este modo, el usuario obtendría un conjunto de Pareto de diseños de FRBCEs con distintos equilibrios entre complejidad y precisión en cada ejecución del método.

- La localización indoor WiFi basada en topología es una tarea exigente. Cuando se afronta un entorno complejo, se corresponde con un problema de clasificación de alta dimensionalidad. Además, presenta la dificultad adicional de la incertidumbre asociada a la naturaleza de las señales WiFi considerada. Por lo tanto, creemos que la aplicación del resto de métodos de diseño de FRBCEs podría mejorar los resultados obtenidos y ampliar el desarrollo de esta línea de investigación.

- Nuestros FRBCEs podrían aplicarse también a otros problemas reales. Por ejemplo, un campo interesante de la literatura en el que podrían encajar los diseños de FRBCEs propuestos es la clasificación no balanceada. El problema del desbalanceo de clases [CJA04], también llamado clasificación con conjuntos de datos no balanceados, básicamente caracteriza una diferencia significativa entre el número de ejemplos de una clase en comparación con el número de ejemplos de las clases restantes. Normalmente, una clase minoritaria es mucho más difícil de clasificar correctamente, al igual que es de especial interés no cometer errores en dicha clase. Los conjuntos de datos no balanceados surgen en aplicaciones como la gestión de riesgos [HHJ06], el diagnóstico médico [MHZ^{+}08] y el reconocimiento de caras [LC07].

- Hay muchas otras técnicas genéricas englobadas en diferentes familias de métodos clásicos de diseño de CEs como la selección dinámica de clasificadores, la fusión de clasificadores, el stacking, la mezcla de expertos o los métodos de inducción de diversidad que podrían combinarse con FRBCSs con el fin de permitir a los FRBCSs tratar con conjuntos de datos de alta dimensión y mejorar su rendimiento.

Part II. Publications

This chapter presents a complete copy of all the scientific papers published. Together they show the work carried out to achieve the stated objectives in this PhD dissertation. Five sections present each of the contribution developed.

1 On Designing Fuzzy Rule-based Multiclassification Systems by Combining FURIA with Bagging and Feature Selection

The first journal paper obtained for this PhD dissertation is:

- K. Trawiński, O. Cordón, and A. Quirin. On Designing Fuzzy Rule-based Multiclassification Systems by Combining FURIA with Bagging and Feature Selection, International Journal of Uncertainty Fuzziness, and Knowledge-based Systems, vol. 19, no 4, pp. 589-633, 2011. DOI: 10.1142/S0218488511007155.
  - State: Published.
  - Impact Factor (JCR): 1.781.
  - Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 31/111. Q2.

International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 19, No. 4 (2011) 589–633

DOI: 10.1142/S0218488511007155

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.

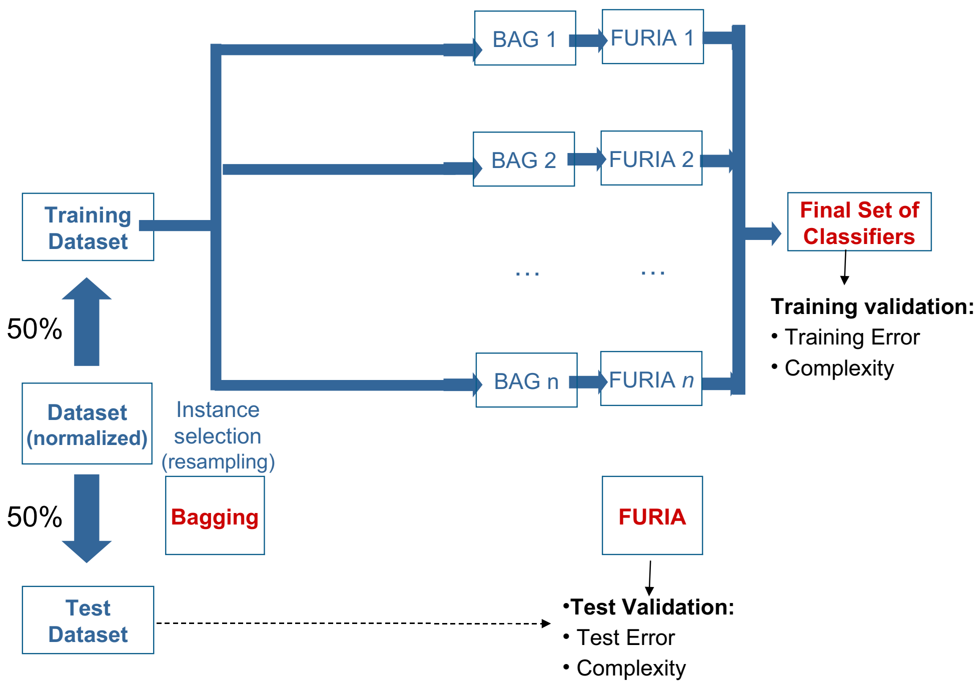


Fig. 1. Our framework: after the instance and the feature selection procedures, the component fuzzy classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based combination method.

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 $\times$ 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

	aba	bre	gla	hea	ion	let	mag	opt	pbl	pen
FURIA-based MCSs										
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

	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
FURIA-based MCSs											
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.

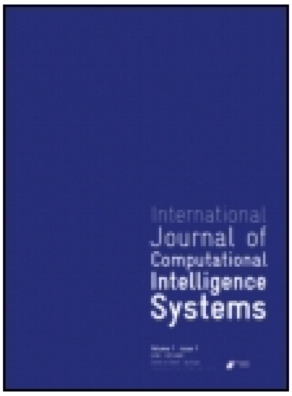
32. Y. Nojima and H. Ishibuchi, Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design, *Int. J. Hybrid Intelligent Systems* **4**(3) (2007) 157–169.
33. D. Optiz and R. Maclin, Popular ensemble methods: an empirical study, *J. Artificial Intelligence Res.* **11** (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.

2 A Study on the Use of Multiobjetive Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers

The next journal paper we managed to publish is:

- K. Trawiński, O. Cordón, and A. Quirin. A Study on the Use of Multiobjetive Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers, International Journal of Computational Intelligence Systems, vol. 4, no 2, pp. 231-253, 2012. DOI: 10.1080/18756891.2012.685272
  - State: Published.
  - In 2012 removed from the JCR list.



International Journal of Computational Intelligence Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcis20

A Study on the Use of Multiobjective Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers

Krzysztof Trawiński, Oscar Cordón & Arnaud Quirin
[a] European Centre for Soft Computing, Mieres, Spain
[b] Edificio Científico-Tecnológico, Calle Gonzalo Gutiérrez Quirós S/N, 33600, Mieres, Asturias

Available online: 23 Apr 2012

To cite this article: Krzysztof Trawiński, Oscar Cordón & Arnaud Quirin (2012): A Study on the Use of Multiobjective Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers, International Journal of Computational Intelligence Systems, 5:2, 231-253

To link to this article: http://dx.doi.org/10.1080/18756891.2012.685272

International Journal of Computational Intelligence Systems, Vol. 5, No. 2 (April, 2012), 231-253

A Study on the Use of Multiobjective Genetic Algorithms for Classifier Selection in FURIA-based Fuzzy Multiclassifiers

Krzysztof Trawiński , Oscar Cordón , Arnaud Quirin

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

Abstract

In a preceding contribution, we conducted a study considering a fuzzy multiclassifier system (MCS) design framework based on Fuzzy Unordered Rule Induction Algorithm (FURIA). It served as the fuzzy rule classification learning algorithm to derive the component classifiers considering bagging and feature selection. In this work, we integrate this approach under the overproduce-and-choose strategy. A state-of-the-art evolutionary multiobjective algorithm, namely NSGA-II, is used to provide a component classifier selection and improve FURIA-based fuzzy MCS. We propose five different fitness functions based on three different optimization criteria, accuracy, complexity, and diversity. Twenty UCI high dimensional datasets were considered in order to conduct the experiments. A combination between accuracy and diversity criteria provided very promising results, becoming competitive with classical MCS learning methods.

Keywords: Fuzzy rule-based multiclassification systems, bagging, FURIA, genetic selection of individual classifiers, diversity measures, evolutionary multiobjective optimization, NSGA-II

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 [1]. 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 [2, 3, 4, 5, 6, 7, 8].

In our previous studies [9, 10, 11, 12], we proposed a MCS methodology based on classical MCS design techniques such as bagging and feature selection with a fuzzy rule-based system (FRBCS) as a base classifier. As a consequence, fuzzy rule-based multiclassification systems (FRBMCSs) were incorporated into an overproduce-and-choose strategy (OCS) [13]. This MCS desing algorithm is based on the generation of a large number of component classifiers, and a subsequent selection of the subset of them best cooperating. As the main tool we used a multicriteria genetic algorithm (GA) for static component classifier selection 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. The resulting FRBMCS design approach thus belong to the genetic fuzzy systems (GFSs) family, one of the most successful approaches to hybridize fuzzy systems with learning and adaptation methods in the last fifteen years [14, 15, 16].

In [17] we extended our previous developments by proposing a fuzzy MCS framework based on Fuzzy Unordered Rule Induction Algorithm (FURIA) [18, 19] as the fuzzy rule classification learning algorithm to derive the component classifiers considering bagging and feature selection. We conducted comprehensive experiments with 20 datasets taken from the UCI machine learning repository and provided a deep study of the results obtained. Several FURIA-based fuzzy MCS composition designs were 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) [21], and the random-greedy feature selection based on MIFS and the GRASP approach [22]. Finally, our approach was compared against two state-of-the-art MCS algorithms (random forests and bagging decion trees) and also with an application of the fuzzy MCS generation approach with other, less powerful fuzzy classifier derivation method [23, 9]. From the obtained results, we drew the conclusion that FURIA-based fuzzy MCSs were a very powerful tool for dealing with high dimensional classification problems.

Even so, we think that the performance of the latter FURIA-based MCS framework can be improved with an OCS approach based on an evolutionary multiobjective (EMO) algorithm [24] considering diversity measures. In the current work we integrate FURIA-based fuzzy MCSs within the OCS strategy. Since there are many optimization criteria considered for MCS design such as accuracy, complexity, and diversity measures [1, 25, 26, 27], the use of a EMO algorithm came naturally to our mind.

In this paper, we study the behavior of FURIA-based fuzzy MCSs with large size ensembles. To do so, we consider the state-of-the-art NSGA-II algorithm [28] to perform classifier selection. Introducing diversity and complexity measures combined with error measures is an interesting approach, which has led to promising results in the area [12, 25, 26, 27, 29, 30]. Hence, we have embedded three measures of this kind in the objective space of the fitness function combining them with an accuracy index, which resulted in five different bicriteria fitness functions.

We think that such GFS may lead to high quality fuzzy MCSs with a good accuracy-complexity tradeoff. To check this assumption, we present experiments on twenty high dimensional datasets from the UCI machine learning repository.

This paper is set up as follows. In the next section, a state of the art about MCSs, fuzzy MCSs, and MCS selection is presented. Sec. 3 recalls FURIA and our approach for designing FURIA-based fuzzy MCSs, while Sec. 4 describes the proposed NSGA-II for component classifier selection focusing on the different two-objective fitness functions to be considered. 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 a FRBMCS. The techniques used to generate MCSs and fuzzy MCSs are described in Sec. 2.1 and 2.2, respectively. Some ways to reduce the size of the ensembles are described in Sec. 2.3. The use of GAs for this purpose is then explored in Sec. 2.4.

2.1. Related work on MCSs

A MCS is the result of the combination of the outputs of a group of individually trained classifiers in order to get a system that is usually more accurate than any of its single components [1]. 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 [31, 32], although they can be used with any other type of classifiers (the use of neural networks is also very extended, see for example [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* [34], they are independently learnt from resampled training sets ("bags"), which are randomly selected with replacement from the original training data set. *Boosting* methods [35] 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 [36]. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features [27, 37]. *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*, such as *random forests* [38].

The interested reader is referred to [32, 33] for two reviews for the case of decision tree (both) and neural network ensembles (the latter), including exhaustive experimental studies.

2.2. Previous Work on Fuzzy MCSs

Focusing on fuzzy MCSs, the use of boosting for the design of fuzzy classifier ensembles has been considered in some works [2, 3, 39, 40]. However, only a few contributions for bagging fuzzy classifiers have been proposed considering fuzzy neural networks (together with feature selection) [41], neuro-fuzzy systems [5], and fuzzy decision trees [8, 7] as component classifier structures.

Especially worth mentioning is the contribution [8]. This approach hybridizes Breimann's idea of random forests [38] with fuzzy decision trees [42]. 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.

Some advanced GFS-based contributions should also be remarked. On the one hand, an FRBCS ensemble design technique is proposed in [43] 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 [44, 6]. 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 [45], where an EMO algorithm generated a Pareto set of FRBCSs with different accuracy-complexity tradeoffs to be combined into an ensemble.

2.3. Determination of the Optimal Set of Component Classifiers in the MCS

Typically, an ensemble of classifiers is post-processed in such a way only a subset of them are kept for the final decision. It is a well known fact that the size of this MCS is an important issue for its tradeoff between accuracy and complexity [32, 33] and that most of the error reduction occurs with the first few additional classifiers [34, 33]. Furthermore, the selection process also participates in the elimination of the duplicates or the poor-performing classifiers.

While in the first studies on MCSs a very small number (around ten) of component classifiers was considered as appropriate to sufficiently reduce the test set prediction error, later research on boosting (that also holds for bagging) suggested that error can be significantly reduced by largely exceeding this number [46]. This has caused the use of very large ensemble sizes (for example comprised by 1,000 individual classifiers) in the last few years [32].

Hence, the determination of the optimal size of the ensemble is an important issue for obtaining both the best possible accuracy in the test data set without overfitting it, and a good accuracy-complexity tradeoff. In pure bagging and boosting approaches, the optimal ensembles are directly composed of all the individual classifiers generated until a specific stopping point, which is determined according to different means (validation data set errors, likelihood, ...). For example, in [32] it is proposed an heuristic method to determine the optimal number guided by the *out-of-bag* error.

However, there is the chance that the optimal ensemble is not comprised by all the component classifiers first generated but on a subset of them carrying a larger degree of disagreement/diversity. This is why different classifier selection methods [47] have been proposed. GAs have been commonly used for this task as we will show in the following subsection.

2.4. Related work on genetic selection of MCSs

In general, the selection of a subset of classifiers is done using the OCS strategy [13, 30], in which a large set of classifiers is produced and then selected to extract the best performing subset. GAs are a popular technique within this strategy. In the literature, performance, complexity and diversity measures are usually considered as search criteria. Complexity measures are employed to simplify the system, whereas diversity measures are used to avoid overfitting. The reader is referred to [10] for a review on these genetic MCS selection approaches.

Among the different genetic OCS methods, we can remark these most related to the current proposal. On the one hand, we find EMO-based approaches such as that in [48], a hierarchical multiobjective GA algorithm, performing feature selection at the first level and classifier selection at the second level, is presented which outperforms classical methods for two handwritten recognition problems. The multiobjective GA allows both performance and diversity to be considered for MCS selection. Another EMO proposal for classifier selecion is presented in [29]. In that contribution, a comparison of a single-objective GA and the NSGA-II EMO algorithm for 14 different objective functions of the mentioned three families of criteria (12 diversity measures, the training error, and the number of classifiers as a complexity measure). The authors applied their study on only one dataset, a digit handwritten recognition problem with 10 classes and 118,735 instances. They concluded saying that the training error is the best criterion for a single GA and a combination of training error and one diversity measure is the best criterion for an EMO algorithm, which supports the developments in the current contribution (see Sec. 6). On the other hand, in [26] a genetic classifier selection method was considered based on a single performance index, either the diversity, including 16 different measures, or the ensemble error. The best results were obtained with the accuracy measure and a specific kind of diversity measures correlated with the error.

3. Bagging FURIA-based fuzzy MCSs

In this section we will detail how the FURIA fuzzy MCSs are designed [17]. 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 an initial 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. FURIA is reviewed in Sec. 3.1, while the instance selection procedure is described in Sec. 3.2.

3.1. FURIA

Fuzzy Unordered Rules Induction Algorithm (FURIA) [18, 19] is an extension of the state-of-the-art rule learning algorithm called RIPPER [49], having its advantages such like simple and comprehensible

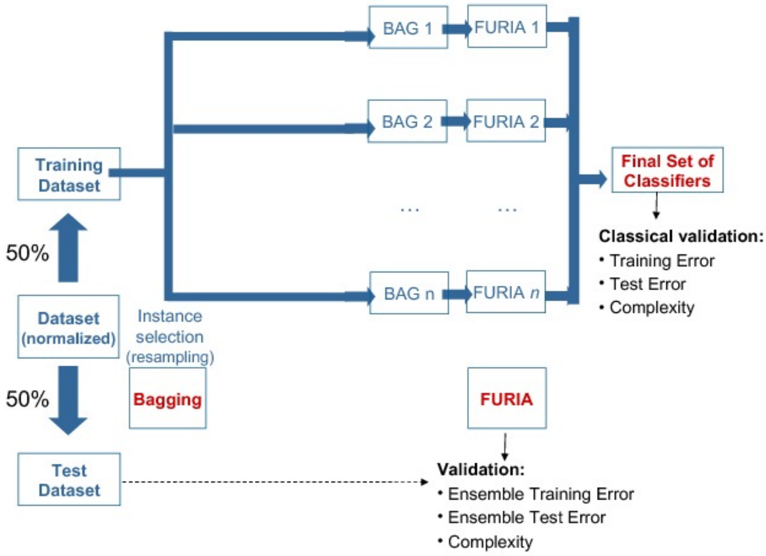


Figure 1: Our framework: after the instance and the feature selection procedures, the component classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based combination method.

fuzzy rule base, and introducing new features. FURIA provides three different extensions of RIPPER:

- It takes an advantage of fuzzy rules instead of crisp ones. Fuzzy rules of FURIA are composed of a class C_j and a certainty degree CD_j in the consequent. The final form of a rule is the following:

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

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

$$CD_j = \frac{2\frac{D_T^{C_j}}{D_T} + \sum_{x\in D_T^{C_j}} \mu_r^{C_j}(x)}{2+\sum_{x\in D_T} \mu_r^{C_j}(x)} \quad (1)$$

  where D_T and $D_T^{C_j}$ stands for the training set and a subset of the training set belonging to the class C_j respectively. 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. Hence, the fuzzy reasoning method used is the so-called voting-based method [50, 51].

- It uses unordered rule sets instead of rule lists. This change ommits a bias caused by the default class rule, which is applied whenever there is an uncovered example detected.

- It proposes a novel rule stretching method in order to manage uncovered examples. The unordered rule set introduces one crucial drawback, there might appear a case when a given example is not covered. Then, to deal with such situation, one rule is generalized by removing its antecedents. The information measure is proposed to verify which rule to "stretch".

The interested reader is referred to [18] for a full description of FURIA.

3.2. FURIA-based fuzzy MCS design approaches

In our previous contribution [17] we conducted comprehensive experiments considering FURIA-based fuzzy MCSs. Since in [52] it was shown that a combination between bagging and feature selection usually leads to good MCS designs, we decided to follow that idea and we integrated FURIA into a framework of that kind. We aimed to combine the diversity induced by the MCS algorithms and the robustness of the FURIA method in order to desing good performance fuzzy MCS for high dimensional problems. By doing so, we wanted to obtain FURIA-based fuzzy MCSs with a good accuracy-complexity tradeoff. We also tried a combination of FURIA with bagging and feature selection separately.

Three different feature selection methods, random subspace [20], mutual information-based feature selection (MIFS) [21], and the random-greedy feature selection based on MIFS and the GRASP approach [22], were considered. For each feature selection algorithm three different feature subsets of different sizes, which were based on the initial number of features in the classification problem, were tested. Finally, our experiments showed that out of the three following MCS methodologies, that is bagging, feature selection, and bagging with feature selection, the former (see Fig. 1) obtained the best performance when combined with FURIA-based FRBCSs.

Thus, in this contribution we are applying directly a bagging approach in order to generate the initial FURIA-based fuzzy MCSs, which will be later selected by the EMO algorithm. Considering this approach, the bags are generated with the same size as the original training set, as commonly done.

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.

4. EMO-based MCS selection method

The second stage of our methodology is to consider the OCS strategy. Our aim is to obtain a good accuracy-complexity tradeoff in the FURIA-based fuzzy MCSs when dealing with high dimensional problems. That is, we aim to obtain fuzzy MCS with a low number of base classifiers, which keep a good accuracy. Thus, we have selected the state-of-the-art NSGA-II EMO algorithm in order to generate good quality Pareto set approximations. Five different biobjective fitness functions combining the three existing kinds of optimization criteria (accuracy, complexity, and diversity) are proposed in order to study the best setting. Fig. 2 shows the final structure of the FURIA-based fuzzy MCS design methodology including the OCS stage. The two subsections below presents briefly the algorithm operation mode and its main components.

4.1. Components of NSGA-II

NSGA-II [28] is based on a Pareto dominance depth approach, where the population is divided into several fronts and the depth of each front shows to which front an individual belongs to. A pseudo-dominance rank being assigned to each individual, which is equal to the front number, is a metric used for the selection of an individual.

We have used a standard binary coding in such a way that a binary digit/value/gene is assigned to each classifier. Then, when the variable takes value 1, it means that the current component classifier belongs to the final ensemble, while when the variable is equal to 0, that classifier is discarded. This approach provides a low operation cost, which leads to the high speed of the algorithm.

We have used a generational approach and elitist replacement strategy. The initial population is composed of randomly generated individuals, keeping one of them with the original fuzzy MCS composed of all the existing classifiers. In each generation, to introduce a high amount of diversity, a binary tournament is performed. That means that two individuals are randomly picked from the current population and the best one is selected. The two winners are crossed over to obtain a single offspring that directly substitutes the loser. We have considered the classical two-point crossover and the simple bit-flip mutation. Both operators crossover and mutation are applied with different pre-specified probabilities.

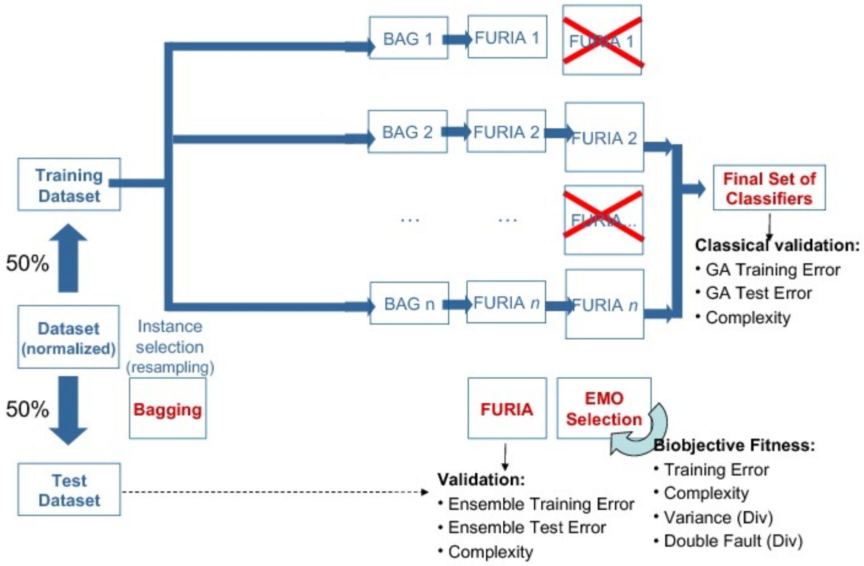


Figure 2: Our framework: after the instance and the feature selection procedures, the component classifiers are derived by the FURIA learning method. Finally, the output is obtained using a voting-based method.

4.2. *The four used evaluation criteria for two-objective NSGA-II*

In this subsection we describe all the considered optimization criteria. We will utilize measures of three different kinds combined into five different two-objective fitness functions:

- **Accuracy**. We use a standard accuracy measure, the training error (TE). TE is computed as follows. Let $h_1(x),\ldots,h_l(x)$ be the outputs of the component classifiers of the selected ensemble E for an input value $x = (x_1,\ldots,x_n)$. For a given sample $\{(x^k,C^k)\}_{k\in\{1\ldots m\}}$, the TE of that MCS is:

$$TE = \frac{1}{m}\cdot\#\{k \mid C^k \neq \underset{j\in\{1\ldots|E|\}}{\arg\max}\ h_j(x^k)\} \quad (2)$$

  with $|E|$ being the number of classifiers in the selected ensemble.

- **Complexity**. The complexity of the ensemble is directly accounted by the number of classifiers in the ensemble:

$$Complx = |E| \quad (3)$$

- **Diversity**. It seems that obtaining a high diversity between classifiers is the goal to be reached, when aiming to achieve performance improvement of MCSs. In the last few years, a group of researchers devoted their attention to the diversity measures [25, 26, 27]. Two measures can be highlighted from the large amount of proposals in this group: the difficulty (θ) and the double fault (δ):

  1. The difficulty measure θ is computed as follows. Let $X = \{i/|E|\}_{i\in\{0,\ldots,|E|\}}$ and $X_k \in X$ be the proportions of classifiers classifying correctly the instance x^k. Then, θ is:

$$\theta = Var(\{X_1,\ldots,X_k,\ldots,X_m\}) \quad (4)$$

  2. The pairwise measure δ for two classifiers h_i and h_j shows the following expression:

$$\delta_{i,j} = \frac{N_{ij}^{00}}{m} \quad (5)$$

with N_{ij}^{00} being the number of examples missclassified by both h_i and h_j. The global value of the measure for the whole selected ensemble is computed as:

$$\delta_{avg} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta_{i,j} \qquad (6)$$

with $L = |E|$ being the number of component classifiers in the ensemble.

From these four criteria we have formed five different two-objective fitness functions. A first option is a standard combination of TE with Complx (from now on called fitness function 2a).

In addition, the use of biobjective functions based on diversity measures is justified by previous findings in the specialized literature. Diversity measures have been deeply studied in [25, 26, 27, 29, 30]. The relationship between diversity measures and accuracy is not clear. In [25], it was shown how the ensemble accuracy and diversity is not as strongly correlated as it could be expected. The authors concluded that accuracy estimation can not be substituted by diversity during the MCS design process. These results were confirmed in [26] in our same framework, classifier selection. In the experimental study developed, the authors drew the conclusion that the use of a single-objective function based on a diversity measure does not outperform the direct use of an error rate. Hence, the combined action of both kinds of measures can lead to a better fuzzy MCS performance in our case. In particular, we combined accuracy measures with the said two diversity measures in [12], obtaining promising results. Hence, in this contribution we will use the combination of TE with θ and δ (fitness functions 2b and 2c respectively) in the current contribution, as in our opinion this may lead to an accuracy improvement, when keeping a low number of classifiers.

Finally, we would like to put more stress on the complexity aspect as proposed in [29, 30], so we join diversity measures with complexity into the two remaining two-objective fitness functions (2d and 2e). By doing so we would like to obtain simple ensembles still having high quality in terms of performance.

Table 1 summarizes the composition of the five biobjective fitness functions proposed.

Table 1. The five fitness function proposed.

abbreviation	1st obj.	2nd obj.
2a	TE	Complx
2b	TE	θ
2c	TE	δ
2d	θ	Complx
2e	δ	Complx

5. Experiments and analysis of results

This section reports all the experiments performed. Firstly, we introduce the experimental setup (Sec. 5.1). Then, in Sec. 5.2 the performance of NSGA-II with the considered five two-objective fitness functions when tackling the classifier selection tasks for FURIA-based fuzzy MCSs is analyzed. Two multiobjective metrics, one unary and one binary [24, 53], are considered to do so. We also show graphs of the obtained Pareto front approximations. Furthermore, we study some representative individual solutions selected from the obtained Pareto sets in Sec. 5.3. Finally, we compare the best single values obtained against the result from the previous stage, that is FURIA-based fuzzy MCSs with bagging not considering classifier selection, as well as against two state-of-the-art algorithms, random forests [38] and bagging C4.5 MCSs [54], in Sec. 5.4.

5.1. Experimental setup

To evaluate the performance of the generated FURIA-based fuzzy MCSs, we have selected 20 datasets with different characteristics concerning the number of examples, features, and classes from the UCI machine learning repository (see Table 2). In order to compare the accuracy of the considered classifiers, we used Dietterichs 5×2-fold cross-validation (5×2-cv), which is considered to be superior to paired k-fold cross validation in classification problems [55].

Table 2. Data sets considered

Data set	#examples	#attr.	#classes
abalone	4178	7	28
breast	700	9	2
glass	214	9	7
heart	270	13	2
ionosphere	352	34	2
magic	19020	10	2
optdigits	5620	64	10
pblocks	5474	10	5
pendigits	10992	16	10
phoneme	5404	5	2
pima	768	8	2
sat	6436	36	6
segment	2310	19	7
sonar	208	60	2
spambase	4602	57	2
texture	5500	40	11
waveform	5000	40	3
wine	178	13	3
vehicle	846	18	4
yeast	1484	8	10

The FURIA-based fuzzy MCSs generated are initially comprised by 50 classifiers. NSGA-II for the component classifier selection works with a population of 50 individuals and runs during 1000 generations. The crossover probability considered is 0.6 and the mutation probability is 0.1. A different run is developed with each of the five fitness function variants for each initial fuzzy MCS, thus resulting in 10 different runs per dataset as a consequence of the 5×2-cv cross validation procedure. All the experiments have been run in a cluster at the University of Granada, Spain, on Intel quadri-core Pentium 2.4 GHz nodes with 2 GBytes of memory, under the Linux operating system.

To compare the Pareto front approximations of the global learning objectives (i.e. MCS test accuracy and complexity) we consider the two of usual kinds of multiobjective metrics [24, 53]. The first group measures the quality of a single nondominated solution set returned by a multiobjective algorithm, while the second group compares the performance of two different multiobjective algorithms. We have selected one metric for each group, hypervolume ratio (HVR) [24] and C-measure [53], respectively.

An useful unary metric to compare Pareto sets is the S metric, proposed by Zitzler [53], and called hypervolume. It measures the volume enclosed by the Pareto front Y'. When there are only two objectives, $S(Y')$ measures the area covered by the Pareto front by adding the areas covered by each individual nondominated point. In the case of a minimization problem, as ours, there is a need to define a reference point. Nevertheless, the relative value of the S metric usually depends upon an arbitrary choice of this point, getting unexpected metric values if it is not correctly fixed [56]. Besides, when the dimension of the objectives is large, it is interesting to normalise them. Alternatively, the hypervolume ratio (HVR) [24] can be considered to avoid these drawbacks. HVR is a very powerful metric, as it both measures diversity and closeness. The HVR can be simply calculated as follows:

$$HVR = \frac{H_1}{H_2} \tag{7}$$

where H_1 and H_2 are the volume (S metric value) of the Pareto front approximation and the true Pareto front, respectively. When HVR equals 1, then the Pareto front approximation and the true one are equal. Thus, HVR values lower than 1 indicate a generated Pareto front that is not as good as the true Pareto front. In our case, as the truen Pareto fronts are not known, we will consider an approximation (a pseudo-optimal Pareto front) obtained by fusing all the (approximate) Pareto fronts generated for each problem instance by any algorithm variant in any run.

As binary metric we will use the coverage, proposed by Zitzler et al. in [53], which compares a pair of non-dominated sets by computing the fraction of each set that is covered by the other:

$$C(X',X'') = \frac{|\{\forall a'' \in X''; \exists a' \in X' : a' \prec a''\}|}{|X''|} \tag{8}$$

where $a' \prec a''$ indicates that the solution a' dominates the solution a'' in a minimization problem and $Y', \overline{Y} \subseteq Y$ are the sets of objective vectors that correspond to X' and $\overline{X}$ (non-dominated decision vectors), respectively.

Hence, the value $C(X',X'') = 1$ means that all the solutions in X'' are dominated by or equal to solutions in X'. The opposite, $C(X',X'') = 0$, represents the situation where none of the solutions in X'' are covered by the set X'. Note that both $C(X',X'')$ and

$C(X'',X')$ have to be considered, since $C(X',X'')$ is not necessarily equal to $1 - C(X'',X')$.

Let us call $\overline{P_i^j}$ the non-dominated solution set returned by NSGA-II usign the variant of fitness function i in the j-th run for a specific problem instance; $P_i = \overline{P_i^1} \bigcup \overline{P_i^2} \bigcup \ldots \bigcup \overline{P_i^{10}}$, the union of the solution sets returned by the ten runs obtained from 5x2cv of algorithm i, and finally $\overline{P_i}$ the set of all non-dominated solutions in the P_i set*(aggregated Pareto fronts). As a complement to the analysis of the results obtained in the two different multiobjective metrics, we will provide graphical representations of some of those aggregated Pareto fronts. When graphically represented, these graphics offers a visual information, not measurable, but sometimes more useful than numeric values.

5.2. Analysis and comparison of the obtained Pareto front approximations

This section is devoted to analyze the performance of the proposed NSGA-II-based approach to classifier selection in FURIA-based fuzzy MCS. First, the quality of the obtained Pareto front approximations considering the five fitness functions defined in Sec. 4.2 will be studied in Sec. 5.2.1. Then, a global comparison among the five methods is made by analyzing their performance on the satisfaction of the two final learning problem goals, the test accuracy and the complexity of the obtained fuzzy MCSs.

On the one hand, regarding the former analysis we draw a table with the minimum, maximum, mean, and standard deviation values obtained for each objective in the Pareto set derived for each of the five fitness functions considered. On the other hand, to compare the Pareto front approximations of the global objectives we consider a table with the same structure as above as well as statistics related to the selected multiobjective metrics, HVR and C. We also plot some of the aggregated Pareto front approximations in order to have a taste of their trends, as drawing all of them would not be feasible.

5.2.1. Analysis of the original Pareto front approximations

As the five two-objective fitness functions considered handle three different types of measures: accuracy, complexity, and diversity, its direct comparison is practically impossible. Instead, in order to give a flavor of the Pareto fronts obtained, we present their characteristic values. We first gather them for all of the folds out of 5x2cv and average them. We show the statistics for each dataset in a different row in Tables 3 to 5. For each fitness function we show the cardinality of the Pareto set (called *Car.*) and for each objective we present the minimum (called *Min.*), maximum (called *Max.*), mean (called *Avg.*), and standard deviation (called *Dev.*) of the averaged 5x2cv values. Let us recall the objectives of the all fitness functions. Function 2a is composed of training error and complexity, while 2b and 2c combine training error and the diversity measures, variance and double fault, respectively. Finally, 2d and 2e assemble complexity with variance and double fault, respectively. Furthermore, we show a visual representation of the aggregated Pareto front approximations for one selected dataset. Figures 3 to 5 represent a visualization of the fronts obtained for the abalone dataset by the five fitness functions.

A first very important conclusion is that, while the first three fitness function variants, 2a, 2b, and 2c, work properly as they allow the multiobjective genetic classifier selection method to derive a significant number of solutions in the Pareto set approximations (cardinal), the other two, 2d and 2e, show a deceptive behavior. On the one hand, function 2d provided a single solution in 10 out of 20 cases and 1.1 solutions in another 5 out of 20 cases. On the other hand, although function 2e allows us to obtain many different solutions in the solution space (Pareto set), all of them correspond to exactly the same objective values (Pareto front). Thus, the latter two fitness functions are not adequate for generating diverse Pareto front approximations.

*Notice that the pseudo-optimal Pareto front is the fusion of the $\overline{P_i}$ sets generated by every variant of the fitness function in all runs developed.

Table 3: Statistics of the Pareto front approximations with the original objectives.

	2a									2b								
		Obj. 1 - TE				Obj. 2 - Complx					Obj. 1 - TE				Obj. 2 - θ			
	Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.	Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.
abalone	14.5	0.505	0.605	0.532	0.029	2.000	17.700	8.821	4.757	58.6	0.509	0.599	0.546	0.025	0.113	0.128	0.119	0.004
breast	7.0	0.000	0.007	0.002	0.003	2.000	3.000	2.701	0.458	6.3	0.000	0.009	0.005	0.004	0.009	0.038	0.025	0.011
glass	8.9	0.004	0.113	0.034	0.036	2.000	6.800	4.395	1.631	11.1	0.007	0.113	0.060	0.036	0.100	0.410	0.258	0.087
heart	8.8	0.001	0.050	0.018	0.020	2.000	4.500	3.372	0.972	9.5	0.001	0.057	0.029	0.019	0.053	0.230	0.143	0.055
ionosphere	6.8	0.003	0.008	0.005	0.002	2.000	2.800	2.504	0.374	5.2	0.003	0.027	0.015	0.010	0.026	0.083	0.054	0.024
magic	4.3	0.097	0.113	0.102	0.008	2.000	6.600	4.250	2.024	11.1	0.098	0.107	0.102	0.003	0.096	0.189	0.152	0.026
optdigits	142.0	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000	196.7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
pblocks	8.2	0.006	0.016	0.009	0.003	2.000	11.200	6.018	3.295	15.8	0.007	0.015	0.010	0.003	0.015	0.055	0.035	0.011
pendigits	17.2	0.000	0.010	0.002	0.002	2.000	9.300	5.839	2.169	22.7	0.000	0.011	0.004	0.003	0.011	0.082	0.040	0.017
phoneme	7.8	0.058	0.086	0.065	0.010	2.000	9.800	5.990	2.864	14.9	0.059	0.083	0.068	0.008	0.076	0.226	0.163	0.039
pima	10.6	0.022	0.084	0.044	0.021	2.000	10.200	5.568	2.875	19.3	0.024	0.100	0.057	0.023	0.090	0.382	0.239	0.074
sat	13.2	0.010	0.049	0.019	0.011	2.000	16.600	8.013	4.646	38.0	0.011	0.049	0.023	0.010	0.047	0.235	0.148	0.044
segment	14.2	0.000	0.012	0.003	0.004	2.000	6.600	4.782	1.534	12.8	0.000	0.015	0.007	0.005	0.014	0.070	0.041	0.017
sonar	7.9	0.000	0.037	0.011	0.016	2.000	3.300	2.853	0.565	7.5	0.000	0.062	0.029	0.023	0.058	0.220	0.148	0.058
spambase	7.8	0.015	0.033	0.020	0.007	2.000	9.400	5.141	2.703	17.3	0.015	0.029	0.020	0.004	0.029	0.122	0.078	0.024
texture	18.3	0.000	0.020	0.003	0.005	2.000	8.000	5.641	1.867	22.3	0.000	0.021	0.006	0.006	0.020	0.127	0.071	0.025
vehicle	17.0	0.002	0.099	0.023	0.026	2.000	13.400	7.669	3.536	24.9	0.003	0.104	0.043	0.029	0.093	0.421	0.277	0.083
waveform	22.8	0.001	0.059	0.011	0.014	2.000	21.000	10.311	5.517	53.3	0.002	0.067	0.021	0.016	0.062	0.411	0.232	0.081
wine	5.3	0.000	0.001	0.000	0.001	2.000	2.100	2.069	0.048	4.6	0.000	0.006	0.004	0.003	0.006	0.020	0.011	0.008
yeast	9.9	0.156	0.250	0.182	0.029	2.000	11.100	5.950	2.956	28.1	0.158	0.254	0.199	0.029	0.189	0.402	0.324	0.051

Table 4: Statistics of the Pareto front approximations with the original objectives.

	2c									2d								
		Obj. 1 - TE				Obj. 2 - δ					Obj. 1 - θ				Obj. 2 - Complx			
	Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.	Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.
abalone	29.3	0.506	0.622	0.538	0.030	0.428	0.460	0.448	0.008	8.3	0.113	0.151	0.122	0.013	2.000	9.400	5.664	2.561
breast	6.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.3	0.018	0.018	0.018	0.000	2.000	2.000	2.000	0.000
glass	8.3	0.005	0.162	0.054	0.058	0.009	0.038	0.025	0.010	1.1	0.158	0.158	0.158	0.000	2.000	2.000	2.000	0.000
heart	6.8	0.000	0.064	0.019	0.024	0.000	0.009	0.005	0.003	1.1	0.093	0.093	0.093	0.000	2.000	2.000	2.000	0.000
ionosphere	57.9	0.004	0.004	0.004	0.000	0.000	0.000	0.000	0.000	1.2	0.045	0.045	0.045	0.000	2.000	2.000	2.000	0.000
magic	8.1	0.097	0.115	0.102	0.007	0.067	0.072	0.070	0.002	1.0	0.124	0.124	0.124	0.000	2.000	2.000	2.000	0.000
optdigits	196.7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	145.7	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
pblocks	10.8	0.006	0.020	0.010	0.005	0.004	0.007	0.006	0.001	1.0	0.022	0.022	0.022	0.000	2.000	2.000	2.000	0.000
pendigits	83.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.0	0.020	0.020	0.020	0.000	2.000	2.000	2.000	0.000
phoneme	10.0	0.058	0.097	0.068	0.013	0.035	0.044	0.040	0.003	1.0	0.114	0.114	0.114	0.000	2.000	2.000	2.000	0.000
pima	11.7	0.021	0.103	0.046	0.026	0.018	0.037	0.029	0.006	1.0	0.145	0.145	0.145	0.000	2.000	2.000	2.000	0.000
sat	2.8	0.010	0.010	0.010	0.000	0.000	1.031	0.323	0.530	1.0	0.079	0.079	0.079	0.000	2.000	2.000	2.000	0.000
segment	69.9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.1	0.024	0.024	0.024	0.000	2.000	2.000	2.000	0.000
sonar	4.5	0.000	0.011	0.003	0.003	0.000	0.001	0.001	0.000	1.1	0.101	0.101	0.101	0.000	2.000	2.000	2.000	0.000
spambase	2.4	0.015	0.015	0.015	0.000	0.000	0.550	0.110	0.246	1.1	0.046	0.046	0.046	0.000	2.000	2.000	2.000	0.000
texture	99.8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.0	0.037	0.037	0.037	0.000	2.000	2.000	2.000	0.000
vehicle	15.2	0.002	0.118	0.030	0.031	0.015	0.034	0.027	0.005	1.0	0.155	0.155	0.155	0.000	2.000	2.000	2.000	0.000
waveform	19.3	0.001	0.001	0.001	0.000	0.000	0.643	0.214	0.371	1.0	0.108	0.108	0.108	0.000	2.000	2.000	2.000	0.000
wine	54.5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.8	0.010	0.010	0.010	0.000	2.000	2.000	2.000	0.000
yeast	15.5	0.156	0.282	0.188	0.034	0.109	0.134	0.123	0.007	1.0	0.246	0.246	0.246	0.000	2.000	2.000	2.000	0.000

Table 5: Statistics of the Pareto front approximations with the original objectives.

	2e								
		Obj. 1 - δ				Obj. 2 - Complx			
	Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.
abalone	120.5	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
breast	86.1	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
glass	102.3	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
heart	94.5	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
ionosphere	89.0	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
magic	136.9	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
optdigits	145.7	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
pblocks	132.5	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
pendigits	136.7	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
phoneme	137.1	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
pima	124.6	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
sat	133.8	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
segment	124.8	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
sonar	99.0	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
spambase	132.1	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
texture	142.2	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
vehicle	126.9	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
waveform	135.9	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
wine	73.3	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000
yeast	131.4	0.000	0.000	0.000	0.000	2.000	2.000	2.000	0.000

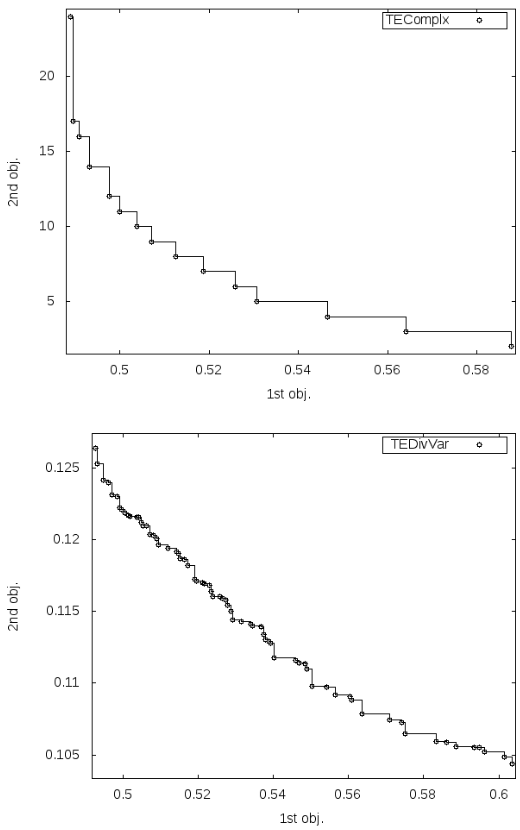


Fig. 3. The Pareto front approximation obtained for *abalone* using fitness function 2a (objective 1 stands for training error and objective 2 for complexity) on the top, and fitness function 2b (Objective 1 stands for training error and objective 2 for variance) on the bottom.

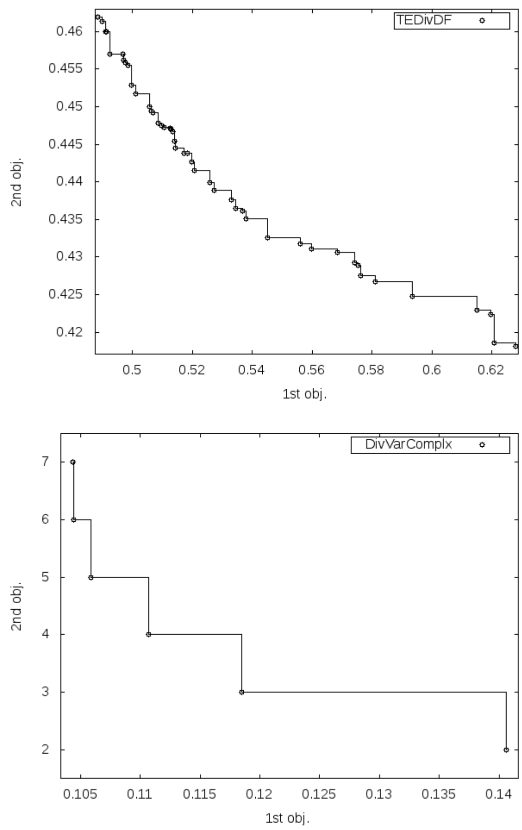


Fig. 4. The Pareto front approximation obtained for *abalone* using fitness function 2c (objective 1 stands for training error and objective 2 for double fault) on the top, and fitness function 2d (Objective 1 stands for variance and objective 2 for complexity) on the bottom.

It can be noticed that the ranges of the objectives vary depending on the dataset given. The training error, which is the first objective of 2a, 2b, and 2c, converges to 0 for several datasets, while growing up to 0.605 overall. The complexity, which is the second objective of 2a, 2d, and 2e lays in the range between 2 and 21, thus showing significant reduction obtained by the multiobjective genetic component classifier selection method (recall that the original number of classifiers is 50). Besides, the variance, being the second objective of 2b and the first of 2d, obtains a minimum value equal to 0 and grows up to 1.202.

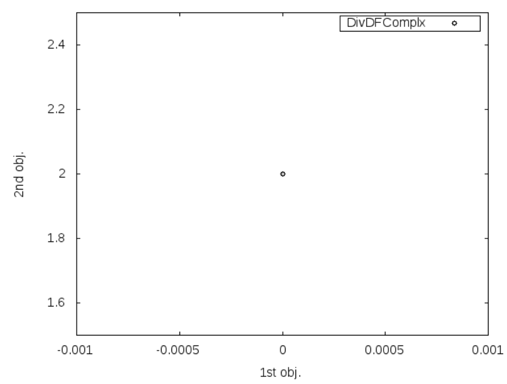


Fig. 5. The Pareto front approximation obtained for *abalone* using fitness function 2e (objective 1 stands for double fault and objective 2 for complexity).

The double fault, which is the second objective of 2c and the first of 2e, lays in the range between 0 and 4.722. In addition, the standard deviation values of the training error vary for each fitness function, 2a, 2b, and 2c variants, depending on the dataset and it is hard to point the one being the most stable. On the other hand, considering the standard deviation of complexity, the lowest values are obtained by both 2d and 2e variants, thus showing the already mentioned deceptive behavior. The same results were obtained for diversity measures.

The combination of double fault and complexity seems to be a special case to be considered. It is a deceptive measure because either the two objectives are not conflicting or there is a specific limit condition where an optimal tradeoff is obtained. For all the datasets, a single optimum of the biobjective function was generated where complexity obtained value 2 and double fault took value 0. Such pair of values could be reasonable, since double fault is a pair-wise metric and in general tends to small ensemble sizes, the same as complexity. Nevertheless, it clearly shows this biobjective fitness function definition is not appropriate for the considered learning task.

A similar case is the 2d variant combining variance with complexity, as for almost all datasets, 19 ouf of 20, it obtained complexity equal to 2. Besides, apart from two datasets, abalone and optdigits, it obtained cardinality equal or close to 1, as already mentioned.

5.2.2. *Perfomance analysis of the five variants in the two global learning objectives*

Since the individual objectives of the Pareto front approximations presented in the previous subsection are not comparable, we have considered two common objectives, namely test error and complexity, in order to compare the results obtained by the five fitness functions proposed. Notice that, these are the actual learning goals that will be considered by the designer in order to choose the final fuzzy MCS structure.

The characteristic values of the Pareto front approximations of the two global learning goals are presented in Tables 6 and 7. The structure of these tables is similar to the ones in the previous subsection. The results are gathered for all of the folds out of 5x2cv and averaged. We show the statistics for each dataset and each fitness function in a different row. For each fitness function we show cardinality (called *Car.*) and for each objective we present minimum (called *Min.*), maximum (called *Max.*), mean (called *Avg.*), and standard deviation (called *Dev.*) of the averaged 5x2cv values.

Table 6. Statistics of the Pareto front approximations with the global learning objectives.

			Obj. 1 - test error				Obj. 2 - Complx			
		Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.
aba	2a	14.5	0.746	0.777	0.758	0.009	2.0	17.7	8.8	4.8
	2b	58.6	0.741	0.763	0.752	0.005	9.3	25.0	17.2	4.2
	2c	29.3	0.745	0.783	0.758	0.009	2.0	16.9	8.5	4.4
	2d	8.3	0.752	0.791	0.767	0.013	2.0	9.4	5.7	2.6
	2e	120.5	0.759	0.803	0.781	0.009	2.0	2.0	2.0	0.0
bre	2a	7.0	0.038	0.060	0.048	0.008	2.0	3.0	2.7	0.5
	2b	6.3	0.037	0.057	0.046	0.007	1.0	3.7	2.5	1.1
	2c	6.0	0.037	0.051	0.045	0.005	3.0	3.4	3.1	0.2
	2d	1.3	0.050	0.051	0.051	0.001	2.0	2.0	2.0	0.0
	2e	86.1	0.037	0.091	0.061	0.012	2.0	2.0	2.0	0.0
gla	2a	8.9	0.286	0.390	0.330	0.033	2.0	6.8	4.4	1.6
	2b	11.1	0.288	0.396	0.331	0.033	1.0	8.9	4.0	2.3
	2c	8.3	0.283	0.403	0.333	0.041	2.1	7.2	4.3	1.9
	2d	1.1	0.360	0.363	0.361	0.002	2.0	2.0	2.0	0.0
	2e	102.3	0.305	0.517	0.400	0.046	2.0	2.0	2.0	0.0
hea	2a	8.8	0.172	0.233	0.202	0.021	2.0	4.5	3.4	1.0
	2b	9.5	0.170	0.231	0.200	0.020	1.0	5.6	3.0	1.4
	2c	6.8	0.178	0.235	0.203	0.021	2.2	4.8	3.3	1.0
	2d	1.1	0.201	0.203	0.202	0.001	2.0	2.0	2.0	0.0
	2e	94.5	0.155	0.299	0.224	0.031	2.0	2.0	2.0	0.0
ion	2a	6.8	0.144	0.187	0.164	0.015	2.0	2.8	2.5	0.4
	2b	5.2	0.145	0.191	0.166	0.020	1.0	3.0	2.0	0.9
	2c	57.9	0.126	0.170	0.148	0.010	13.4	28.1	20.9	3.6
	2d	1.2	0.156	0.160	0.158	0.003	2.0	2.0	2.0	0.0
	2e	89.0	0.129	0.248	0.181	0.026	2.0	2.0	2.0	0.0
mag	2a	4.3	0.132	0.144	0.136	0.005	2.0	6.6	4.2	2.0
	2b	11.1	0.132	0.143	0.136	0.003	1.0	8.2	4.1	2.1
	2c	8.1	0.132	0.145	0.136	0.005	2.0	6.8	4.0	1.8
	2d	1.0	0.146	0.146	0.146	0.000	2.0	2.0	2.0	0.0
	2e	136.9	0.142	0.159	0.150	0.003	2.0	2.0	2.0	0.0
opt	2a	142.0	0.655	0.703	0.678	0.009	2.0	2.0	2.0	0.0
	2b	196.7	0.625	0.641	0.633	0.003	18.0	37.0	25.9	3.3
	2c	196.7	0.625	0.641	0.633	0.003	18.0	37.0	25.9	3.3
	2d	145.7	0.654	0.704	0.678	0.009	2.0	2.0	2.0	0.0
	2e	145.7	0.654	0.704	0.678	0.009	2.0	2.0	2.0	0.0
pbl	2a	8.2	0.028	0.035	0.031	0.003	2.0	11.2	6.0	3.3
	2b	15.8	0.027	0.034	0.030	0.002	1.0	8.8	4.1	2.1
	2c	10.8	0.027	0.038	0.031	0.003	2.0	10.9	5.6	3.0
	2d	1.0	0.034	0.034	0.034	0.000	2.0	2.0	2.0	0.0
	2e	132.5	0.031	0.047	0.038	0.003	2.0	2.0	2.0	0.0
pen	2a	17.2	0.016	0.032	0.020	0.004	2.0	9.3	5.8	2.2
	2b	22.7	0.016	0.034	0.022	0.005	1.0	11.3	4.4	2.5
	2c	83.0	0.014	0.018	0.016	0.001	15.8	26.7	21.5	2.5
	2d	1.0	0.032	0.032	0.032	0.000	2.0	2.0	2.0	0.0
	2e	136.7	0.029	0.042	0.035	0.003	2.0	2.0	2.0	0.0
pho	2a	7.8	0.125	0.152	0.133	0.009	2.0	9.8	6.0	2.9
	2b	14.9	0.127	0.151	0.135	0.007	1.0	8.8	4.3	2.1
	2c	10.0	0.125	0.160	0.136	0.011	2.0	10.0	5.2	2.7
	2d	1.0	0.153	0.153	0.153	0.000	2.0	2.0	2.0	0.0
	2e	137.1	0.144	0.183	0.162	0.008	2.0	2.0	2.0	0.0

In general, the highest cardinality is obtained by the 2e variant, which combines double fault and complexity objectives. However, we have considered it as a deceptive fitness function in the previous subsection. On the other hand, the 2d variant, which is composed of variance and complexity objectives, almost always provides cardinality equal to 1 and was also categorized as deceptive. Thus, both variants are tricky and do not constitute good approach to provide high quality Pareto front approximations.

Table 7. Statistics of the Pareto front approximations with the global learning objectives.(cont.)

			Obj. 1 - test error				Obj. 2 - Complx			
		Car.	Min.	Max.	Avg.	Dev.	Min.	Max.	Avg.	Dev.
pim	2a	10.6	0.235	0.291	0.256	0.018	2.0	10.2	5.6	2.9
	2b	19.3	0.233	0.296	0.261	0.016	1.0	11.6	4.4	2.7
	2c	11.7	0.236	0.290	0.257	0.016	2.0	11.0	5.8	3.0
	2d	1.0	0.277	0.277	0.277	0.000	2.0	2.0	2.0	0.0
	2e	124.6	0.231	0.318	0.274	0.018	2.0	2.0	2.0	0.0
sat	2a	13.2	0.102	0.130	0.110	0.008	2.0	16.6	8.0	4.6
	2b	38.0	0.101	0.132	0.111	0.007	1.0	17.2	6.3	3.8
	2c	2.8	0.102	0.104	0.103	0.001	20.2	22.7	21.3	1.4
	2d	1.0	0.129	0.129	0.129	0.000	2.0	2.0	2.0	0.0
	2e	133.8	0.119	0.140	0.129	0.004	2.0	2.0	2.0	0.0
seg	2a	14.2	0.029	0.048	0.036	0.006	2.0	6.6	4.8	1.5
	2b	12.8	0.029	0.051	0.038	0.006	1.0	6.8	3.4	1.6
	2c	69.9	0.027	0.037	0.032	0.002	14.4	26.1	19.8	2.9
	2d	1.1	0.047	0.047	0.047	0.000	2.0	2.0	2.0	0.0
	2e	124.8	0.037	0.070	0.052	0.006	2.0	2.0	2.0	0.0
son	2a	7.9	0.203	0.284	0.245	0.031	2.0	3.3	2.9	0.6
	2b	7.5	0.217	0.289	0.252	0.026	1.0	3.7	2.5	1.0
	2c	4.5	0.213	0.271	0.242	0.022	2.8	3.3	3.0	0.1
	2d	1.1	0.269	0.274	0.272	0.003	2.0	2.0	2.0	0.0
	2e	99.0	0.188	0.406	0.292	0.047	2.0	2.0	2.0	0.0
spa	2a	7.8	0.057	0.072	0.061	0.005	2.0	9.4	5.1	2.7
	2b	17.3	0.056	0.070	0.061	0.004	1.0	9.8	4.7	2.4
	2c	2.4	0.056	0.058	0.057	0.001	11.9	13.8	12.8	1.1
	2d	1.1	0.072	0.072	0.072	0.000	2.0	2.0	2.0	0.0
	2e	132.1	0.065	0.090	0.077	0.005	2.0	2.0	2.0	0.0
tex	2a	18.3	0.032	0.065	0.040	0.009	2.0	8.0	5.6	1.9
	2b	22.3	0.033	0.067	0.043	0.009	1.0	9.2	4.2	1.9
	2c	99.8	0.028	0.035	0.032	0.001	17.2	30.4	23.8	3.0
	2d	1.0	0.067	0.067	0.067	0.000	2.0	2.0	2.0	0.0
	2e	142.2	0.058	0.087	0.071	0.005	2.0	2.0	2.0	0.0
veh	2a	17.0	0.257	0.302	0.275	0.012	2.0	13.4	7.7	3.5
	2b	24.9	0.255	0.316	0.282	0.015	1.0	13.4	5.4	3.2
	2c	15.2	0.260	0.308	0.278	0.014	2.0	14.1	7.0	3.6
	2d	1.0	0.307	0.307	0.307	0.000	2.0	2.0	2.0	0.0
	2e	126.9	0.270	0.350	0.310	0.017	2.0	2.0	2.0	0.0
wav	2a	22.8	0.148	0.195	0.158	0.011	2.0	21.0	10.3	5.5
	2b	53.3	0.146	0.199	0.163	0.012	1.0	23.1	7.1	4.8
	2c	19.3	0.146	0.152	0.149	0.002	24.0	29.3	26.5	1.8
	2d	1.0	0.197	0.197	0.197	0.000	2.0	2.0	2.0	0.0
	2e	135.9	0.181	0.213	0.196	0.006	2.0	2.0	2.0	0.0
win	2a	5.3	0.051	0.100	0.076	0.019	2.0	2.1	2.1	0.0
	2b	4.6	0.054	0.091	0.074	0.014	1.0	2.3	1.6	0.7
	2c	54.5	0.018	0.063	0.038	0.013	15.2	33.8	24.4	4.6
	2d	1.8	0.065	0.076	0.071	0.006	2.0	2.0	2.0	0.0
	2e	73.3	0.037	0.231	0.125	0.046	2.0	2.0	2.0	0.0
yea	2a	9.9	0.404	0.453	0.423	0.015	2.0	11.1	5.9	3.0
	2b	28.1	0.396	0.467	0.424	0.016	1.0	10.9	4.8	2.6
	2c	15.5	0.405	0.472	0.426	0.019	2.0	12.5	6.3	3.0
	2d	1.0	0.445	0.445	0.445	0.000	2.0	2.0	2.0	0.0
	2e	131.4	0.421	0.495	0.458	0.014	2.0	2.0	2.0	0.0

The other variants, 2a, 2b, and 2c, combining training error with complexity, variance, and double fault, respectively, seem to be performing well.

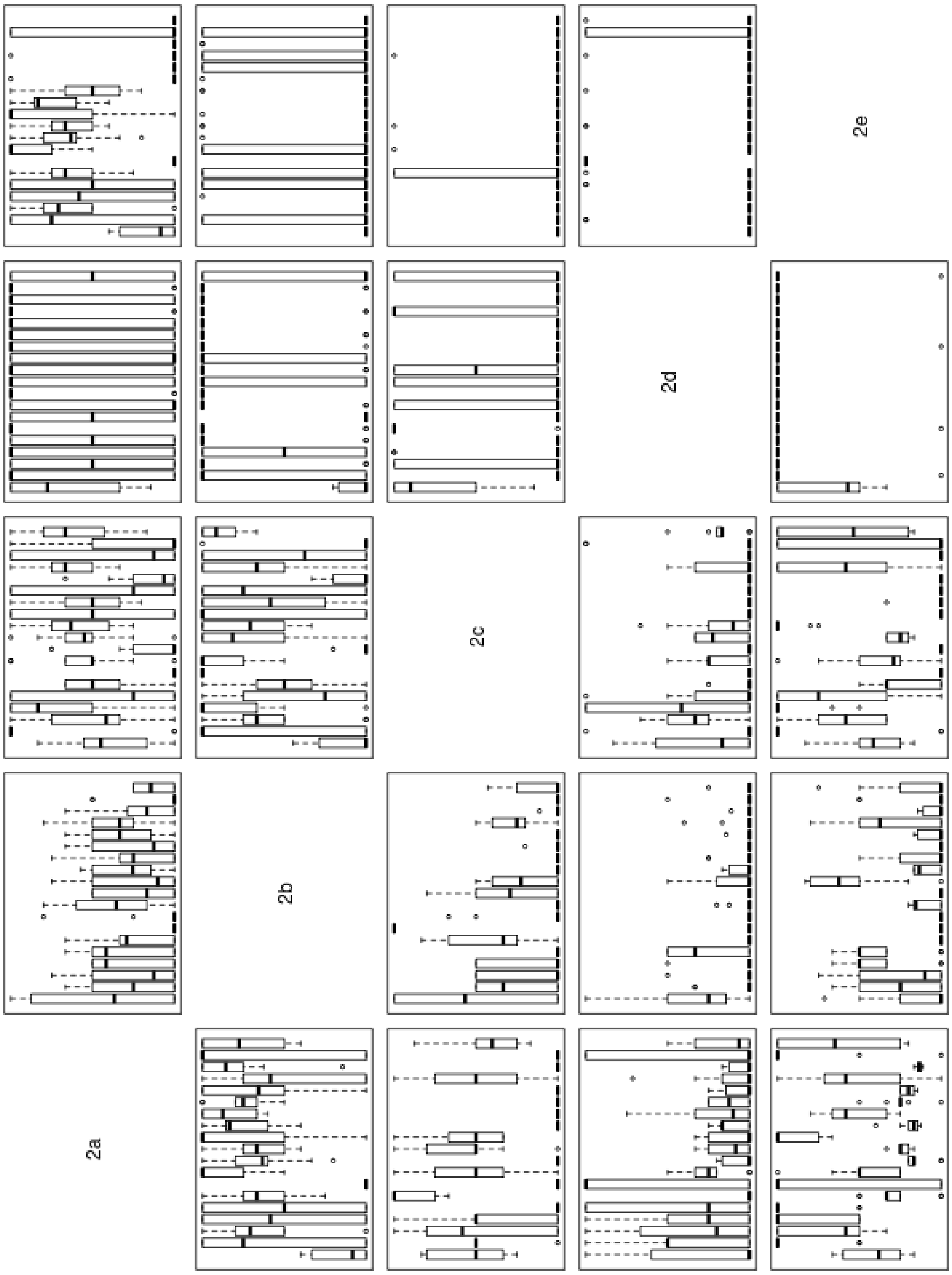
Having an insight to the results, we may emphasize the 2c variant, as the one having the lowest *Avg.* test error value for 10 out of 20 cases (+2 ties). Notice that the 2a variant is just slightly worse than the other two approaches. Considering the complexity, we may see that the 2e variant obtained the lowest value in all of the cases, while the 2d variant was not much worse with 19 out of 20 best values (19 ties with the 2e variant). The 2a and 2b variants obtained only one best result (both ties with the 2d and 2e variants), placing the 2c variant in the last position with no best value obtained.

Table 8. Comparison of Pareto fronts using HVR measure.

	2a	2b	2c	2d	2e
aba	**0.9973**	0.5126	**0.9973**	0.9961	0.9962
bre	0.6632	**0.9955**	0.3321	0.6627	0.6644
gla	0.8455	**0.9867**	0.8314	0.8376	0.8469
hea	0.6582	**0.9858**	0.5915	0.6564	0.6625
ion	0.9437	**0.9796**	0.5294	0.9416	0.9464
mag	0.9323	**0.9988**	0.9324	0.9300	0.9307
opt	**0.9952**	0.3335	0.3335	**0.9952**	**0.9952**
pbl	0.8555	**0.9983**	0.8555	0.8547	0.8553
pen	0.9609	**0.9992**	0.4307	0.9580	0.9587
pho	0.9267	**0.9978**	0.9266	0.9224	0.9241
pim	0.8700	**0.9944**	0.8700	0.8650	0.8730
sat	0.9554	**0.9988**	0.1738	0.9510	0.9528
seg	0.9483	**0.9982**	0.3295	0.9452	0.9472
son	0.6544	**0.9797**	0.3927	0.6492	0.6597
spa	0.9071	**0.9978**	0.1542	0.9047	0.9060
tex	0.9587	**0.9983**	0.3518	0.9525	0.9542
veh	0.8523	**0.9940**	0.8520	0.8459	0.8521
wav	0.9638	**0.9984**	0.2068	0.9554	0.9585
win	0.9240	**0.9893**	0.1066	0.9213	0.9265
yea	0.9315	**0.9947**	0.9311	0.9256	0.9301
avg.	0.8450	**0.8920**	0.5299	0.8415	0.8870
dev.	0.2202	0.2682	0.3263	0.2194	0.1058

Table 8 presents the results using the HVR metric. Besides, we have used box-plots based on the C metric that calculates the dominance degree of the Pareto front approximations of every pair of algorithms (see Fig. 6). Each rectangle contains ten box-plots representing the distribution of the C values for a certain ordered pair of fitness functions in the 20 problem instances (from abalone to yeast in alphabetical order). Each box refers to algorithm A in the corresponding row and algorithm B in the corresponding column and gives the fraction of B covered by A $(C(A,B))$. Consider for instance the top right boxplots, which represent the fraction of solutions of the 2e variant, considering the joint optimization of double fault and complexity, covered by the non-dominated sets produced by the 2a vari-

Figure 6: Comparison of the Pareto fronts using C-measure by means of box-plots.



ant, composed of the training error and complexity measures. In each box, the minimum and maximum values are the lowest and highest lines, the upper and lower ends of the box are the upper and lower quartiles, and a thick line within the box shows the median.

Our analysis of the HVR measure clearly points out the best performing fitness function for the final learning goal. The 2b variant, considering the joint optimization of training error and variance, obtained the highest value in 18 out of 20 cases. Nevertheless, it provided some instability for two remaining datasets and thus a high standard deviation value. For abalone and optdigits it obtained the worst values among the five fitness function designs. Function 2a, optimizing training error and complexity, obtained two ties. Even being a deceptive fitness function, the 2d, which joins variance and complexity as objectives, and 2e variant, which combines double fault and complexity, obtained one tie as well as fitness function 2c, jointly optimizing training error and double fault.

Concerning the average value on the twenty datasets considered, the order is 2b, 2e, 2a, 2d, 2c. It is a surprising fact that the two deceptive functions are not located in the two last positions but are able to overcome almost all variants for 2e, and one other variant for 2d. It seems that, although they are not able to derive a diverse set of solutions, the selected fuzzy MCSs obtained show a good performance in the global learning objectives tradeoff curve. Anyway, from all the latter analysis we may conclude that the 2b fitness function seems to be the best performing variant.

The analysis of the C-measure (see Fig. 6) highlights the 2a and the 2b variants, which clearly outperform the other fitness functions, especially the latter. When comparing these two approaches between them, the 2b fitness function obtains better results. Notice that, comparisons with either 2d or 2e variants provide deceptive results due to the small number of solutions contained in their Pareto front approximations.

Hence, considering the information provided by the two multiobjective metrics we may clearly draw the conclusion that the 2b fitness function is the best performing approach.

Finally, in order to complement the latter analysis, the aggregated Pareto fronts will be represented graphically for three of the datasets: abalone, waveform, and magic (see Figs. 7, 8, and 9, respectively) in order to allow an easy visual comparison of the performance of the different variants of the fitness functions.

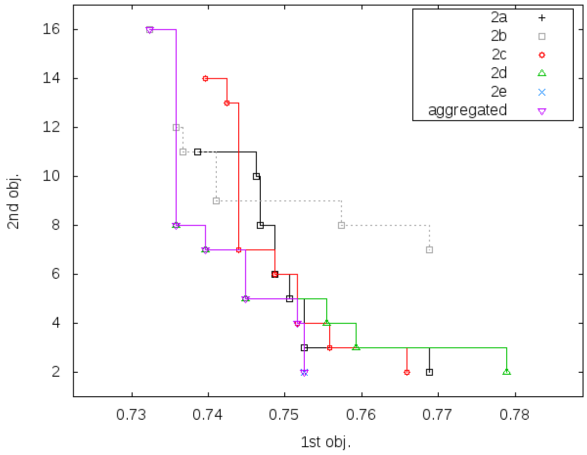


Fig. 7. The Pareto front approximations obtained for *abalone* using the five fitness functions. Objective 1 stands for test error and objective 2 for complexity. The pseudo-optimal Pareto front is also drawn for reference.

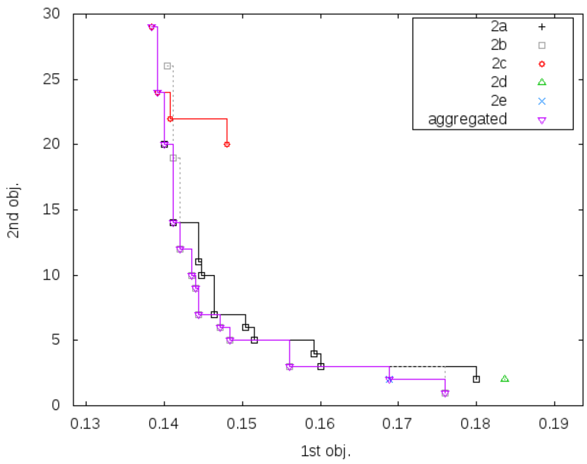


Fig. 8. The Pareto front approximations obtained for *waveform* using the five fitness functions. Objective 1 stands for test error and objective 2 for complexity. The pseudo-optimal Pareto front is also drawn for reference.

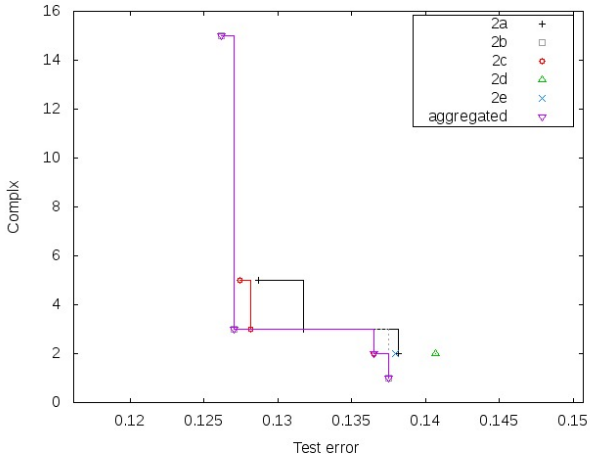


Fig. 9. The Pareto front approximations obtained for *magic* using the five fitness functions. Objective 1 stands for test error and objective 2 for complexity. The pseudo-optimal Pareto front is also drawn for reference.

It can be seen how the graphs corroborate the conclusions of the metrics analysis: the 2b fitness function obtained the worst HVR metric value for the abalone dataset, while it obtained the best performance, which is close to the pseudo-optimal Pareto front, when looking at the waveform and magic datasets. Notice how, the 2e variant obtains a single nondominated solution in the three problems but in all of them this solution is included in the pseudo-optimal Pareto front, thus justifying the good values obtained in the HVR and C metrics. That is not the case for function 2d whose small number of nondominated solutions are usually far from the pseudo-optimal front. The bad performance of the 2c variant is also clearly observed.

5.3. Analysis and comparison of single solutions selected from the obtained Pareto front approximations

In this section, we aim to analyze the final performance of our proposal by imitating the process a human designer will develop in order to select a desired FURIA-based fuzzy MCS structure from those available in the obtained accuracy-complexity nondominated fronts.

From each Pareto front approximation we have selected four different solutions, the one having the best value in the first objective in the considered fitness function, the one with the best value in the second objective in the considered fitness function, the one with the best tradeoff value, and the one with the best test error value. The tradeoff solution is selected as follows: we compute 1000 random weights $w1 \in [0,1]$, take the average value of the aggregation function of both objectives $Obj1$ and $Obj2$: $(w1 * Obj1 + (1 - w1) * Obj2)$, and select the solution with the highest aggregated value. For each solution we present the values of three global learning objectives, Training error (*Tra*), test error (*Tst*), and complexity (*Cmpl*) in Tables 9 and 10. The average and standard deviation value for each of four different solutions in the 20 problems is also presented in Table 11. We do not show the two diversity measures values, since that was not the final learning objective. Note that we used diversity measures combined with accuracy and complexity measures in order to improve the accuracy-complexity tradeoff in the obtained FURIA-based fuzzy MCS.

From the results obtained we may draw following conclusions:

- The best performance in terms of test accuracy was obtained by the 2c variant. It outperforms the other approaches in 6 out of 20 cases (+7 ties) and also obtains the best average value. The 2b approach was one step behind obtaining 5 best results (+6 ties) and the second best average value. Of course, the 2d and 2e variants obtained the worst results, since they directly do not include accuracy in the objective space.
- Considering the complexity criterion the best results were obtained by 2a, 2d, and 2e variants. For all datasets they obtained the lowest number of classifiers equal to 2 and the lowest average value. Note that we discard 2b as the best value, which obtained number of classifiers equal to 1 15 times, since it is not considered as a MCS, but as a single classifier.
- It is rather hard to point a single approach finding the best accuracy-complexity tradeoff. The 2d and 2e approaches should be rather discarded, since for all of the solutions out of all datasets they provided the same complexity (equal to 2). Although the 2c fitness function provided the best averaged

Table 9: Statistics of four single solutions selected from the Pareto fronts.

		Best of 1st obj.			Best of 2nd obj.			Best tradeoff			Best test		
		Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl
aba	2a	0.505	0.751	17.700	0.605	0.776	2.000	0.605	0.776	2.000	0.512	0.746	14.400
	2b	0.509	0.750	22.300	0.599	0.756	9.400	0.509	0.750	22.000	0.536	0.741	18.600
	2c	0.506	0.750	16.800	0.622	0.780	2.000	0.506	0.750	20.000	0.511	0.745	13.900
	2d	0.599	0.756	9.400	0.689	0.791	2.000	0.689	0.791	2.000	0.606	0.752	8.000
	2e	0.611	0.773	2.000	0.611	0.773	2.000	0.611	0.773	2.000	0.649	0.759	2.000
bre	2a	0.000	0.039	3.000	0.007	0.054	2.000	0.007	0.054	2.000	0.001	0.038	2.900
	2b	0.000	0.045	3.600	0.009	0.045	1.000	0.000	0.045	6.000	0.005	0.037	2.700
	2c	0.000	0.038	3.000	0.000	0.038	3.000	0.000	0.038	3.000	0.000	0.037	3.000
	2d	0.014	0.051	2.000	0.014	0.051	2.000	0.014	0.051	2.000	0.014	0.050	2.000
	2e	0.010	0.052	2.000	0.010	0.052	2.000	0.010	0.052	2.000	0.023	0.037	2.000
gla	2a	0.004	0.309	6.800	0.113	0.364	2.000	0.113	0.364	2.000	0.015	0.286	5.600
	2b	0.007	0.325	8.700	0.113	0.359	1.000	0.063	0.348	15.000	0.052	0.288	4.700
	2c	0.005	0.300	7.200	0.162	0.397	2.100	0.066	0.323	7.000	0.021	0.283	5.500
	2d	0.142	0.360	2.000	0.142	0.360	2.000	0.142	0.360	2.000	0.142	0.360	2.000
	2e	0.133	0.372	2.000	0.133	0.372	2.000	0.133	0.372	2.000	0.193	0.305	2.000
hea	2a	0.001	0.184	4.500	0.050	0.199	2.000	0.050	0.199	2.000	0.013	0.172	3.700
	2b	0.001	0.197	5.500	0.057	0.215	1.000	0.001	0.198	3.000	0.030	0.170	2.900
	2c	0.000	0.185	4.800	0.064	0.214	2.200	0.000	0.185	5.000	0.010	0.178	3.900
	2d	0.074	0.201	2.000	0.074	0.201	2.000	0.074	0.201	2.000	0.074	0.201	2.000
	2e	0.059	0.204	2.000	0.059	0.204	2.000	0.059	0.204	2.000	0.100	0.155	2.000
ion	2a	0.003	0.153	2.800	0.008	0.149	2.000	0.008	0.149	2.000	0.006	0.144	2.500
	2b	0.003	0.162	3.000	0.027	0.169	1.000	0.003	0.162	3.000	0.013	0.145	2.300
	2c	0.004	0.126	18.700	0.004	0.126	18.700	0.004	0.126	16.000	0.004	0.126	18.700
	2d	0.024	0.156	2.000	0.024	0.156	2.000	0.024	0.156	2.000	0.024	0.156	2.000
	2e	0.014	0.162	2.000	0.014	0.162	2.000	0.014	0.162	2.000	0.034	0.129	2.000
mag	2a	0.097	0.133	6.600	0.113	0.144	2.000	0.113	0.144	2.000	0.097	0.132	5.600
	2b	0.098	0.132	8.200	0.107	0.143	1.000	0.100	0.135	3.000	0.098	0.132	7.400
	2c	0.097	0.133	6.600	0.115	0.144	2.000	0.108	0.140	2.000	0.098	0.132	5.600
	2d	0.118	0.146	2.000	0.118	0.146	2.000	0.118	0.146	2.000	0.118	0.146	2.000
	2e	0.116	0.145	2.000	0.116	0.145	2.000	0.116	0.145	2.000	0.119	0.142	2.000
opt	2a	0.401	0.686	2.000	0.401	0.686	2.000	0.401	0.686	2.000	0.452	0.655	2.000
	2b	0.175	0.632	30.200	0.175	0.632	30.200	0.175	0.632	37.000	0.198	0.625	26.000
	2c	0.175	0.632	30.200	0.175	0.632	30.200	0.175	0.632	37.000	0.198	0.625	26.000
	2d	0.400	0.685	2.000	0.400	0.685	2.000	0.400	0.685	2.000	0.451	0.654	2.000
	2e	0.400	0.685	2.000	0.400	0.685	2.000	0.400	0.685	2.000	0.451	0.654	2.000
pbl	2a	0.006	0.029	11.200	0.016	0.035	2.000	0.016	0.035	2.000	0.007	0.028	6.800
	2b	0.007	0.028	8.600	0.015	0.032	1.000	0.010	0.030	3.000	0.009	0.027	4.800
	2c	0.006	0.028	10.900	0.020	0.037	2.000	0.013	0.034	3.000	0.007	0.027	7.600
	2d	0.017	0.034	2.000	0.017	0.034	2.000	0.017	0.034	2.000	0.017	0.034	2.000
	2e	0.016	0.033	2.000	0.016	0.033	2.000	0.016	0.033	2.000	0.020	0.031	2.000
pen	2a	0.000	0.017	9.300	0.010	0.031	2.000	0.010	0.031	2.000	0.000	0.016	8.200
	2b	0.000	0.017	11.300	0.011	0.033	1.000	0.005	0.024	2.000	0.000	0.016	8.700
	2c	0.000	0.014	21.800	0.000	0.014	21.800	0.000	0.014	24.000	0.000	0.014	21.800
	2d	0.011	0.032	2.000	0.011	0.032	2.000	0.011	0.032	2.000	0.011	0.032	2.000
	2e	0.010	0.032	2.000	0.010	0.032	2.000	0.010	0.032	2.000	0.013	0.029	2.000
pho	2a	0.058	0.126	9.800	0.086	0.151	2.000	0.086	0.151	2.000	0.059	0.125	9.000
	2b	0.059	0.127	8.800	0.083	0.150	1.000	0.059	0.127	9.000	0.061	0.127	7.600
	2c	0.058	0.126	10.000	0.097	0.160	2.000	0.080	0.145	9.000	0.059	0.125	9.400
	2d	0.089	0.153	2.000	0.089	0.153	2.000	0.089	0.153	2.000	0.089	0.153	2.000
	2e	0.090	0.152	2.000	0.090	0.152	2.000	0.090	0.152	2.000	0.097	0.144	2.000

test accuracy, it should also be skipped, as this good performance is obtained at the cost of the highest complexity, with a very significant difference. Thus, the two left fitness functions are 2a and 2b. The first one provides rather lower complexity on average, whereas the latter one obtains better averaged accuracy with a slightly higher complexity.

To conclude, let us try to have an insight into the influence of the relation between the two objectives and the final success in the learning problem. Combination of diversity measures with complexity tends to produce small ensemble sizes. The two fitness functions obtained ensembles composed of only two classifiers. Consequently, these ensembles do not have a high quality, they obtained rather low accuracy. Combination of the training error with complexity seems to be a good way to look for the good balance between performance and the number of classifiers in the ensemble. However, it seems to overfit quite often. Our last proposal was a combination of training error with two diversity measures. Such two fitness functions, 2b and 2c, lead to finally selected high quality ensembles with a good accuracy-complexity tradeoff, as the ensembles are kept quite small in most of the cases. Thus, we may conclude that the combination of training error and diversity measures for genetic classifier selection in FURIA-based fuzzy classifiers leads to the obtaining

Table 10: Statistics of four single solutions selected from the Pareto fronts.(cont.)

		Best of 1st obj.			Best of 2nd obj.			Best tradeoff			Best test		
		Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl
pim	2a	0.027	0.245	10.200	0.087	0.283	2.000	0.087	0.283	2.000	0.033	0.235	6.900
	2b	0.024	0.245	11.600	0.100	0.285	1.000	0.070	0.267	2.000	0.041	0.233	6.800
	2c	0.021	0.247	11.000	0.103	0.287	2.000	0.054	0.266	12.000	0.033	0.236	7.000
	2d	0.103	0.277	2.000	0.103	0.277	2.000	0.103	0.277	2.000	0.103	0.277	2.000
	2e	0.090	0.279	2.000	0.090	0.279	2.000	0.090	0.279	2.000	0.128	0.231	2.000
sat	2a	0.010	0.103	16.600	0.049	0.130	2.000	0.049	0.130	2.000	0.013	0.102	11.600
	2b	0.011	0.103	17.200	0.049	0.132	1.000	0.011	0.103	15.000	0.012	0.101	14.600
	2c	0.010	0.102	21.400	0.010	0.102	21.600	0.010	0.102	17.000	0.010	0.102	21.200
	2d	0.053	0.129	2.000	0.053	0.129	2.000	0.053	0.129	2.000	0.053	0.129	2.000
	2e	0.050	0.130	2.000	0.050	0.130	2.000	0.050	0.130	2.000	0.060	0.119	2.000
seg	2a	0.000	0.031	6.600	0.012	0.047	2.000	0.012	0.047	2.000	0.001	0.029	5.500
	2b	0.000	0.033	6.700	0.015	0.043	1.000	0.004	0.036	6.000	0.002	0.029	5.100
	2c	0.000	0.027	17.600	0.000	0.027	17.600	0.000	0.027	20.000	0.000	0.027	17.600
	2d	0.017	0.047	2.000	0.017	0.047	2.000	0.017	0.047	2.000	0.017	0.047	2.000
	2e	0.013	0.048	2.000	0.013	0.048	2.000	0.013	0.048	2.000	0.019	0.037	2.000
son	2a	0.000	0.212	3.300	0.037	0.253	2.000	0.037	0.253	2.000	0.005	0.203	3.000
	2b	0.000	0.228	3.700	0.062	0.252	1.000	0.000	0.228	3.000	0.015	0.217	3.200
	2c	0.000	0.222	3.300	0.011	0.232	2.800	0.000	0.222	3.000	0.002	0.213	3.000
	2d	0.064	0.274	2.000	0.064	0.274	2.000	0.064	0.274	2.000	0.065	0.269	2.000
	2e	0.042	0.262	2.000	0.042	0.262	2.000	0.042	0.262	2.000	0.081	0.188	2.000
spa	2a	0.015	0.058	9.400	0.033	0.071	2.000	0.033	0.071	2.000	0.016	0.057	7.600
	2b	0.015	0.057	9.800	0.029	0.070	1.000	0.015	0.057	11.000	0.018	0.056	6.800
	2c	0.015	0.056	12.800	0.015	0.056	12.200	0.015	0.056	11.000	0.015	0.056	12.800
	2d	0.034	0.072	2.000	0.034	0.072	2.000	0.034	0.072	2.000	0.034	0.072	2.000
	2e	0.034	0.072	2.000	0.034	0.072	2.000	0.034	0.072	2.000	0.040	0.065	2.000
tex	2a	0.000	0.033	8.000	0.020	0.064	2.000	0.020	0.064	2.000	0.000	0.032	7.300
	2b	0.000	0.034	9.200	0.021	0.062	1.000	0.001	0.035	14.000	0.001	0.033	7.800
	2c	0.000	0.028	23.200	0.000	0.028	23.200	0.000	0.028	21.000	0.000	0.028	23.200
	2d	0.024	0.067	2.000	0.024	0.067	2.000	0.024	0.067	2.000	0.024	0.067	2.000
	2e	0.021	0.062	2.000	0.021	0.062	2.000	0.021	0.062	2.000	0.025	0.058	2.000
veh	2a	0.002	0.267	13.400	0.099	0.290	2.000	0.099	0.290	2.000	0.011	0.257	9.500
	2b	0.003	0.272	13.400	0.104	0.303	1.000	0.058	0.289	2.000	0.023	0.255	7.500
	2c	0.002	0.271	14.000	0.118	0.302	2.000	0.058	0.287	7.000	0.015	0.260	8.600
	2d	0.112	0.307	2.000	0.112	0.307	2.000	0.112	0.307	2.000	0.112	0.307	2.000
	2e	0.107	0.300	2.000	0.107	0.300	2.000	0.107	0.300	2.000	0.139	0.270	2.000
wav	2a	0.001	0.150	21.000	0.059	0.194	2.000	0.059	0.194	2.000	0.003	0.148	17.400
	2b	0.002	0.149	23.100	0.067	0.192	1.000	0.010	0.159	7.000	0.003	0.146	18.700
	2c	0.001	0.146	26.200	0.001	0.146	26.400	0.001	0.146	29.000	0.001	0.146	26.400
	2d	0.064	0.197	2.000	0.064	0.197	2.000	0.064	0.197	2.000	0.064	0.197	2.000
	2e	0.061	0.194	2.000	0.061	0.194	2.000	0.061	0.194	2.000	0.072	0.181	2.000
win	2a	0.000	0.052	2.100	0.001	0.054	2.000	0.001	0.054	2.000	0.000	0.051	2.100
	2b	0.000	0.072	1.700	0.006	0.057	1.000	0.000	0.072	2.000	0.004	0.054	1.400
	2c	0.000	0.021	17.400	0.000	0.021	17.400	0.000	0.021	17.000	0.000	0.018	18.700
	2d	0.007	0.066	2.000	0.007	0.066	2.000	0.007	0.066	2.000	0.007	0.065	2.000
	2e	0.007	0.058	2.000	0.007	0.058	2.000	0.007	0.058	2.000	0.022	0.037	2.000
yea	2a	0.156	0.406	11.100	0.250	0.452	2.000	0.250	0.452	2.000	0.158	0.404	10.100
	2b	0.158	0.410	10.900	0.254	0.464	1.000	0.158	0.410	11.000	0.188	0.396	7.100
	2c	0.156	0.412	12.500	0.282	0.467	2.000	0.156	0.412	13.000	0.159	0.405	10.800
	2d	0.281	0.445	2.000	0.281	0.445	2.000	0.281	0.445	2.000	0.281	0.445	2.000
	2e	0.260	0.453	2.000	0.260	0.453	2.000	0.260	0.453	2.000	0.291	0.421	2.000

Table 11: A comparison of the averaged performance of the four single solutions selected from the obtained Pareto sets.

		Best of 1st obj.			Best of 2nd obj.			Best tradeoff			Best test		
		Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl	Tra	Tst	Cmpl
2a	avg.	0.064	0.199	8.770	0.103	0.221	2.000	0.103	0.221	2.000	0.047	0.193	7.085
	dev.	0.140	0.207	5.373	0.152	0.210	0.000	0.152	0.210	0.000	0.116	0.202	4.135
2b	avg.	0.054	0.201	10.875	0.095	0.220	2.880	0.063	0.205	8.800	0.056	0.191	8.235
	dev.	0.120	0.200	7.337	0.134	0.202	6.698	0.118	0.201	8.752	0.122	0.197	6.411
2c	avg.	0.053	0.193	14.470	0.090	0.210	10.660	0.062	0.198	13.800	0.047	0.189	13.235
	dev.	0.119	0.203	7.681	0.148	0.214	10.212	0.118	0.204	9.518	0.116	0.200	7.878
2d	avg.	0.112	0.223	2.370	0.117	0.225	2.000	0.117	0.225	2.000	0.083	0.221	2.300
	dev.	0.150	0.206	1.655	0.166	0.211	0.000	0.166	0.211	0.000	0.064	0.202	1.342
2e	avg.	0.107	0.223	2.000	0.107	0.223	2.000	0.107	0.223	2.000	0.000	0.200	2.000
	dev.	0.153	0.210	0.000	0.153	0.210	0.000	0.153	0.210	0.000	0.000	0.203	0.000

good results and is a promising approach. Nevertheless, this is a quite subjective and user-dependent decision.

5.4. Comparison between EMO-selected and non-selected FURIA-based fuzzy MCSs

This subsection presents a final benchmarking of the performance of NSGA-II combined with FURIA-

Table 12: A comparison of the NSGA-II FURIA-based fuzzy MCSs against static FURIA-based MCS.

NSGA-II combined with FURIA-based MCSs.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	**0.741**	**0.037**	0.283	**0.170**	**0.126**	**0.132**	**0.625**	**0.027**	**0.014**	**0.125**	**0.231**	**0.101**	**0.027**	**0.188**	**0.056**	**0.028**	**0.255**	**0.146**	**0.018**	**0.396**
fitness func.	2b	2b	2c	2b	2c	2a	2b	2c	2c	2c	2e	2b	2c	2e	2b	2c	2b	2c	2c	2b
nr of cl.	18.6	2.7	5.5	2	18.7	5.6	26	4.8	21.8	9	2	14.6	17.6	2	6.8	23.2	7.5	18.7	18.7	7.1
FURIA-based MCSs algorithms Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.753	**0.037**	0.313	0.178	0.134	0.136	0.628	0.028	0.015	0.136	0.235	0.105	0.035	0.198	0.061	0.036	0.276	0.156	0.036	0.408
nr of cl.	10	10	7	7	7	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10
FURIA-based MCSs algorithms. Ensemble size 50.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.748	0.041	0.287	0.182	0.145	0.135	0.630	0.028	0.016	0.135	0.241	0.102	0.034	0.226	0.059	0.031	0.275	0.149	0.035	0.400
C4.5 ensembles with bagging. Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.772	0.043	0.306	0.194	0.149	0.134	0.697	0.03	0.028	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	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Random forests. Small ensemble sizes.																				
	aba	bre	gla	hea	ion	mag	opt	pbl	pen	pho	pim	sat	seg	son	spa	tex	veh	wav	win	yea
test err.	0.777	0.041	**0.282**	0.211	0.14	0.134	0.695	0.031	0.016	0.119	0.264	0.104	0.034	0.239	0.06	0.04	0.269	0.185	0.048	0.438
nr of cl.	7	7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

based fuzzy MCSs. The aim of this work is to propose an advanced OCS framework, embedding NSGA-II with five different two-objective fitness function designs into FURIA-based fuzzy MCS. By doing so, we would like to obtain a good accuracy-complexity tradeoff. We present a comparison between the best results obtained from the genetically selected FURIA-based fuzzy MCSs, against these ensembles obtained with a fixed ensemble size. FURIA-based fuzzy MCSs are comprised by 7 or 10 classifiers, the small ensemble sizes providing the best results in our previous contribution [17], and with 50 classifiers, the original structure of the EMO-selected fuzzy MCSs in the previous sections. We also compare it with two state-of-the-art algorithms, random forests [38] and bagging C4.5 MCSs [54], comprised by 7 or 10 classifiers [17].

Table 12 presents test errors for all the datasets. It may be clearly seen that our new approach obtained the best performance overall. It outperformed the others in 18 out of 20 cases (+1 tie). Considering complexity, EMO-selected fuzzy MCSs keep a reasonably low number of classifiers, obtaining value 2 (3 times) in the best case and value 26 (for the optdigits dataset) in the worst case. Comparing to the original small ensemble sizes it is enough to increase the amount of classifier up to 2.5 times in the worst case in order to obtain good performance. Notice that in 11 out of 20 cases the EMO-selected fuzzy MCSs obtained the lowest complexity of the five MCS design variants considered. Thus, we may draw the conclusion that NSGA-II combined with FURIA-based fuzzy MCSs is a good approach to obtain high quality, well performing ensembles with a good accuracy-complexity tradeoff, when dealing with high dimensional datasets.

6. Conclusions and future works

In this study, we proposed a methodology in which a bagging approach together with a feature selection technique is used to train Fuzzy Unordered Rules Induction Algorithm (FURIA) in order to obtain a Fuzzy Multiclassifier System. We used a single winner-based method on top of the base classifiers. We proved that a single FURIA classifier performs well and is able to provide instability capabilities to some extent. Then, we tested FURIA Multiclassifier Systems with bagging, feature selection, and combination of both of them. By using the said techniques, we aimed to obtain FURIA-based fuzzy MCSs properly dealing with high dimensional data.

We have conducted comprehensive experiments over 21 datasets taken from the UCI machine learning repository. It turned out that the combination of bagging with FURIA turned out to be the best from among proposed ones. This approach provided promising results in comparison to the two state-of-the-art algorithms.

One of the next steps we will consider in the future line is to employ a search algorithm such like single and multiobjective metaheuristics, greedy algorithms, local search, etc. to look for an optimal size of the ensemble. The other way to follow is to try to combine classifiers in a dynamic manner, in a way that a classifier or a set of them is responsible just for a particular data region. Furthermore, we would like to study the influence of other parameters (FURIA parameters, MCS parameters, etc.).

References

1. L. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
2. S.J. Verzi, G.L. Heileman, and M. Georgiopoulos. Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory. *Neural Networks*, 19(4):446–468, 2006.
3. W. Pedrycz and K.C. Kwak. Boosting of granular models. *Fuzzy Sets and Systems*, 157(22):2934–2953, 2006.
4. H. Takahashi and H. Honda. A new reliable cancer diagnosis method using boosted fuzzy classifier with a SWEEP operator method. *Journal of Chemical Engineering of Japan*, 38(9):763–773, 2005.
5. J. Canul-Reich, L. Shoemaker, and L.O. Hall. Ensembles of fuzzy classifiers. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, London (UK), 2007.
6. Y. Nojima and H. Ishibuchi. Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design. *International Journal of Hybrid Intelligent Systems*, 4(3):157–169, 2007.
7. C. Marsala. Data mining with ensembles of fuzzy decision trees. In *IEEE Symposium on Computational Intelligence and Data Mining*, pages 348–354, Nashville (USA), 2009.
8. 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):729–747, 2010.
9. 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 International Workshop on Genetic and Evolving Fuzzy Systems (GEFS)*, pages 11–16, Witten-Bommerholz (Germany), 2008.
10. 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. *International Journal of Hybrid Intelligent Systems*, 7(1):45–64, 2010.
11. K. Trawiński, A. Quirin, and O. Cordón. Bi-criteria genetic selection of bagging fuzzy rule-based multiclassification systems. In *IFSA/EUSFLAT Conf.*, pages 1514–1519, 2009.
12. 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. *International Conference on Intelligent Systems Design and Applications (ISDA)*, 0:121–127, 2009.
13. D. Partridge and W.B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.
14. O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. World Scientific, 2001.
15. O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems*, 141(1):5–31, 2004.
16. F. Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46, January 2008.
17. K. Trawiński, O. Cordón, and A. Quirin. On designing fuzzy multiclassifier systems by combining furia with bagging and feature selection. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 19(4):589–633, 2011.
18. J. C. Hühn and E. Hüllermeier. FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.
19. J. C. Hühn and E. Hüllermeier. An analysis of the FURIA algorithm for fuzzy rule induction. In *Advances in Machine Learning I*, pages 321–344. 2010.
20. T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
21. R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.
22. T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
23. 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.
24. C.A. Coello, G.B. Lamont, and D.A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd Edition*. Springer, 2007.
25. Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.
26. D. Ruta and B. Gabrys. Classifier selection for major-

ity voting. *Information Fusion*, 6(1):63–81, 2005.
27. A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.
28. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
29. E.M. Dos Santos, R. Sabourin, and P. Maupin. Single and multi-objective genetic algorithms for the selection of ensemble of classifiers. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3070–3077, Vancouver, 2006.
30. 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):2993–3009, 2008.
31. T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
32. R.E. Banfield, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180, 2007.
33. D. Optiz and R. Maclin. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.
34. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
35. R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
36. Z.H. Zhou. Ensembling local learners through multimodal perturbation. *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):725–735, 2005.
37. 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):418–435, 1992.
38. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
39. M.J. del Jesus, F. Hoffmann, L.J. Navascues, and L. Sánchez. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3):296–308, 2004.
40. L. Sánchez and J. Otero. Boosting fuzzy rules in classification problems under single-winner inference. *International Journal of Intelligent Systems*, 22(9):1021–1034, 2007.
41. H. Takahashi and H. Honda. Lymphoma prognostication from expression profiling using a combination method of boosting and projective adaptive resonance theory. *Journal of Chemical Engineering of Japan*, 39(7):767–771, 2006.
42. C. Z. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(1):1–14, 1998.
43. J.J. Aguilera, M. Chica, M.J. del Jesus, and F. Herrera. Niching genetic feature selection algorithms applied to the design of fuzzy rule based classification systems. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1794–1799, London (UK), 2007.
44. Y. Nojima and H. Ishibuchi. Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion. In *International Conference on Hybrid Intelligent Systems and Conference on Neuro-Computing and Evolving Intelligence*, Auckland, New Zealand, 2006.
45. H. Ishibuchi and Y. Nojima. Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers. *International Journal of Hybrid Intelligent Systems*, 3(3):129–145, 2006.
46. R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *International Conference on Machine Learning*, pages 322–330, Nashville (USA), 1997.
47. B.V. Dasarathy and B.V. Sheela. A composite classifier system design: Concepts and methodology. *Proceedings of IEEE*, 67(5):708–713, 1979.
48. L.S. Oliveira, M. Morita, R. Sabourin, and F. Bortolozzi. Multi-objective genetic algorithms to create ensemble of classifiers. *Lecture Notes in Computer Science*, 3410:592–606, 2005.
49. W. W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.
50. H. Ishibuchi, T. Nakashima, and T. Morisawa. Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*, 103(2):223–238, 1999.
51. O. Cordón, M.J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20:21–45, 1999.
52. P. Panov and S. Džeroski. Combining bagging and random subspaces to create better ensembles. In *IDA'07: Proceedings of the 7th international conference on Intelligent data analysis*, pages 118–129, Berlin, Heidelberg, 2007. Springer-Verlag.
53. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3:257–271, 1999.
54. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Fran-

cisco, CA, USA, 1993.
55. T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
56. J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *2002 Congress on Evolutionary Computation CEC '02*, volume 1, pages 711–716, 2002.

3 A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers

This publication was published in the best journal from the Artificial Intelligence category:

- K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez. A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers, IEEE Transactions of Fuzzy Systems, vol. 21, no 5, pp. 950-965, 2013, 2013. DOI: 10.1109/TFUZZ.2012.2236844.
  - State: Published.
  - Impact Factor (JCR 2012): 5.484.
  - Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 1/115. Q1.

A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers

Krzysztof Trawiński, Oscar Cordón, *Senior Member, IEEE*, Luciano Sánchez, *Member, IEEE*, and Arnaud Quirin

***Abstract*—Fuzzy set theory has been widely and successfully used as a mathematical tool to combine the outputs provided by the individual classifiers in a multiclassification system by means of a fuzzy aggregation operator. However, to the best of our knowledge, no fuzzy combination method has been proposed, which is composed of a fuzzy rule-based system. We think this can be a very promising research line as it allows us to benefit from the key advantage of fuzzy systems, i.e., their interpretability. By using a fuzzy linguistic rule-based classification system as a combination method, the resulting classifier ensemble would show a hierarchical structure, and the operation of the latter component would be transparent to the user. Moreover, for the specific case of fuzzy multiclassification systems, the new approach could also become a smart way to allow standard fuzzy classifiers to deal with high-dimensional problems, avoiding the curse of dimensionality, as the chance to perform classifier selection at class level is also incorporated, into the method. We conduct comprehensive experiments considering 20 UCI datasets with different dimensionality, where our approach improves or at least maintains accuracy, while reducing complexity of the system, and provides some interpretability insight into the multiclassification system reasoning mechanism. The results obtained show that this approach is able to compete with the state-of-the-art multiclassification system selection and fusion methods in terms of accuracy, thus providing a good interpretability–accuracy tradeoff.**

***Index Terms*—Bagging, classifier fusion, classifier selection, fuzzy rule-based multiclassification systems, genetic fuzzy systems, interpretability–accuracy tradeoff, linguistic selection and fusion of individual classifiers.**

I. Introduction

MULTICLASSIFICATION systems (MCSs), which are also called classifier ensembles, are machine learning tools capable of obtaining better performance than a single classifier when dealing with complex classification problems, especially when the number of dimensions or the size of the data is really large [1]. The most common base classifiers are decision trees [2], neural networks [3], and, more recently, fuzzy classifiers [4]–[8].

MCS design is mainly based on two stages [9]: the learning of the component classifiers and the combination mechanism for the individual decisions provided by them into the global MCS output. Since an MCS is the result of the combination of the outputs of a group of individually trained classifiers, the accuracy of the finally derived MCS relies on the performance and the proper integration of these two tasks. The best possible situation for an ensemble is where the individual classifiers are both accurate and diverse, in the sense that they make their errors on different parts of the problem space [3]. Hence, MCSs rely for their effectiveness on the "instability" of the base learning algorithm. On the one hand, the correct definition of the set of base classifiers is fundamental to the overall performance of MCSs. Different approaches have been, thus, proposed to succeed on generating diverse component classifiers with uncorrelated errors such as data resampling techniques (mainly, bagging [10] and boosting [11]), specific diversity induction mechanisms (feature selection [2], diversity measures [12], use of different learning models, etc.), or combinations between the latter two families (data resampling and specific diversity induction mechanisms) [13].

On the other hand, the research area of combination methods is also very active. It does not only consider the issue of aggregating the results provided by all the initial set of component classifiers derived from the first learning stage to compute the final output (what is usually called *classifier fusion* [14], [15]). It also involves either locally selecting the best single classifier which will be taken into account to provide a decision for each specific input pattern (static or dynamic classifier selection [16]) or globally selecting the subgroup of classifiers which will be considered for every input pattern (overproduce-and-choose strategy (OCS) [17]). Besides, hybrid strategies between the two groups have also been introduced [1].

While the weighted majority voting (MV) could be considered as the most extended fusion-based combination method [18], many other proposals have been developed in the specialized literature [19], including several successful procedures based on the use of fuzzy set theory and, specifically, of fuzzy aggregation operators [20], [21]. However, up to our knowledge, there has not been any previous proposal of an MCS combination method considering the use of a fuzzy linguistic system (specifically, a fuzzy rule-based classification system (FRBCS)) to accomplish this task.

To our mind, that alternative constitutes a very smart design as it carries several advantages. First, it provides the MCS with

Manuscript received May 21, 2012; revised September 28, 2012; accepted December 12, 2012. Date of publication January 3, 2013; date of current version October 2, 2013. This work was supported by the Spanish Ministerio de Economía y Competitividad under Projects TIN2012-38525-C02-01 and TIN2011-24302, including funding from the European Regional Development Fund.

K. Trawiński is with the European Centre for Soft Computing, 33600 Mieres, Spain (e-mail: krzysztof.trawinski@softcomputing.es).

O. Cordón is with the European Centre for Soft Computing, 33600 Mieres, Spain, and also with the Department of Computer Science and Artificial Intelligence and the Research Center on Information and Communication Technologies, University of Granada, 18071 Granada, Spain (e-mail: oscar.cordon@softcomputing.es).

L. Sánchez is with the Department of Computer Science, University of Oviedo, 33203 Gijón, Spain (e-mail: luciano@uniovi.es).

A. Quirin was with the European Centre for Soft Computing, 33600 Mieres, Spain. He is now with Gradiant, 36310 Vigo, Spain (e-mail: arnaud.quirin@softcomputing.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2012.2236844

a higher degree of interpretability, making the combination method operation more transparent for the user. In the same way, the resulting MCS design takes a pure hierarchical structure, as in other classical approaches such as stacking [22]. Besides, by using an advanced learning technique incorporating the feature selection capability to derive the FRBCS implementing the combination method, it can deal with both classifier fusion and selection, thus allowing us to reduce the MCS complexity while improving its generalization ability. Finally, by combining a set of component fuzzy classifiers by means of an FRBCS-based combination method (FRBCS-CM), we will come up with a proper approach to deal with the curse of dimensionality always present when applying a fuzzy system to a high-dimensional problem [23]. This issue is addressed by means of a genetic fuzzy system (GFS) [24]–[27] in recent studies [28], [29].

In this paper, we introduce a framework to derive an FRBCS playing the role of the MCS combination method. This fuzzy linguistic combination method can be applied to any classifier ensemble structure with the only restriction that the component classifiers must additionally provide certainty degrees associated with each class in the dataset. The fuzzy linguistic combination method presents an interpretable structure as it is based on the use of a single disjunctive fuzzy classification rule per problem class, as well as on the classical single-winner fuzzy reasoning method [23], [30]. The antecedent variables correspond to the component (fuzzy) classifier (and thus, its number is bounded by the existing component classifiers), and each of them has a weight associated, representing the certainty degree of each ensemble member in the classification of each class. A specific genetic algorithm (GA) to design such an FRBCS-CM will be proposed with the ability of selecting features and linguistic terms in the antecedent parts of the rules. In such a way, it will perform both classifier fusion and classifier selection at class level. The resulting system is, thus, GFS (in particular, a genetic fuzzy rule-based classification system (GFRBCS)), dealing with the interpretability–accuracy tradeoff in fuzzy MCS design in a proper way [31].

In this paper, the novel FRBCS-CM will be applied on fuzzy rule-based multiclassification systems (FRBMCSs) generated from bagging [8], where the base classifier (an FRBCS) directly incorporates the feature selection ability. Therefore, the resulting FRBMCS will show a clear hierarchical structure composed of two levels of fuzzy classifiers, allowing it to deal with high-dimensional problems: a first lower level with the fuzzy classifier composing the FRBMCS and a second upper level integrated by the FRBCS-CM combining the outputs of the latter. A comprehensive study will be conducted on 20 datasets of different dimensions from the UCI machine learning repository to test the accuracy and complexity of the derived FRBMCSs. First, we will analyze the introduced fuzzy linguistic combination method. Then, we compare the novel FRBCS-CM with the state-of-the-art crisp and fuzzy multiclassification combination methods, as well as with a hybrid method based on a GA, considering both classifier selection and classifier fusion [32]. Finally, we will show some interpretability aspects of the proposed FRBCS-CM.

This paper is organized as follows. In the next section, the preliminaries that are required for a good understanding of our work (i.e., MCS combination methods, fuzzy MCS combination methods, and our approach to design FRBMCSs considering bagging) are reviewed. Section III describes the proposed FRBCS-CM framework and structure, as well as the GA considered to design it. The experiments developed and their analysis are shown in Section IV. Finally, Section VI collects some concluding remarks.

II. Preliminaries

This section explores the current literature related to classifier ensemble combination methods and reviews our generation method for FRBMCSs.

A. Multiclassification System Combination Methods

Two main approaches arise in the literature for the combination of the outputs provided by a previously generated set of base classifiers into a single MCS output [15]: *classifier fusion* and *classifier selection*.

Classifier fusion relies on the assumption that all ensemble members make independent errors. Thus, combining the decisions of the ensemble members may lead to increasing the overall performance of the system (MV, sum, product, maximum, and minimum are commonly used functions [19]). However, this family of methods carries the following drawback: There is no guarantee that a particular ensemble generation technique will achieve the error independence, and thus, it does not improve the final classification performance. That is the reason for the extended use of weighted MV [18], [33]–[35].

Alternatively, classifier selection is based on the fact that not all the individual classifiers but only a subset of them will influence on the final decision for each input pattern. On the one hand, a general family of classifier selection methods assumes that each individual classifier is an expert in some local regions of the problem space [36], thereby avoiding the error independence assumption. In this approach, the accuracy of each classifier surrounding the region of the feature space, where the unknown pattern to be classified is located, is previously estimated, and the best one is selected to classify that specific pattern. Two categories of classifier selection techniques exist: static and dynamic [15], [16]. In the first case, regions of competence are defined during the training phase, while in the second case, they are defined during the classification phase based on the attributes of the sample to be classified (for dynamic classifier selection, see [37]). Nevertheless, there is a drawback to both selection strategies: When the local expert does not classify the test pattern correctly, misclassification cannot be avoided.

On the other hand, there is another family of static classifier selection methods based on the assumption that candidate classifiers could be redundant. In [38], it was formally shown that finding the most relevant subset of classifiers leads to better performance than combining all the available classifiers. These methods are grouped under the name of OCS [17] (also known as test-and-select methodology [39]). They are based on the fact that a large set of candidate classifiers is generated and then selected (removing duplicates and poor-performing candidate classifiers) to extract the best performing subset which composes the final MCS used to classify the whole test set. GAs are commonly used for this task [7], [40], [41], [42]. Consequently, OCS methods determine the optimal ensemble size by considering a tradeoff between accuracy and complexity. However, OCS could be subject to overfitting, as a fixed subset of classifiers that is defined using a training/optimization dataset may not be well adapted for the classification of every pattern in the test set [37].

In order to overcome the problems of each family in isolation, hybrid methods between both families have been proposed [32], [37], [43]. The latter approach is worth mentioning, as the authors proposed a GA selecting the contribution of the component classifiers for the final decision at each specific class level (i.e., the decision of a weak learner can be considered to classify class A and not considered to classify class B).

In this paper, we will follow the latter approach since our FRBCS-CM belongs to static OCS being able to either completely remove a whole candidate classifier or to reduce its role to only some specific classes with a specific weight measuring our confidence in the base classifier for that specific class. All of that will be performed using a human-interpretable structure generated by means of a GFRBCS.

B. Multiclassification System Fuzzy Combination Methods

Fuzzy set theory has been extensively and successfully considered for MCS combination, especially classifier fusion, as fuzzy aggregation operators are able to model the imprecision and uncertainty involved in the MCS combination process [44]. Two different groups of fuzzy operators have been considered in the literature [34]: 1) *the classical simple fuzzy aggregation operators*, such as minimum, maximum, simple average, or product; and 2) *more advanced fuzzy operators*, including the fuzzy integral [45], the BAsic Defuzzification Distributions (BADD) defuzzification strategy [46], Zimmermann's compensatory operator [47], and the decision templates [48].

Some studies have developed experimental comparisons of the performance of different MCSs considering the latter fuzzy connectives as fusion combination operator [20], [21]. Additionally, in [21], it was shown that fuzzy combination methods outperformed nonfuzzy ones and that decision templates based on Euclidean distance and fuzzy integral were the best methods overall, when applied with boosting.

Besides, some other works have extended the scope of the latter [44], [49], [50]. For example, Lu and Yamaoka [50] introduced a fuzzy combination method based on a complex fuzzy reasoning process. Although this method uses an FRBS as a refinement module for the fuzzy combination method decisions, this strategy shows several problems such as its specificity to the consideration of a simple three-classifier ensemble, its highly complex structure composed of two different nature fuzzy reasoning modules, the need of manually defining the fuzzy rules in the refinement module,[1] and the impossibility to perform classifier selection (which, of course, is not required in the simple ensemble structure considered).

The proposal made in this paper is aimed to solve all the latter drawbacks by designing a single fuzzy linguistic combination method in the form of a fully understandable FRBCS, automatically derived by a GFRBCS, which shows the capability of performing both classifier fusion and selection.

[1]This could be feasible when using a very small number of component classifiers—only three—but not dealing with a more usual larger number. In fact, the FRBSs considered in their experimentation are only composed of a single rule with three inputs, and the authors mention that they were not able to incorporate expert knowledge to the Bayesian component classifier.

C. Bagging Fuzzy Multiclassification Systems

In this paper, we will follow a methodology for component fuzzy classifier generation that we previously presented in [8]. To generate FRBCMSs, we embedded fuzzy unordered rules induction algorithm (FURIA) [51], [52] into an MCS framework based on classical MCS design approaches [2], [10], [53]. We concluded that pure bagging without additional feature selection obtained the best performance when combined with FURIA-based FRBCSs. Thus, we consider the use of bagging with the entire feature set to generate initial FURIA-based fuzzy MCSs.

In order to build these FRBMCSs, a normalized dataset is split into two parts: a training set and a test set. The training set is submitted to an instance selection procedure in order to provide the K individual training sets (the so-called *bags*) to train the K FURIA-based fuzzy FRBCSs. In every case, the bags are generated with the same size as the original training set, as commonly done.

The fuzzy classification rules R_j^k considered show a class C_j^k and a certainty degree CF_j^k in the consequent: If x_1^k is A_{j1}^k and ... and x_n^k is A_{jn}^k, then Class C_j^k with CF_j^k, $j = 1, 2, \ldots, N, k = 1, 2, \ldots, K$.

After performing the training stage on all the bags in parallel, we get an initial FRBMCS, which is validated using training and test errors as well as a complexity measure based on the total number of rules in the FRBCSs. The voting-based fuzzy reasoning method is used to take the decision provided by each weak learner. Thus, the class with the highest accumulated degree is the one assigned to each component classifier. Finally, the MV is applied as a fusion method: The class with the most votes among all the classifiers is selected as the final output. The lowest order class is taken in the case of a tie.

Regardless of the fuzzy rule generation method considered to derive the component FRBCSs (in this paper, we use FURIA due to its capability to generate accurate and compact fuzzy classifiers for high-dimensional datasets, but any method generating fuzzy classification rules with a certainty degree could be used), the two-level hierarchical structure composed of the individual classifiers in the first level and the proposed FRBCS-CM in the second will be maintained, allowing the system to deal with high-dimensional problems in a proper way, while maintaining their descriptive power.

III. Genetic Fuzzy Classifier System to Design a Fuzzy Linguistic Combination Method For (Fuzzy) Classifier Ensembles

The next sections will, respectively, introduce the global framework of our novel fuzzy linguistic combination method

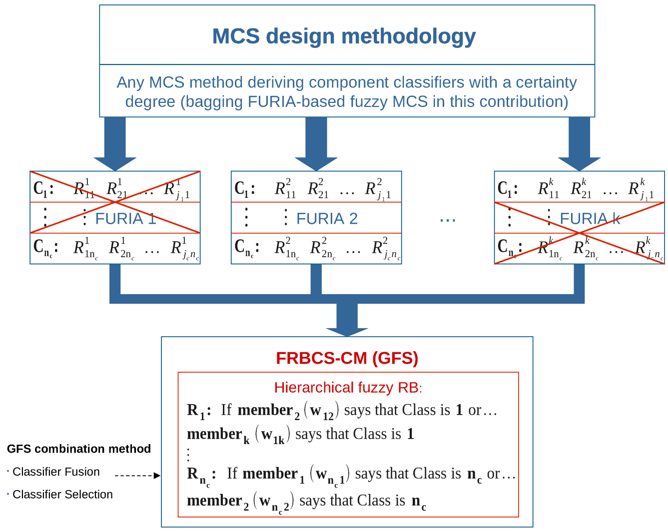


Fig. 1. Our framework: After the instance selection, the component classifiers are derived by the FRBCS learning method (or by any method deriving component classifiers with a certainty degree). Then, the fuzzy linguistic rule-based classification system playing the role of MCS combination method (an FRBCS-CM) selects rules with a proper behavior in order to obtain a good interpretability–accuracy tradeoff. Finally, the output is obtained using the FRBCS-CM fuzzy reasoning mechanism.

and provide a detailed description of the FRBCS-CM structure and of the composition of the GFS designed to derive its fuzzy knowledge base.

A. Framework

As mentioned previously, the aim of this paper is to design a fuzzy linguistic rule-based classification system playing the role of MCS combination method (an FRBCS-CM). Our design must fulfill several requirements, namely, 1) showing a human-understandable structure; 2) being able to deal with high-dimensional problems avoiding the curse of dimensionality; 3) having the chance to be automatically learned from training data; and 4) being able to perform both classifier fusion and selection in order to derive low complexity fuzzy classifier ensembles with a good accuracy–complexity tradeoff (see Fig. 1).[2]

Using the novel FRBCS-CM together with a fuzzy classifier ensemble, we will have the additional advantage of handling a two-level hierarchical structure composed of the individual classifiers in the first level and the FRBCS-CM in the second. These kinds of hierarchical structures [56]–[59] are well known in the area as they allow fuzzy systems to properly deal with high-dimensional problems, while maintaining their descriptive power, especially when considering the single-winner rule fuzzy reasoning method in the component fuzzy classifiers as done in our case. One step further, using it in combination with a bagging fuzzy classifier ensemble strategy, as done in this paper, we can also benefit from some collateral advantages for the overall design of the FRBMCS: 1) the simplicity of the implicit parallelism of bagging, which allows for an easy parallel implementation; and 2) the problem partitioning due to the internal feature selection at the component classifier level and the classifier selection capability of the fuzzy linguistic combination method, resulting in a tractable dimension for learning fuzzy rules for each individual classifier and for achieving a compact fuzzy classifier ensemble. These characteristics will make the fuzzy ensemble using the FRBCS-CM especially able to deal with the curse of dimensionality.

Our approach might, thus, be assigned to the stacking (or stacked generalization) group [22], which after bagging and boosting is probably the most popular approach in the literature. Its basis lies in the definition of the metalearner, playing a role of the (advanced) MCS combination method, giving a hierarchical structure of the ensemble. Its task is to gain knowledge of whether training data have been properly learned and to be able to correct badly trained base classifiers. The FRBCS-CM proposed in this paper acts as the metalearner, by discarding the rule subsets in the base fuzzy classifiers providing incorrect decisions at individual class level and promoting the ones leading to a correct classification.

Moreover, fuzzy classification rules with a class and a certainty degree in the consequent used in the FRBCS-CM allow the user to get an understandable insight into the MCS. This means that this approach will allow interpretability (to some extent) of such a complicated system.

The proposed FRBCS-CM is built under the GFS approach. A specific GA, which uses a sparse matrix to codify features and linguistic terms in the antecedent parts of the rules, performs both classifier fusion and classifier selection at class level.

B. Fuzzy Linguistic Combination

As described in Section II-C, the FRBCSs considered in the ensemble will be based on fuzzy classification rules with a class and a certainty degree in the consequent. Let R_j^k be the jth rule of the kth member of an ensemble of K components:

$$\text{if } x \text{ is } A_j^k, \text{ then Class } C_j^k \text{ with } CF_j^k$$

where $C_j^k \in \{1, \ldots, n_c\}$, and n_c is the number of classes.

We will use the expression $\mathcal{G}^k = \{R_1^k, \ldots, R_{N_k}^k\}$ to denote the list of fuzzy rules comprising the kth ensemble member. Let us partition each one of these lists into so many sublists $\mathcal{G}_c^k$ as classes existing in the problem. $\mathcal{G}_c^k$ contains the rules of $\mathcal{G}^k$ whose consequent is the class c.

Let us also define $R^k(x)$ as the intermediate output of the kth ensemble member, which is the fuzzy subset of the set of class labels computed as follows:

$$R^k(x)(c) = \bigvee_{\{j|C_j^k=c\}} CF_j^k \cdot A_j^k(x). \qquad (1)$$

Each component of the FRBCS maps an input value x to so many degrees of membership as the number of classes in the problem. The highest of these memberships determines the classification of the pattern. That is to say, the kth

[2]We should remind that the proposed combination method can be applied to any MCS with the only restriction that the component classifiers must additionally provide certainty degrees associated with each class in the dataset. For example, one of the novel approaches to design fuzzy classifiers proposed in [54] and [55] could be used.

FRBCS classifies an object x as being of class $\text{FRBCS}^k(x) = \arg\max_{c\in\{1,\ldots,n_c\}} R^k(x)(c)$. Observe also that $R^k(x)(c)$ is the result of applying the fuzzy reasoning mechanism to the knowledge base defined by the sublist $\mathcal{G}_c^k$.

The simplest linguistic combination of the component FRBCSs consists of stacking a selection of some of the rules R_j^k into a single large rule base. Let us define a binary matrix $[b_{ck}] \in \{0,1\}^{n_c\times K}$, and let us agree that if b_{ck} is zero, then $\mathcal{G}_c^k$ is removed from the ensemble, and $R^k(x)(c) = 0$. This selection is equivalent to an hierarchical FRBCS comprising n_c expressions of the form:

if (member$_1$ says that class is c) or ...

or (member$_K$ says that class is c), then class is c

where the asserts "(member$_k$ says that class is c)" have a degree of certainty b_{ck} determined by the rules in the sublist $\mathcal{G}_c^k$, and those asserts for which b_{ck} is zero are omitted. The fuzzy output of this selected ensemble is

$$R^I(x)(c) = \bigvee_{\{(j,k)|C_j^k=c\}} b_{ck}\cdot CF_j^k \cdot A_j^k(x). \tag{2}$$

We can define more powerful linguistic selections which extend this basic fuzzy reasoning schema. In this paper, we will use a *sparse* matrix of weights $[w_{ck}] \in [0,1]^{n_c\times K}$ and operate as follows:

$$R^{II}(x)(c) = \bigvee_{\{(j,k)|C_j^k=c\}} w_{ck}\cdot CF_j^k \cdot A_j^k(x). \tag{3}$$

Thus, the selected ensemble can be seen as a hierarchical knowledge base with n_c fuzzy classification rules with weights in the antecedent part

if (member$_1(w_{c1})$ says that class is c) or ... (member$_K(w_{cK})$

says that class is c), then class is c

where the asserts "(member$_k(w_{ck})$ says that class is c)" have a certainty determined by the rules in the sublist $\mathcal{G}_c^k$, after multiplying their confidence degrees by the same factor w_{ck}:

if x is A_j^k, then Class C_j^k with $w_{C_j^k k}\cdot CF_j^k$.

Again, those rules where $w_{C_j^k k} = 0$ are omitted.

In this case, any of these hierarchical rule bases that we have introduced is univocally determined by a matrix $[w_{ck}]$. Therefore, the genetic search of the best selection involves finding the best matrix $[w_{ck}]$ according to certain criteria that will be defined next. Notice that this search is a selection, because $[w_{ck}]$ is a sparse matrix. As we will explain later, in this paper, the number of terms of w_{ck} different from zero is a design parameter.

C. Fitness Function

We propose that the quality of a selected and combined fuzzy ensemble is defined by three components (e, m_1, m_2) (see Fig. 2); thus, the fitness of a possible FRBCS-CM design is a triplet comprising three real numbers:

1) Training error e: We compute the error of each ensemble for a large number of bootstrapped resamples of the training set and use a quantile of the distribution of these errors as the first term of the fitness. This is intended to avoid overfitting when there are outliers in the training set, as well as to detect the most robust selections, which are expected to generalize better. The issue of dealing with overfitting handling in the accuracy measure has also been considered in [60], [61].
2) Error margin m_1: The second component of the fitness function depends on the distance between the misclassified examples and their nearest decision surface. Given an example x, we have approximated this value by the difference between the highest and the second highest term of $R^{II}(x)(c)$, and defined that the error margin of an ensemble is the worst (i.e., the highest) value of this difference for any example x in the training set.
3) Classification margin m_2: The third component depends on the distance between the correctly classified instances and their nearest decision surface, which is approximated as before, by the difference between the highest and the second highest terms in $R^{II}(x)(c)$. In this case, however, the margin of an ensemble is the lowest value of this difference for all the examples of the training set; we seek a decision surface with the highest margin.

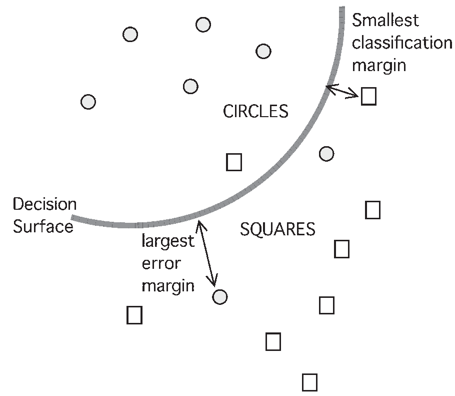


Fig. 2. Fitness of an ensemble has three components. (a) Quantile of the bootstrap estimation of the training error, (b) the largest distance between a misclassified example and the decision surface, and (c) the smallest distance between a correctly classified example and the decision surface.

Given an instance x, let us define "winner rule" as the rule with highest activation and "most promising rule" to classify this pattern as the rule with highest activation among those whose consequent is different from that of the winner rule.

The decision surface is formed by the points for which there is a tie between the activations of the winner rule and the most promising rule. In this respect, if an instance is close to the decision surface, the difference between these two activations will be small. If the instance is moved toward the decision surface, this difference will be further decreased. The opposite is also true: If the instance is separated from the decision surface, this difference will increase. In this respect, we can take this difference as a measure of distance between the instance and the nearest decision surface.

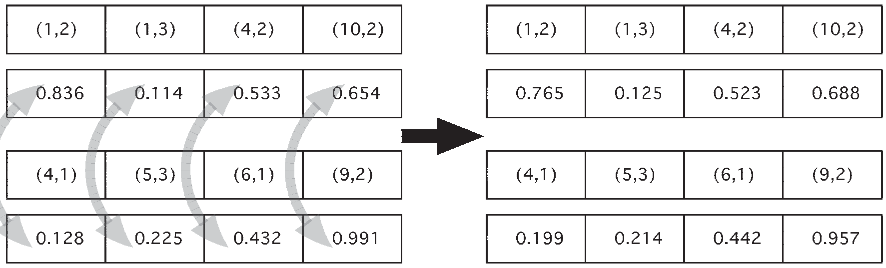


Fig. 3. Coding scheme and crossover operation: An individual is a sparse matrix, which is represented by a list of indices and a list of values.

A lexicographical ordering is defined between two triplets:

$$(e, m_1, m_2) \prec (e', m_1', m_2') \iff \begin{cases} (e < e') \\ (e = e') \text{ and } m_1 < m_1' \\ (e = e') \text{ and } (m_1 = m_1') \text{ and } (m_2 > m_2'). \end{cases} \tag{4}$$

D. Coding Scheme, Genetic Operators, and Evolutionary Scheme

An individual is an sparse matrix $[w_{ck}]$, which will be stored as two fixed-length ordered lists of indexes (c, k) and their corresponding values w_{ck}, as displayed in Fig. 3. The chromosome length is defined according to the maximum percentage of nonzero values in the matrix, which is a parameter whose value is set by the user in advance. The initial population is randomly generated. We have decided to apply an arithmetic crossover [62] between the lists of values of both individuals, leaving the lists of indices unchanged. The mutation operator randomly alternates a nonuniform mutation of an element of the list of values [63] and the random generation of a completely new individual.

Finally, since the fitness function is not scalar, we have decided to implement a tournament-based steady-state GA [64], where at each generation, the two last elements in each tournament are replaced by the offspring of the two winners. This offspring is the result of the application of the crossover operator mentioned before, followed by a mutation with a certain probability.

IV. Experiments and Analysis of Results

This section is devoted to validate our new fuzzy linguistic combination method. While Section IV-A introduces the experimental setup considered, the following ones show the results obtained in the experiments developed and their analysis. In the first place, we compare the performance of FRBCS-CM combined with FURIA-based fuzzy MCSs with bagging with the full ensemble using standard MV in Section IV-B. Then, Section IV-C shows a comparison of the novel FRBCS-CM with the state-of-the-art crisp and fuzzy multiclassification combination methods, as well as with a hybrid method based on GA considering both classifier selection and classifier fusion [32]. For that comparison, apart from the standard MV, we select average (AVG) [1] and decision templates (DT) [48] based on Euclidean distance, as crisp and fuzzy fusion methods, respectively, being the best methods of each group according to Kuncheva [21]. Since the proposed FRBCS-CM includes classifier selection and classifier fusion, we also apply classifier selection with the mentioned classifier fusion methods in order to make a fair comparison. To select classifiers, we will use two standard greedy approaches, i.e., Greedy Forward Selection (FS) and Greedy Backward Selection (BS) [40], which will use the aforementioned classifier fusion methods (these methods are also used to guide the search of the greedy algorithms). The hybrid method based on GA proposed in [32] (GA-Dimililer) embeds both classifier selection and classifier fusion; thus, we directly apply it with no modifications. Then, we will show some interpretability aspects of the fuzzy linguistic combination method proposed in Section IV-D Finally, the runtime values for the fuzzy MCSs, FRBCS-CM, and the other combination methods are presented in Section IV-E.

TABLE I
Datasets Considered

Data set	#examples	#attr.	#classes
Low dimensional:			
abalone	4178	7	28
breast	700	9	2
glass	214	9	7
heart	270	13	2
magic	19020	10	2
pblocks	5474	10	5
phoneme	5404	5	2
pima	768	8	2
wine	178	13	3
yeast	1484	8	10
High dimensional:			
ionosphere	352	34	2
optdigits	5620	64	10
pendigits	10992	16	10
sat	6436	36	6
segment	2310	19	7
sonar	208	60	2
spambase	4602	57	2
texture	5500	40	11
vehicle	846	18	4
waveform	5000	40	3

A. Experimental Setup

To evaluate the performance of the FRBCS-CM in the ensembles generated, 20 popular datasets from the UCI machine learning repository have been selected (see Table I). In all of them, every attribute is continuous. As can be seen, the number of features ranges from a small value (i.e., 5) to a large one (i.e., 64), while the number of examples does so from 208 to 19 020. We divided them into two groups with low dimensionality (with < 15 attr.) and with high dimensionality (with ≥ 15 attr.), as can be seen in Table I.

In order to compare the accuracy of the considered classifiers, we used Dietterich's 5×2-fold cross-validation (5×2-cv) [65]. The bagging FRBMCSs generated are initially comprised by 50 classifiers.

The steady-state GA for the FRBCS-CM derivation works with a population of 100 individuals and runs for 1000 generations (the equivalent to 20 generations in a usual GA with generational replacement and crossover probability equal to 1). The tournament size is 5, and the mutation probability considered is 0.1. Five different values have been tested for the chromosome size: 10%, 25%, 50%, 75%, and 90% of the terms of the original $[w_{ck}]$ matrix were allowed to be nonzero, thus corresponding to a maximum reduction of the 90%, 75%, 50%, 25%, and 10% of the original FRBMCS size, respectively. The combination methods based on AVG and DT do not require any predefined parameters. The former is a mathematical function averaging all the component classifier votes for each class, while the latter is based on learning specific characteristics of examples of the given class by taking an average of them during the learning phase. Then, in the test phase, the final decision is based on a distance of the given examples to the obtained templates for each class.

In order to make a fair comparison, we use the same parameters for GA-Dimililer as for our GA to obtain the FRBCS-CM. GA-Dimililer is based on a binary coding, where crisp votes of each component classifier for each class are assigned one binary value. When a binary value is equal to 1, it means that a corresponding vote of the given component classifier is activated during the testing phase. The fitness function uses a well-known F-score in order to guide the genetic search. The final decision is taken based on weighted MV. For more details, see [32].

The Wilcoxon Signed-rank test has been used for a deeper insight of the results. Unlike the commonly used t-$test$, the Wilcoxon test does not assume normality of the samples [66], which would be unrealistic in the case of the UCI datasets. The confidence level considered for the null hypothesis rejection is 5%.

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 GB of memory, under the Linux operating system.

Since there are several parameter values and aspects to be tested, the analysis of the obtained results will be performed in parts and following an incremental approach for the sake of comprehensibility.

B. Analysis of Fuzzy Rule-Based Classification System-Based Combination Method Combined With Fuzzy Multiclassification System Design Approach

In the first place, we have conducted experiments on the FRBCS-CM combined with FURIA-based fuzzy MCSs with bagging. We aim to test the performance of the novel fuzzy linguistic combination method in comparison with the whole initial fuzzy classifier ensemble. The results are presented in Table II (the best result for a given dataset is presented in bold font), showing the test error obtained for MV (operating on the full original ensemble) and FRBCS-CM (nonzero values: 10%, 25%, 50%, 75%, and 90%), as well as the accuracy improvement percentage for each of the five different parameters considered for FRBCS-CM with respect to the accuracy of MV.

From the viewpoint of Table II, it can be noticed that in almost all cases, FRBCS-CMs show an accuracy improvement of FURIA-based fuzzy MCSs with bagging with respect to MV. Only for 10% of nonzero values, the average improvement was negative, considering both low- and high-dimensional datasets. It also happens for 25% of nonzero values, considering low-dimensional datasets. Considering the datasets separately, the FRBCS-CM outperforms MV in 17 out of 20 cases in at least one of the FRBCS-CMs designed. Breast, pima, and abalone are the only three datasets, where MV showed performance advantage in comparison with the FRBCS-CMs.

From the viewpoint of Table III, it can be noticed that FURIA-based fuzzy MCSs with bagging obtain roughly 2035 overall average number of rules for the full ensemble, while the FRBCS-CM combined with FURIA-based fuzzy MCSs with bagging obtains much less rules, roughly 211, 511, 1033, 1525, and 1829 overall average number of rules for 10%, 25%, 50%, 75%, and 90% of nonzero values, respectively. As expected, the nonzero value parameter is strongly correlated (namely, it is an inverse correlation) with the reduction of the number of rules.

In conclusion, the FRBCS-CM with 75% and 90% of nonzero values is not only able to outperform MV fusion mechanism operating on the full original ensemble (which is not the aim in itself) but is also very competitive in terms of complexity reduction, after the selection of the component classifiers. Even when considering a nonzero parameter with low values (such as 10% or 25%), it results in a strong complexity reduction (roughly 90% and 75% less rules than a full ensemble), while our approach is able to maintain the original accuracy. Moreover, it allows the user to get some insights of the MCS reasoning mechanism, which increases interpretability of the generated MCSs (as will be seen in Section IV-D). In this study, we are looking for a good interepretability–accuracy tradeoff through the proposed MCS combination method.

To confirm the latter assumptions, Table IV presents the p-values of the statistical tests performed in order to check if the Wilcoxon test shows significant differences between FRBCS-CM (nonzero values: 10%, 25%, 50%, 75%, and 90%) and the full ensemble with MV in terms of accuracy (while reducing complexity). The results showing a significant differences between both algorithms are presented in bold font. For all the nonzero values considered apart from 10%, the Wilcoxon test shows that the FRBCS-CM is at least significantly not different from the full ensemble with MV. **Even more, for 50%, 75%, and 90%, significant differences are shown in favor of the FRBCS-CM**. Excluding the FRBCS-CM with 10%, which provides roughly ten times simpler fuzzy MCSs, the FRBCS-CM is able to maintain accuracy (and in some cases even improve), while obtaining the complexity reduction.

TABLE II
ACCURACY OF FRBCS-CM COMBINED WITH FURIA-BASED FUZZY MCSS WITH BAGGING

	MV	FRBCS-CM 10%		FRBCS-CM 25%		FRBCS-CM 50%		FRBCS-CM 75%		FRBCS-CM 90%	
Dataset	Test err.	Test err.	Improv.	Test err.	Improv.	Test err.	Improv.	Test err.	Improv.	Test err.	Improv.
Low dim.:											
abalone	**0.7458**	0.7581	-1.2305	0.7537	-0.7947	0.7493	-0.3496	0.7470	-0.1197	0.7461	-0.0335
breast	**0.0409**	0.0472	-0.6295	0.0469	-0.6002	0.0452	-0.4285	0.0438	-0.2857	0.0432	-0.2284
glass	0.2822	0.3159	-3.3645	0.2879	-0.5608	0.2832	-0.0935	**0.2692**	1.3084	0.2710	1.1215
heart	0.1822	0.1785	0.3704	0.1733	0.8889	0.1719	1.0371	**0.1696**	1.2593	0.1696	1.2593
magic	0.1346	0.1340	0.0631	0.1314	0.3270	0.1309	0.3764	0.1302	0.4437	**0.1300**	0.4595
pblocks	0.0288	0.0285	0.0329	0.0265	0.2375	0.0271	0.1754	0.0268	0.2010	**0.0261**	0.2741
phoneme	0.1332	0.1277	0.5514	**0.1252**	0.7994	0.1261	0.7069	0.1256	0.7550	0.1264	0.6847
pima	**0.2385**	0.2492	-1.0677	0.2484	-0.9896	0.2411	-0.2604	0.2432	-0.4687	0.2424	-0.3906
wine	0.0393	0.0461	-0.6742	0.0382	0.1124	**0.0303**	0.8989	0.0404	-0.1124	0.0393	0.0000
yeast	0.4008	0.4155	-1.4690	0.4054	-0.4582	**0.3985**	0.2291	0.4034	-0.2561	0.4013	-0.0539
Avg. Low	0.2227	0.2301	-0.7418	0.2237	-0.1038	0.2204	0.2292	0.2199	0.2725	**0.2196**	0.3093
High dim.:											
ionosphere	0.1459	0.1527	-0.6841	0.1413	0.4575	0.1458	0.0013	**0.1430**	0.2860	0.1430	0.2864
optdigits	0.0329	0.0337	-0.0854	0.0327	0.0142	0.0327	0.0142	0.0318	0.1068	**0.0313**	0.1566
pendigits	0.0156	0.0174	-0.1820	0.0152	0.0346	0.0140	0.1619	0.0140	0.1547	**0.0138**	0.1783
sat	0.1021	0.1067	-0.4662	0.1027	-0.0622	0.0997	0.2331	**0.0986**	0.3418	0.1005	0.1616
segment	0.0336	0.0334	0.0173	0.0319	0.1645	0.0304	0.3203	0.0316	0.1991	**0.0302**	0.3377
sonar	0.2269	0.2404	-1.3462	0.2183	0.8654	0.2077	1.9231	0.2077	1.9231	**0.2058**	2.1154
spambase	0.0587	0.0569	0.1825	0.0559	0.2869	0.0555	0.3217	**0.0539**	0.4825	0.0546	0.4086
texture	0.0307	0.0343	-0.3564	0.0312	-0.0509	0.0304	0.0255	0.0291	0.1636	**0.0285**	0.2218
vehicle	0.2726	0.2773	-0.4728	**0.2664**	0.6147	0.2690	0.3546	0.2664	0.6147	0.2674	0.5201
waveform	0.1492	0.1554	-0.6200	0.1490	0.0160	0.1503	-0.1120	0.1489	0.0240	**0.1479**	0.1280
Avg. High	0.1068	0.1108	-0.4013	0.1045	0.2341	0.1036	0.3244	0.1025	0.4296	**0.1023**	0.4514
Avg. All	0.1647	0.1704	-0.5715	0.1641	0.0651	0.1620	0.2768	0.1612	0.3511	**0.1609**	0.3804

TABLE III
COMPLEXITY OF FRBCS-CM COMBINED WITH FURIA-BASED FUZZY MCSS WITH BAGGING

	MV	FRBCS-CM 10%		FRBCS-CM 25%		FRBCS-CM 50%		FRBCS-CM 75%		FRBCS-CM 90%	
Dataset	# Rules	# Rules	Red. %	# Rules	Red. %	# Rules	Red. %	# Rules	Red. %	# Rules	Red. %
Low dim.:											
abalone	3990.9	398.2	90	995.7	75	1996.9	50	2983.6	25	3578.4	10
breast	435.2	46.1	89	110.9	75	217.0	50	326.2	25	391.0	10
glass	590.3	57.4	90	140.6	76	289.9	51	434.4	26	527.6	11
heart	466.0	49.4	89	120.3	74	235.3	50	352.6	24	420.7	10
magic	3882.1	421.0	89	968.3	75	1965.6	49	2969.9	23	3475.8	10
pblocks	1329.4	131.2	90	328.9	75	628.1	53	967.8	27	1182.2	11
phoneme	2197.3	241.7	89	587.8	73	1132.5	48	1679.0	24	2000.2	9
pima	1050.9	110.9	89	260.7	75	530.1	50	782.4	26	946.3	10
wine	231.4	23.7	90	57.9	75	116.4	50	172.7	25	207.7	10
yeast	2449.0	260.8	89	630.9	74	1198.4	51	1825.1	25	2198.4	10
Avg. Low	1662.3	174.0	90	420.2	75	831.0	50	1249.4	25	1492.8	10
High dim.:											
ionosphere	367.7	37.8	90	95.4	74	211.0	43	279.8	24	333.6	9
optdigits	3584.6	359.2	90	893.5	75	1787.7	50	2678.8	25	3227.2	10
pendigits	4395.3	448.8	90	1098.1	75	2208.7	50	3299.9	25	3964.3	10
sat	4207.2	427.2	90	1046.9	75	2107.2	50	3128.1	26	3762.8	11
segment	1175.3	130.1	89	290.9	75	593.4	50	876.9	25	1051.4	11
sonar	319.3	32.4	90	80.4	75	162.0	49	240.0	25	287.6	10
spambase	2220.9	229.0	90	557.2	75	1115.5	50	1661.7	25	2002.6	10
texture	2912.2	300.1	90	716.6	75	1458.8	50	2175.0	25	2610.9	10
vehicle	1415.3	154.3	89	380.4	73	735.3	48	1075.3	24	1282.6	9
waveform	3484.3	354.0	90	861.5	75	1749.8	50	2601.2	25	3137.6	10
Avg. High	2408.2	247.3	90	602.1	75	1212.9	49	1801.7	25	2166.1	10
Avg. All	2035.2	210.7	90	511.1	75	1022.0	50	1525.5	25	1829.4	10

C. Comparison of Fuzzy Rule-Based Classification System-Based Combination Method and Other Multiclassification System Combination Methods

In this section, we compare the novel FRBCS-CM with some state-of-the-art crisp and fuzzy multiclassification combination methods, as well as with a hybrid method based on GA, considering both classifier selection and classifier fusion [32]. In

TABLE IV
WILCOXON SIGNED-RANK TEST FOR THE COMPARISON OF FRBCS-CM AND THE FULL ENSEMBLE WITH MV

Comparison	p-value
FRBCS-CM 10% vs MV	**-(0.0054)**
FRBCS-CM 25% vs MV	=(0.4221)
FRBCS-CM 50% vs MV	**+(0.0333)**
FRBCS-CM 75% vs MV	**+(0.0152)**
FRBCS-CM 90% vs MV	**+(0.0038)**

TABLE V
COMPARISON OF FRBCS-CM (10% AND 25%) AND GREEDY FS APPROACHES (MV, AVG, AND DT) IN TERMS OF ACCURACY AND COMPLEXITY

	FRBCS-CM 10%		FRBCS-CM 25%		Greedy FS MV		Greedy FS AVG		Greedy FS DT	
Dataset	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules
Low dim.:										
abalone	0.7581	398.2	0.7537	995.7	**0.7524**	1211	0.7582	1047.6	0.7610	1037.7
breast	0.0472	46.1	0.0469	110.9	0.0455	33	0.0418	25.7	**0.0398**	24.1
glass	0.3159	57.4	**0.2879**	140.6	0.2981	88.7	0.3271	43.6	0.3000	54.7
heart	0.1785	49.4	**0.1733**	120.3	0.1859	48.9	0.2015	35.7	0.1874	33.4
magic	0.1340	421.0	**0.1314**	968.3	0.1329	528.2	0.1328	424.6	0.1323	417.3
pblocks	0.0285	131.2	**0.0265**	328.9	0.0282	248.2	0.0302	108.9	0.0296	106.1
phoneme	0.1277	241.7	0.1252	587.8	0.1260	493.2	**0.1232**	381.1	0.1258	339.4
pima	0.2492	110.9	**0.2484**	260.7	0.2503	239.3	0.2516	149.4	0.2596	118.1
wine	0.0461	23.7	**0.0382**	57.9	0.0629	9.1	0.0551	6.8	0.0607	6.2
yeast	0.4155	260.8	**0.4054**	630.9	0.4116	511.5	0.4142	389.5	0.4189	434.9
Avg. Low	0.2301	174.0	**0.2237**	420.2	0.2294	341.1	0.2336	261.3	0.2315	257.2
High dim.:										
ionosphere	0.1527	37.8	**0.1413**	95.4	0.1584	27	0.1532	22.2	0.1646	24.4
optdigits	0.0337	359.2	**0.0327**	893.5	0.0367	652.67	0.0352	428.7	0.0351	423.7
pendigits	0.0174	448.8	0.0152	1098.1	0.0171	892.1	**0.0150**	569.8	0.0162	470.8
sat	0.1067	427.2	0.1027	1046.9	0.1044	1214	0.1010	728.7	**0.1005**	800.6
segment	0.0334	130.1	0.0319	290.9	**0.0318**	165.6	0.0326	109.2	0.0336	86.7
sonar	0.2404	32.4	0.2183	80.4	**0.2163**	24.4	0.2337	22.9	0.2452	19.8
spambase	0.0569	229.0	**0.0559**	557.2	0.0576	340.7	0.0573	286.1	0.0574	292.8
texture	0.0343	300.1	**0.0312**	716.6	0.0343	433.6	0.0330	333.8	0.0336	352.5
vehicle	0.2773	154.3	**0.2664**	380.4	0.2671	364.1	0.2690	173.3	0.2693	193.4
waveform	0.1554	354.0	**0.1490**	861.5	0.1508	1355.9	0.1535	753.1	0.1533	727.1
Avg. High	0.1108	247.3	**0.1045**	602.1	0.1075	547.0	0.1084	342.8	0.1109	339.2
Avg. All	0.1704	210.7	**0.1641**	511.1	0.1684	444.1	0.1710	302.0	0.1712	298.2

order to make a fair study, we will do three different comparisons, which will be structured taking the complexity (number of rules) of the fuzzy MCSs obtained as a base.

In the first study, we will compare the FRBCS-CM with 10% (FRBCS-CM 10%) and 25% (FRBCS-CM 25%) of nonzero values with Greedy FS combined with MV (Greedy FS MV), AVG (Greedy FS AVG), and DT (Greedy FS DT). The operation of Greedy FS approach leads to the selection of small MCSs with a large complexity (number of rules) reduction. For that reason, we have selected FRBCS-CM 10% and 25%, which show the same tendency.

The opposite situation takes place for the Greedy BS approach, which tends to obtain large MCSs with a small complexity (number of rules) reduction. The same happens with the FRBCS-CM with 75% (FRBCS-CM 75%) and 90% (FRBCS-CM 90%) of nonzero values. Hence, we will compare these FRBCS-CMs with Greedy BS combined with MV (Greedy BS MV), AVG (Greedy BS AVG), and DT (Greedy BS DT) in the second study.

Finally, GA-Dimililer [32] is an approach that can be considered in between Greedy FS and Greedy BS concerning complexity reduction. This genetic search obtains medium size MCSs with an intermediate complexity (number of rules) reduction. Thus, in the third study, we will consider FRBCS-CM variants, providing a similar complexity, namely, FRBCS-CM 25% and FRBCS-CM with 50% (FRBCS-CM 50%) of nonzero values for a comparison.

Notice that FRBCS-CM allows the user to estimate the reduction of the complexity of the final MCS *a priori* by selecting the appropriate nonzero parameter value. This high flexibility, an *a priori* choice of how simple will the MCS obtained be, constitutes an advantage over the aforementioned approaches.

The three experiments developed are described as follows.

TABLE VI
WILCOXON SIGNED-RANK TEST FOR THE COMPARISON OF FRBCS-CM (10% AND 25%) AND GREEDY FS APPROACHES (MV, AVG, AND DT)

Comparison	p-value
FRBCS-CM 10% vs Greedy FS MV	=(0.1977)
FRBCS-CM 10% vs Greedy FS AVG	=(0.6142)
FRBCS-CM 10% vs Greedy FS DT	=(0.6813)
FRBCS-CM 25% vs Greedy FS MV	**+(0.0022)**
FRBCS-CM 25% vs Greedy FS AVG	**+(0.0036)**
FRBCS-CM 25% vs Greedy FS DT	**+(0.0012)**

1) Comparison of Fuzzy Rule-Based Classification System-Based Combination Method and Greedy Forward Selection Approaches: The results obtained by FRBCS-CM 10% and 25%, as well as Greedy FS approaches MV, AVG, and DT, are presented in Table V. From the viewpoint of this table, it can be noticed that FRBCS-CM 10% outperforms Greedy FS approaches AVG and DT in terms of overall average test error as well as for average test error for low-dimensional datasets, while being slightly inferior considering average test error for high-dimensional datasets by 0.0024 and 0.0004, respectively. Our approach obtains approximately 90 rules less on average in comparison with both Greedy FS approaches, which gives around 30% of additional complexity reduction.

Let us consider Greedy FS MV apart, since it is a special case. This approach outperforms FRBCS-CM 10% in terms of overall average test error and also average test error for both low- and high-dimensional datasets. However, the performance difference between both algorithms is very small, as it is equal to 0.002. Moreover, the number of rules obtained by Greedy FS MV is twice as big as the number of rules obtained by our approach.

Considering FRBCS-CM 25%, it outperforms all three Greedy FS approaches in terms of overall average test error

TABLE VII
COMPARISON OF FRBCS-CM AND GREEDY BS APPROACHES (MV, AVG, AND DT) IN TERMS OF ACCURACY AND COMPLEXITY

	FRBCS-CM 75%		FRBCS-CM 90%		Greedy BS MV		Greedy BS AVG		Greedy BS DT	
Dataset	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules	Test err.	# Rules
Low dim.:										
abalone	0.7470	2983.6	**0.7461**	3578.4	0.7484	2711.3	0.7524	3306.9	0.7511	3398.7
breast	0.0438	326.2	0.0432	391	0.0412	415.9	0.0386	426.6	**0.0372**	427.4
glass	**0.2692**	434.4	0.2710	528	0.2832	560.5	0.2720	576.4	0.2776	577.5
heart	0.1696	352.6	0.1696	421	0.1778	444.6	0.1770	455.7	**0.1674**	454.7
magic	0.1302	2969.9	0.1300	3475.8	0.1338	2247.8	0.1326	3203.6	**0.1298**	3319
pblocks	0.0268	967.8	**0.0261**	1182.2	0.0286	1259	0.0269	1288	0.0263	1297.3
phoneme	0.1256	1679.0	0.1264	2000	0.1291	1442.8	0.1271	2046	**0.1248**	2049.4
pima	0.2432	782.4	0.2424	946	0.2385	957	**0.2375**	1025	0.2414	1027.7
wine	0.0404	172.7	0.0393	208	0.0393	222.4	0.0371	226.9	**0.0360**	226.9
yeast	0.4034	1825.1	0.4013	2198.4	0.4011	1901.3	**0.3978**	2296.7	0.4018	2291.9
Avg. Low	0.2199	1249.4	0.2196	1492.8	0.2221	1216.3	0.2199	1485.2	**0.2193**	1507.1
High dim.:										
ionosphere	0.1430	279.8	0.1430	334	0.1476	353.3	0.1430	361.2	**0.1413**	360.6
optdigits1	0.0318	2678.8	0.0313	3227.2	0.0329	3398.5	0.0284	3513.8	**0.0279**	3513.1
pendigits	0.0140	3299.9	0.0138	3964.3	0.0156	4167.2	0.0129	4306.4	**0.0126**	4307.5
sat	0.0986	3128.1	0.1005	3762.8	0.1022	3575.2	**0.0967**	4006.8	0.0971	4055
segment	0.0316	876.9	**0.0302**	1051.4	0.0330	1100.5	0.0309	1151.3	0.0306	1151.4
sonar	0.2077	240.0	**0.2058**	288	0.2260	306.4	0.2183	312.1	0.2163	311.9
spambase	**0.0539**	1661.7	0.0546	2002.6	0.0579	2135.5	0.0554	2152.4	0.0549	2139.8
texture	0.0291	2175.0	0.0285	2610.9	0.0308	2759.8	**0.0268**	2852.9	0.0270	2852.8
vehicle	0.2664	1075.3	0.2674	1283	0.2723	1304.7	0.2641	1387.6	**0.2600**	1380
waveform	0.1489	2601.2	0.1479	3137.6	0.1498	3125.9	**0.1468**	3408.3	0.1472	3381.1
Avg. High	0.1025	1801.7	0.1023	2166.1	0.1068	2222.7	0.1023	2345.3	**0.1015**	2345.3
Avg. All	0.1612	1525.5	0.1609	1829.4	0.1644	1719.5	0.1611	1915.2	**0.1604**	1926.2

and average test error for low- and high-dimensional datasets. However, it obtains a slightly higher complexity, i.e., approximately 57 more rules on average than Greedy FS MV (around 13% of complexity increase), and roughly 200 more rules on average in comparison with Greedy FS AVG and DT (around 67% of complexity increase).

From these analyses, we may draw two conclusions. First, the novel FRBCS-CM proposed is competitive with the state-of-the-art fusion methods, when using Greedy FS. Second, our approach allows the user a higher degree of flexibility to define the desired tradeoff between accuracy and complexity by means of the specification of the nonzero parameter value. Choosing it *a priori* gives an estimation of how simplified will the final MCS be, which does not characterize the Greedy FS approaches.

Furthermore, Table VI presents the p-values of the statistical tests performed in order to check if the Wilcoxon test shows significant differences between FRBCS-CM 10% and 25% and the Greedy FS approaches in terms of accuracy. The results showing a significant difference between both algorithms are presented in bold font. From the viewpoint of this table, it can be clearly noticed that our approach is not significantly different for FRBCS-CM 10% and significantly different for FRBCS-CM 25% in comparison with the accuracy of the Greedy FS approaches (in favor of FRBCS-CM 25%), which confirm the conclusions drawn.

2) Comparison of Fuzzy Rule-Based Classification System-Based Combination Method and Greedy Backward Selection Approaches: The results obtained by FRBCS-CM 75% and 90%, as well as Greedy FS approaches MV, AVG, and DT, are presented in Table VII. From the viewpoint of this table, it can be noticed that FRBCS-CM 90% obtains similar results to Greedy BS AVG and DT, outperforming Greedy BS MV in terms of the overall average test error as well as the average test error for both low- and high-dimensional datasets. Considering the overall average test error, the ranking is Greedy BS with DT, FRBCS-CM, Greedy BS with AVG, and Greedy BS with MV. We should emphasize that the difference between the first three algorithms is negligible: 0.0005 between Greedy BS with DT and FRBCS-CM and 0.0002 between FRBCS-CM and Greedy BS with AVG.

TABLE VIII
WILCOXON SIGNED-RANK TEST FOR THE COMPARISON OF FRBCS-CM (75% AND 90%) AND GREEDY BS APPROACHES (MV, AVG, AND DT)

Comparison	p-value
FRBCS-CM 75% vs Greedy BS MV	**+(0.0117)**
FRBCS-CM 75% vs Greedy BS AVG	=(0.6291)
FRBCS-CM 75% vs Greedy BS DT	=(0.0825)
FRBCS-CM 90% vs Greedy BS MV	**+(0.0029)**
FRBCS-CM 90% vs Greedy BS AVG	=(0.5460)
FRBCS-CM 90% vs Greedy BS DT	=(0.1614)

Considering the number of rules, FRBCS-CM 90% obtains roughly 95 and 85 rules less, on average, than the competitive Greedy BS DT and AVG, respectively (around 5% and 4.5% of additional complexity reduction). On the opposite, FRBCS-CM 90% obtains roughly 110 more rules on average than Greedy BS MV (around 6% of complexity increase); however, Greedy BS MV obtains a lower accuracy.

To complete the analysis, let us consider FRBCS-CM 75%, which obtains a lower complexity, as well as a lower accuracy. We will compare it with Greedy BS MV, since FRBCS-CM 90% already obtains less rules than Greedy BS AVG and DT. FRBCS-CM 75% outperforms Greedy BS MV in terms of overall average test error and also average test error for both low- and high-dimensional datasets. Moreover, it obtains roughly 194 rules less than Greedy BS MV (around 13% of additional complexity reduction).

TABLE IX
COMPARISON OF FRBCS-CM AGAINST GA-DIMILILER IN TERMS OF ACCURACY AND COMPLEXITY

	FRBCS-CM 25%		FRBCS-CM 50%		GA-Dimililer	
Dataset	Tst err.	# Rules	Tst err.	# Rules	Tst err.	# Rules
Low dim.:						
abalone	0.7537	995.7	**0.7493**	1996.9	0.7494	2391.9
breast	0.0469	110.9	0.0452	217.0	**0.0409**	221.1
glass	0.2879	140.6	**0.2832**	289.9	0.3131	173.8
heart	0.1733	120.3	**0.1719**	235.3	0.1726	221.1
magic	0.1314	968.3	**0.1309**	1965.6	0.1336	2123.6
pblocks	**0.0265**	328.9	0.0271	628.1	0.0402	314.1
phoneme	**0.1252**	587.8	0.1261	1132.5	0.1301	996.9
pima	0.2484	260.7	0.2411	530.1	**0.2398**	530
wine	0.0382	57.9	**0.0303**	116.4	0.0348	71.2
yeast	0.4054	630.9	**0.3985**	1198.4	0.4116	902.4
Avg. Low	0.2237	420.2	**0.2204**	831.0	0.2266	794.6
High dim.:						
ionosphere	**0.1413**	95.4	0.1458	211.0	0.1464	190.3
optdigits1	**0.0327**	893.5	**0.0327**	1787.7	0.0721	661.5
pendigits	0.0152	1098.1	**0.0140**	2208.7	0.0160	1874.6
sat	0.1027	1046.9	**0.0997**	2107.2	0.1040	1431.9
segment	0.0319	290.9	**0.0304**	593.4	0.0345	414.2
sonar	0.2183	80.4	**0.2077**	162.0	0.2231	158.8
spambase	0.0559	557.2	**0.0555**	1115.5	0.0574	1026
texture	0.0312	716.6	**0.0304**	1458.8	0.0325	1240.4
vehicle	**0.2664**	380.4	0.2690	735.3	0.2721	425.7
waveform	**0.1490**	861.5	0.1503	1749.8	0.1532	828.9
Avg. High	0.1045	602.1	**0.1036**	1212.9	0.1111	825.2
Avg. All	0.1641	511.1	**0.1620**	1022.0	0.1689	809.9

The flexibility of our approach is again clearly appreciated, since it can define different accuracy–complexity tradeoffs, which is not allowed by Greedy BS approaches. Our approach also turned out to be competitive with the state-of-the-art fusion methods, when using Greedy BS, in terms of accuracy.

The latter conclusion is confirmed in Table VIII, which presents the p-values of the statistical tests performed in order to check if the Wilcoxon test shows significant differences between FRBCS-CM 75% and 90%, as well as Greedy BS approaches in terms of accuracy. The results showing a significant difference between both algorithms are presented in bold font. From the viewpoint of this table, it can be noticed that our approach is not significantly different for both nonzero values, when compared with Greedy BS AVG and DT, while being significantly different (in favor of the FRBCS-CM) when compared with Greedy BS MV.

We should again remind that the aim of this study is to propose the MCS combination methods, providing a good interpretability–accuracy tradeoff; thus, our approach has to be competitive in terms of accuracy (but not necessarily the best), while allowing some interpretability insights at the same time. We actually think that this goal has been obtained in view of the performed experiment.

3) Comparison of Fuzzy Rule-Based Classification System-Based Combination Method and a Hybrid Method Based on Genetic Algorithm Considering Classifier Selection and Classifier Fusion: The results obtained by FRBCS-CM 25% and 50% and GA-Dimililer are presented in Table IX. From the viewpoint of this table, it can be noticed that both FRBCS-CMs outperform GA-Dimililer, considering the overall average test error as well as the average test error for both low- and high-dimensional datasets. GA-Dimililer turns out to be an inferior approach even in comparison with FRBCS-CM 25%, which obtains almost 40% rules less on average. Thus, we may conclude that the novel FRBCS-CM proposed is competitive with the state-of-the-art hybrid method based on GA, considering classifier selection and classifier fusion, while also being able to obtain simpler MCSs.

TABLE X
WILCOXON SIGNED-RANK TEST FOR THE COMPARISON OF FRBCS-CM (25% AND 50%) AGAINST GA-DIMILILER

Comparison	p-value
FRBCS-CM 25% vs GA-Dimililer	**+(0.0340)**
FRBCS-CM 50% vs GA-Dimililer	**+(0.0007)**

The statistical tests performed clearly show that fact. The FRBCS-CM with both nonzero values is statistically different from GA-Dimiliter (see Table X), with differences being in favor of our approach.

D. Interpretability Study on Fuzzy Rule-Based Classification System-Based Combination Method

The proposed fuzzy linguistic combination method provides a good degree of interpretability to the MCS, making the combination method operation mode more transparent for the user. Furthermore, when combined with a fuzzy MCS, the whole system takes a pure hierarchical structure based on fuzzy classification rules structure (in the sense that the weak learners constitute individual FRBCSs, becoming the input to the FRBCS-based combination method). The type of rules with a class and a certainty degree in the consequent used in the FRBCS-CM allows the user to get an understandable insight into the MCS, thus allowing interpretability of such a complicated system to some extent. The global framework of our proposal was already presented in Fig. 1.

To illustrate the interpretability capabilities of FRBCS-CM, we will show how it works on two of the datasets presented in Table I (one from each group, low and high dimensionality). The fuzzy rule base obtained with FRBCS-CM 10% on the wine dataset is presented in Fig. 4. For the sake of comprehensibility, the sparse matrix introduced in Section III was transposed (on the left side), and its values were sorted according to the class label c, exposing which classifier fuzzy rules are considered to obtain the final decision k with a given weight w. The corresponding FURIA fuzzy rules (on the right side), presented in Section II-C, were simplified (notice that we do not show the exact values of the attributes to keep a simple structural representation) as well as sorted according to the class label. Fuzzy rule base obtained by the FRBCS-CM (in the middle), with a linguistic terms in the antecedent parts of the rules, composes a human-interpretable structure, making this combination method operation mode more transparent for the user. The FRBCS-CM transition from the second level of the fuzzy MCS, that is the combination method, to the first level, i.e., the FURIA-based fuzzy component classifiers, is clearly shown. The FRBCS-CM provides the user with an insight of which rules of component base classifier are selected for a given class label.

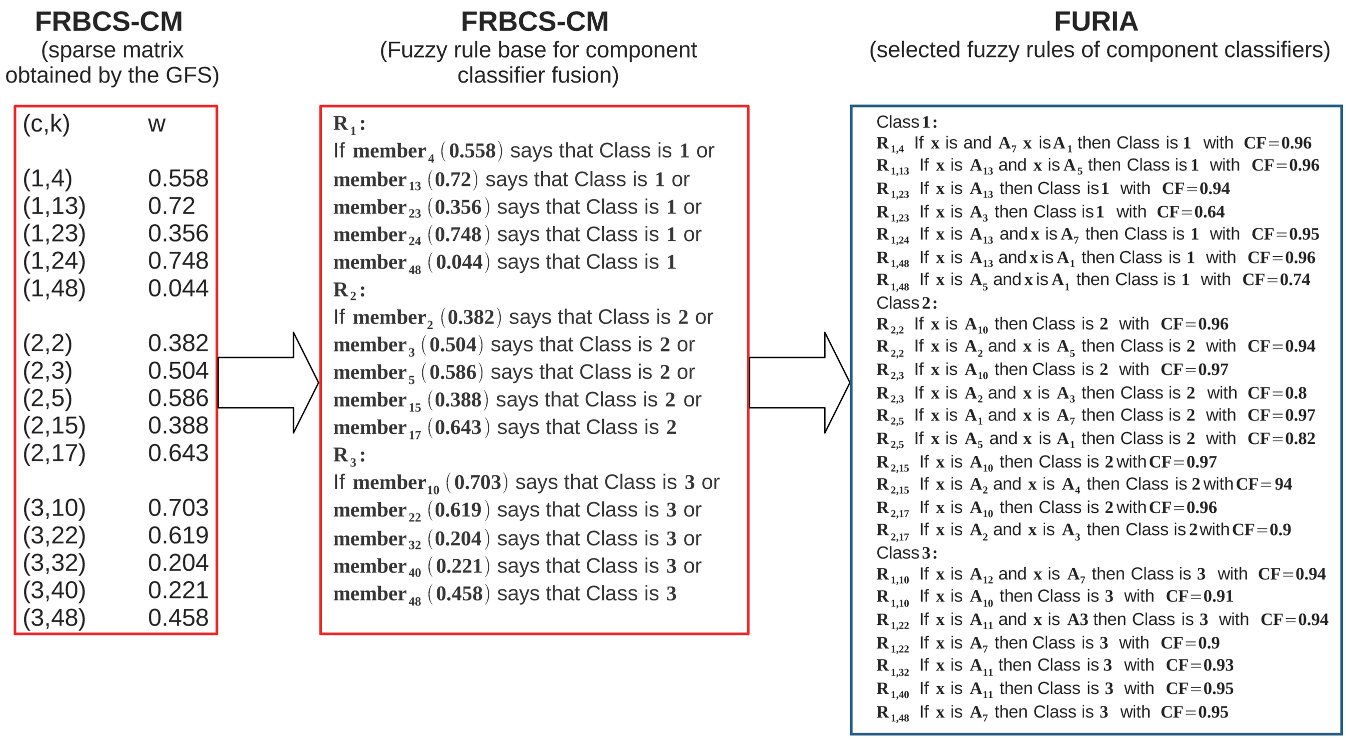


Fig. 4. Example showing how FRBCS-CM selects and combines fuzzy rules of the selected component FURIA-based fuzzy classifiers. The wine dataset was used for illustration with FRBCS-CM considering 10% of the nonzero values.

The magic dataset was selected as the representative from the high-dimensional group. Fig. 5 presents the fuzzy rule base obtained with FRBCS-CM 25%. The corresponding FURIA fuzzy rules are not shown in this case due to their large number. Nevertheless, the transparency of the fuzzy linguistic combination method can be clearly seen again. The user is provided with fuzzy rule base, indicating which rules of which component base classifier are selected for a given class label.

Thus, we may conclude that the proposed approach is capable of providing interpretability to the MCS to some extent. Moreover, up to our knowledge, there has not been any previous proposal of an MCS combination method that tried to deal with this matter.

E. Comparison of the Runtime of the Fuzzy Multiclassification systems, Fuzzy Rule-Based Classification System-Based Combination Method, and the Other Multiclassification System Combination Methods

The runtime values in seconds for the fuzzy MCSs, FRBCS-CM, and the other MCS combination methods are presented in Table XI. From the viewpoint of this table, the following conclusions can be drawn.

1) Considering fuzzy MCSs, the runtime varies from less than 1 s (0.30 s for wine dataset) to several hours (9.05 h for texture dataset), obtaining roughly 2.45 h considering overall average runtime.

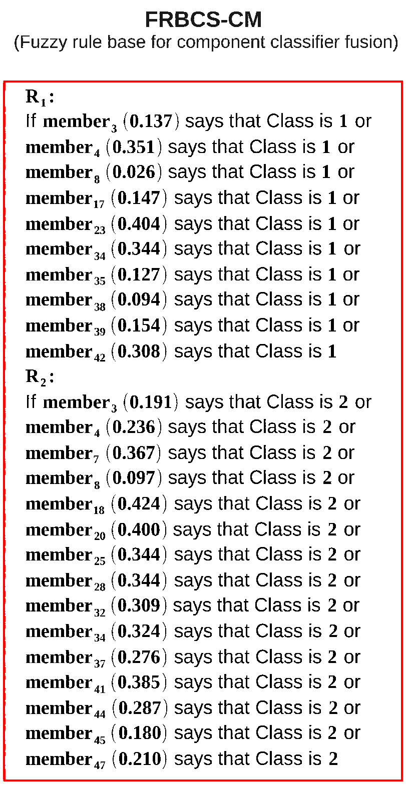


Fig. 5. Example showing the FRBCS-CM fuzzy rule base. The magic dataset was used for illustration with FRBCS-CM considering 25% of the nonzero values.

TABLE XI
AVERAGE RUN TIMES OF THE FUZZY MCSS, FRBCS-CM, AND THE OTHER MCS COMBINATION METHODS (IN SECONDS)

	fuzzy	FRBCS-CM					Greedy FS			Greedy BS			GA
Dataset	MCSs	10%	25%	50%	75%	90%	MV	AVG	DT	MV	AVG	DT	Dimil.
Low dim.:													
abalone	133.77	706.91	734.69	773.09	754.10	750.95	6584.02	14.01	657.46	16419.09	105.43	10344.75	5363.25
breast	0.72	25.25	25.24	25.90	25.80	26.99	400.35	0.10	0.79	897.88	0.66	4.07	46.48
glass	0.60	10.98	11.01	10.99	11.06	11.19	171.76	0.22	2.51	290.40	1.28	18.93	54.28
heart	0.58	9.76	9.80	9.98	9.84	9.99	171.86	0.09	0.54	350.64	0.39	2.39	25.69
magic	2946.21	1496.35	1554.24	1406.49	1397.34	1405.72	14964.89	1.06	26.02	36107.43	29.45	209.59	1473.34
pblocks	34.98	509.17	508.69	513.96	507.70	514.42	4747.74	0.72	16.68	7272.40	8.88	83.61	724.40
phoneme	2627.49	477.63	474.02	475.04	477.47	477.17	4761.22	0.50	8.76	21642.29	6.34	41.54	420.61
pima	156.91	28.59	28.51	29.04	28.51	28.63	978.39	0.52	3.74	1416.70	0.76	4.51	66.82
wine	0.30	3.20	3.22	3.36	3.22	3.23	48.65	0.17	0.77	210.61	0.48	3.68	21.81
yeast	22.76	111.98	110.81	111.98	112.24	110.68	1738.36	1.10	30.00	4614.72	9.77	199.10	488.58
Avg. Low	592.43	337.98	346.02	335.98	332.73	333.90	3456.72	1.85	74.73	8922.22	16.34	1091.22	868.53
High dim.:													
ionosphere	1.07	5.94	5.92	5.97	5.92	5.94	174.93	0.08	0.63	428.27	0.39	2.57	27.74
optdigits	298.72	439.18	470.02	437.75	406.36	428.26	5629.17	2.70	59.09	3377.23	6.80	100.52	1933.24
pendigits	372.60	1185.77	1117.63	1185.79	1162.23	1168.96	12547.1	3.36	100.49	14470.00	24.32	990.22	5613.07
sat	2952.88	540.87	545.92	536.25	531.76	566.40	9396.63	2.80	60.59	16069.53	17.52	290.59	1948.69
segment	964.85	171.47	171.79	187.31	184.22	176.76	1974.24	0.63	11.05	3394.29	4.93	64.43	763.54
sonar	41.00	7.46	7.42	7.46	7.54	7.53	111.60	0.09	0.49	249.64	0.36	2.07	22.94
spambase	1700.49	303.14	298.22	301.73	280.43	182.85	3990.22	0.52	8.15	5784.29	3.46	22.76	356.05
texture	3259.79	592.81	580.82	559.64	594.44	572.79	5790.39	2.21	71.74	7375.31	15.10	617.22	2407.66
vehicle	164.52	29.86	29.80	29.81	29.50	29.27	978.80	0.42	4.30	1471.12	1.44	15.94	180.05
waveform	2040.56	370.75	365.52	373.35	374.13	372.27	7806.02	1.30	24.09	9734.29	4.22	36.95	802.17
Avg. High	1179.65	364.72	359.31	362.51	357.65	351.10	4839.91	1.41	34.06	6235.40	7.85	214.33	1405.51
Avg. All	886.04	351.35	352.67	349.24	345.19	342.50	4148.32	1.63	54.39	7578.81	12.10	652.77	1137.02

2) The proposed FRBCS-CM is placed at the fourth place after Greedy FS AVG, Greedy BS AVG, and Greedy FS DT, while it is faster than Greedy BS DT, GA-Dimililer, Greedy FS MV, and Greedy BS MV, respectively.
3) The proposed FRBCS-CM obtains similar runtime values (roughly 350 s considering overall average runtime), regardless of the nonzero percentage value specified.
4) For both Greedy FS and Greedy BS, the highest runtime values are obtained for MV (due to its implementation), while AVG is capable of learning within the smallest runtime.
5) Both Greedy FS and Greedy BS approaches obtain so small runtime values due to the characteristics of AVG. It is a simple mathematical function, which does not require any memory use (as in the case of DT). Each iteration of the Greedy algorithm is simply based on the calculation of the AVG value and the comparison with the best value obtained so far.
6) GA-Dimililer obtains roughly 3.15 h, considering overall average runtime. This value is approximately three times higher than that of the proposed FRBCS-CM, which is also based on the use of a GA.

V. CONCLUDING REMARKS

We have proposed a novel MCS fuzzy linguistic combination method based on the use of an FRBCS automatically derived by means of a GA. The new fuzzy linguistic combination method shows very interesting characteristics, especially its transparency and its capability to jointly perform classifier fusion and selection. In addition, when combined with a fuzzy classifier ensemble, the overall system shows a hierarchical structure (called stacking in the literature). Thus, this means that FRBCSs can deal with high-dimensional problems that avoid the curse of dimensionality, allowing the user to select an appropriate accuracy–complexity tradeoff.

We carried out exhaustive experiments using 20 datasets from the UCI repository with different dimensionality. A comparison with some state-of-the-art classifier fusion methods, such as MV, AVG, and DT combined with Greedy FS and Greedy BS, as well as with the hybrid method based on a GA considering both classifier selection and classifier fusion [32], led us to the conclusion that our approach is competitive in terms of both accuracy and complexity. Furthermore, we were able to show that this approach allows us to get some insights into the MCS fusion method, which is not a usual case in the field of MCSs, thus leading to a good interpretability–accuracy tradeoff in fuzzy MCSs.

APPENDIX
BRIEF DESCRIPTION OF THE FUZZY UNORDERED RULES INDUCTION ALGORITHM

To make this paper self-contained, we introduce the main features of the algorithm used as a base classifier. Fuzzy unordered rules induction algorithm (FURIA) [51], [52] is a novel FRBCS, extending the state-of-the-art rule learning algorithm called RIPPER [67]. It maintains its advantages such as simple and comprehensible fuzzy rule base or an internal feature selection algorithm, while introducing some new features. FURIA provides three different extensions of RIPPER.

1) It takes an advantage of fuzzy rules instead of crisp ones. Fuzzy rules of FURIA are composed of a class C_j and a certainty degree CD_j in the consequent. The final form of a rule is the following:

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

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

$$CD_j = \frac{2\frac{D_T^{C_j}}{D_T} + \sum_{x \in D_T^{C_j}} \mu_r^{C_j}(x)}{2 + \sum_{x \in D_T} \mu_r^{C_j}(x)} \quad (5)$$

where D_T and $D_T^{C_j}$ stands for the training set and a subset of the training set belonging to the class C_j, respectively. 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. Hence, the fuzzy reasoning method used is the so-called voting-based method [30], [68].

2) It uses unordered rule sets instead of rule lists. This change omits a bias caused by the default class rule, which is applied whenever there is an uncovered example detected.
3) It proposes a novel rule stretching method in order to manage uncovered examples. The unordered rule set introduces one crucial drawback: There might appear a case when a given example is not covered. Then, to deal with such a situation, one rule is generalized by removing its antecedents. The information measure is proposed to verify which rule to "stretch."

See [51] for a full description of FURIA.

REFERENCES

[1] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. New York: Wiley, 2004.
[2] T. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[3] D. Optiz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, 1999.
[4] J. Canul-Reich, L. Shoemaker, and L. O. Hall, "Ensembles of fuzzy classifiers," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, London, U.K., Jul. 2007, pp. 1–6.
[5] W. Pedrycz and K. C. Kwak, "Boosting of granular models," *Fuzzy Sets Syst.*, vol. 157, no. 22, pp. 2934–2953, 2006.
[6] 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 *Proc. 8th Int. Conf. Hybrid Intell. Syst.*, Barcelona, Spain, 2008, pp. 549–554.
[7] 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 Intell. Syst.*, vol. 7, no. 1, pp. 45–64, 2010.
[8] K. Trawiński, O. Cordón, and A. Quirin, "On designing fuzzy rule-based multiclassification systems by combining FURIA with bagging and feature selection," *Int. J. Uncert. Fuzz. Knowl.-Based Syst.*, vol. 19, no. 4, pp. 589–633, 2011.
[9] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, May 1979.
[10] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[11] R. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[12] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Diversity in search strategies for ensemble feature selection," *Inf. Fusion*, vol. 6, no. 1, pp. 83–98, 2005.
[13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[14] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their application to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 3, pp. 418–435, May/Jun. 1992.
[15] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, Apr. 1997.
[16] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognit.*, vol. 34, no. 9, pp. 1879–1881, 2001.
[17] D. Partridge and W. B. Yates, "Engineering multiversion neural-net systems," *Neural Comput.*, vol. 8, no. 4, pp. 869–893, 1996.
[18] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern.*, vol. 27, no. 5, pp. 553–568, Sep. 1997.
[19] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–238, Mar. 1998.
[20] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis, "Soft combination of neural classifiers: A comparative study," *Pattern Recognit. Lett.*, vol. 20, no. 4, pp. 429–444, 1999.
[21] L. Kuncheva, ""Fuzzy" versus "nonfuzzy" in combining classifiers designed by boosting," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 729–741, Dec. 2003.
[22] D. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[23] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling With Linguistic Information Granules*. New York: Springer-Verlag, 2005.
[24] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific, 2001.
[25] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5–31, 2004.
[26] F. Herrera, "Genetic fuzzy systems: Taxonomy, current research trends and prospects," *Evol. Intell.*, vol. 1, pp. 27–46, 2008.
[27] O. Cordón, "A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *Int. J. Approx. Reason.*, vol. 52, no. 6, pp. 894–913, 2011.
[28] R. Alcalá, M. J. Gacto, and F. Herrera, "A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 666–681, Aug. 2011.
[29] J. Alcalá-Fdez, R. Alcalá, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 857–872, Oct. 2011.
[30] O. Cordón, M. J. del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *Int. J. Approx. Reason.*, vol. 20, pp. 21–45, 1999.
[31] J. Casillas, F. Herrera, R. Pérez, M. J. del Jesus, and P. Villar, "Special issue on genetic fuzzy systems and the interpretability-accuracy trade-off," *Int. J. Approx. Reason.*, vol. 44, no. 1, pp. 4–31, Jan. 2007.
[32] N. Dimililer, E. Varoglu, and H. Altincay, "Classifier subset selection for biomedical named entity recognition," *Appl. Intell.*, vol. 31, pp. 267–282, 2009.
[33] R. Battiti and A. M. Colla, "Democracy in neural nets: Voting schemes for classification," *Neural Netw.*, vol. 7, no. 4, pp. 691–707, 1994.
[34] L. Kuncheva, "Combining classifiers: Soft computing solutions," in *Pattern Recognition. From Classical to Modern Approaches*, S. Pal and A. Pal, Eds.. Singapore: World Scientific, 2001, ch. 15, pp. 427–452.
[35] M. Kim, S. Min, and I. Han, "An evolutionary approach to the combination of multiple classifiers to predict a stock price index," *Exp. Syst. Appl.*, vol. 31, pp. 241–247, 2006.
[36] X. Zhu, X. Wu, and Y. Yang, "Dynamic selection for effective mining from noisy data streams," in *Proc. 4th IEEE Int. Conf. Data Mining*, Brighton, U.K., Nov. 2004, pp. 305–312.
[37] E. M. Dos Santos, R. Sabourin, and P. Maupin, "A dynamic overproduce-and-choose strategy for the selection of classifier ensembles," *Pattern Recognit.*, vol. 41, no. 10, pp. 2993–3009, 2008.
[38] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, 2002.
[39] A. J. C. Sharkey and N. E. Sharkey, "The *test and select* approach to ensemble combination," in *Proc. Int. Workshop Multiclassifier Syst.*, Cagliari, Italy, 2000, pp. 30–44.
[40] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[41] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Single and multi-objective genetic algorithms for the selection of ensemble of classifiers," in *Proc. Int. Joint Conf. Neur. Netw.*, Vancouver, BC, Canada, 2006, pp. 3070–3077.
[42] 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. Comput. Intell. Syst.*, vol. 5, no. 2, pp. 231–253, 2012.
[43] B. Gabrys and D. Ruta, "Genetic algorithms in classifier fusion," *Appl. Soft Comput.*, vol. 6, no. 4, pp. 337–347, 2006.
[44] M. C. C. Abreu and A. M. P. Canuto, "An experimental study on the importance of the choice of the ensemble members in fuzzy combination methods," in *Proc. 7th Int. Conf. Intell. Syst. Des. Appl.*, Rio de Janeiro, Brazil, 2003, pp. 723–728.
[45] M. Grabisch, "Fuzzy integral in multicriteria decision making," *Fuzzy Sets Syst.*, vol. 89, pp. 279–298, 1995.
[46] D. Filev and R. R. Yager, "A generalized defuzzification method under BAD distributions," *Int. J. Intell. Syst.*, vol. 6, pp. 687–697, 1989.
[47] H. Zimmermann and P. Zysno, "Latent connectives in human decision making," *Fuzzy Sets Syst.*, vol. 4, no. 1, pp. 37–51, 1980.
[48] L. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognit.*, vol. 34, no. 2, pp. 299–314, 2001.
[49] P. Bulacio, S. Guillaume, E. Tapia, and L. Magdalena, "A selection approach for scalable fuzzy integral combination," *Inf. Fusion*, vol. 11, no. 2, pp. 208–213, 2010.
[50] Y. Lu and F. Yamaoka, "Fuzzy integration of classification results," *Pattern Recognit.*, vol. 30, no. 11, pp. 1877–1891, 1997.
[51] J. C. Hühn and E. Hüllermeier, "FURIA: An algorithm for unordered fuzzy rule induction," *Data Mining Knowl. Discovery*, vol. 19, no. 3, pp. 293–319, 2009.
[52] J. C. Hühn and E. Hüllermeier, "An analysis of the FURIA algorithm for fuzzy rule induction," in *Proc. Adv. Mach. Learn. I*, 2010, pp. 321–344.
[53] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.
[54] R. Senge and E. Hüllermeier, "Top-down induction of fuzzy pattern trees," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 241–252, Apr. 2011.
[55] Y.-C. Chen, N. R. Pal, and I.-F. Chung, "An integrated mechanism for feature selection and fuzzy rule extraction for classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 683–698, Aug. 2012.
[56] V. Torra, "A review of the construction of hierarchical fuzzy systems," *Int. J. Intell. Syst.*, vol. 17, no. 5, pp. 531–543, 2002.
[57] A. E. Gegov and P. M. Frank, "Hierarchical fuzzy control of multivariable systems," *Fuzzy Sets Syst.*, vol. 72, pp. 299–310, 1995.
[58] R. R. Yager, "On the construction of hierarchical fuzzy systems model," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 1, pp. 55–66, Feb. 1998.
[59] O. Cordón, F. Herrera, and I. Zwir, "A hierarchical knowledge-based environment for linguistic modeling: Models and iterative methodology," *Fuzzy Sets Syst.*, vol. 138, no. 2, pp. 307–341, 2003.
[60] J. J. Rubio, P. Angelov, and J. Pacheco, "An uniformly stable backpropagation algorithm to train a feedforward neural network," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 356–366, Mar. 2011.
[61] E. Lughofer, J.-L. Bouchot, and A. Shaker, "On-line elimination of local redundancies in evolving fuzzy systems," *Evol. Syst.*, vol. 2, pp. 165–187, 2011.
[62] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for the behaviour analysis," *Artif. Intell. Rev.*, vol. 12, pp. 265–319, 1998.
[63] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1996.
[64] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computin*, (ser. Natural Computing Series), 2nd ed. New York: Springer-Verlag, 2007.
[65] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.
[66] E. Lehmann, *Nonparametric Statistical Methods Based on Ranks*. New York: McGraw-Hill, 1975.
[67] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1995, pp. 115–123.
[68] H. Ishibuchi, T. Nakashima, and T. Morisawa, "Voting in fuzzy rule-based systems for pattern classification problems," *Fuzzy Sets Syst.*, vol. 103, no. 2, pp. 223–238, 1999.

Krzysztof Trawiński was born in Wrocław, Poland, in 1985. He received the M.Sc. degree in computer science from the Wrocław University of Technology in 2008, focusing on building a system for real estate appraisal problem. It mainly included the field of genetic fuzzy systems. He also received the Master degree in soft computing and intelligent data analysis from the European Centre for Soft Computing, Mieres, Spain, and the University of Oviedo, Gijón, Spain, in July 2010. His Master project "Designing Fuzzy Multiclassifier Systems by Combining FURIA with Bagging and Feature Selection" was mainly focused on designing a methodology in which a bagging approach, together with a feature selection technique, is used to train a fuzzy unordered rules induction algorithm in order to obtain a fuzzy multiclassifier system. From 2008 to 2009, he was an Intern with the Applications of Fuzzy Logic and Evolutionary Algorithms (EAs) Unit, European Centre for Soft Computing, as a part of the Leonardo da Vinci programme.

He took part in the Socrates/Erasmus (2006–2007) and GFF (2007–2008) programs, both in Dresden, Germany. Since June 2009, he has been a Research Assistant with the European Centre for Soft Computing. He has contributed to more than ten papers published in international conferences and journals. His Ph.D. dissertation is related to fuzzy classifier ensembles dealing with high-dimensional systems by considering data resampling, feature selection, and multiobjective EAs for classifier selection to get an appropriate accuracy–complexity tradeoff. His main research interests include multiclassifiers, fuzzy multiclassifiers, fuzzy rule-based systems, genetic fuzzy systems, and evolving fuzzy systems.

Oscar Cordón (SM'11) received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Spain, in 1994 and 1997, respectively.

He is currently a Full Professor with the University of Granada (UGR), Granada, Spain. He was the Founder and Leader of that University's Virtual Learning Center between 2001 and 2005. From 2006 to 2011, he was one of the founding researchers of the European Centre for Soft Computing, Mieres, Spain, in his role of Principal Researcher. He is still contracted as a Scientific Consultant at the Centre. He has been, for more than 15 years, an internationally recognized contributor to R&D Programs in fuzzy systems, evolutionary algorithms, and ant colony optimization. He has published around 280 peer-reviewed scientific publications, including a research book on genetic fuzzy systems and 68 JCR-SCI-indexed journal papers, advised 13 Ph.D. dissertations, participated in 34 research projects and contracts (coordinating 20 of them), co-edited eight special issues in international journals and three research books, and given 12 plenary talks at national and international conferences. By November 2012, his publications had received 2003 citations, carrying an h index of 23, and he was included in the 1% of most-cited researchers in the world (source: Thomson Reuters's Web of Knowledge. 6472 citations and h index of 38 in ScholarGoogle). He also has an approved international patent on an intelligent system for forensic identification that is being commercialized in Mexico.

Prof. Cordón is an Associate Editor of 11 international journals and was recognized as the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Associate Editor in 2008. He is a reviewer for more than 30 international journals. Since 2004, he has taken many different representative positions with EUSFLAT and the IEEE Computational Intelligence Society (CIS). He is currently a member of the IEEE CIS AdCom (term 2010–2012). He received the UGR Young Researcher Career Award in 2004. In July 2010, he received the IEEE CIS Outstanding Early Career Award in its 2011 edition: the first such award conferred. In 2011, his developments on soft computing methods in forensic identification were recognized with the International Fuzzy Systems Association (IFSA) Award for Outstanding Applications of Fuzzy Technology and with the granting of the EUSFLAT Best Ph.D. Thesis Award to Dr. Ibañez's Ph.D. dissertation, which was co-advised by Dr. Cordón.

Luciano Sánchez (M'07) received the M.Sc. and Ph.D. degrees in electronic and control engineering from the University of Oviedo, Gijón, Spain, in 1991 and 1994, respectively.

He is currently a Professor with the Department of Computer Science, University of Oviedo, and the Head of the research group "Metrology and Models" and Founding Partner of the spin-off of the research group IDALIA S.L. He has led four research projects supported by the Spanish government and 13 privately funded research contracts. He was an External Consultant with Electromateriales KLK during 1991–1995, with Saint-Gobain Cristaleria during 2000–2002, with HC Energia during 2003–2005, with IT-VASA during 2005–2008, and with Indra Sistemas S.A. during 2010–2011, and a Visiting Scholar with the University of California at Berkeley in 1995 and General Electric Global Research in 1996. His research interests include the theoretical study of algorithms for mathematical modeling and intelligent data analysis and the application of these techniques to practical problems of industrial modeling, signal processing, and dimensional metrology, with special interest in the study of low-quality data and fuzzy metainformation. He is a member of the Editorial Board of three international journals, part of the steering committee of the Spanish Metaheuristicas, Algoritmos Evolutivos y Bioinspirados series of conferences, and more than 60 Spanish and international conferences. He is an author of more than 40 international journal papers and more than 100 contributions to conferences and book chapters.

Arnaud Quirin received the M.Sc. degree in 2002 and the Ph.D. degree in computer science in 2005, both from the University Louis Pasteur of Strasbourg, Schiltigheim, France.

He was a Teaching and Researcher Assistant with the University Louis Pasteur of Strasbourg until September 2006. From 2006 to 2012, he was with the European Centre for Soft Computing, Mieres, Spain. He is currently a Data Mining Researcher with Gradiant, Vigo, Spain. He has contributed to more than 30 peer-reviewed scientific publications, including three book chapters and eight JCR-SCI-indexed journal papers. In addition, he is a Reviewer for six international journals. He was involved in several national project, private contracts, and one European project related to the application of evolutionary algorithms to image classification. He currently participates in several R&D programs in the fields of evolutionary algorithms, multiobjective graph-based mining, fuzzy multiclassifier systems, and genetic fuzzy algorithms.

4 Multiobjective Genetic Classifier Selection for Random Oracles Fuzzy Rule-Based Multiclassifiers: How Beneficial is the Additional Diversity?

This publication was also published in the top rank journal from the Artificial Intelligence category:

- K. Trawiński, O. Cordón, L. Sánchez, and A. Quirin. Multiobjective Genetic Classifier Selection for Random Oracles Fuzzy Rule-Based Multiclassifiers: How Beneficial is the Additional Diversity?, Knowledge-based Systems, In press, 2013. DOI: 10.1016/j.knosys.2013.08.006.
  - State: In press.
  - Impact Factor (JCR 2012): 4.104.
  - Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 6/115. Q1.

ARTICLE IN PRESS

Knowledge-Based Systems xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?

Krzysztof Trawiński [a,*], Oscar Cordón [a,b], Arnaud Quirin [c], Luciano Sánchez [d]

[a] European Centre for Soft Computing, Edificio de Investigación, Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Asturias, Spain
[b] Department of Computer Science and Artificial Intelligence (DECSAI) and Research Center on Information and communication Technologies (CITIC-UGR), University of Granada, 18071 Granada, Spain
[c] Galician Research and Development Center in Advanced Telecommunications (GRADIANT), Communications Area, Edif. CITEXVI, Local 14, 36310 Vigo, Spain
[d] Computer Science Department, University of Oviedo, Campus de Viesques, 33203 Gijón, Asturias, Spain

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Fuzzy rule-based classifier ensembles
Random oracles
Bagging
Genetic classifier selection
Diversity measures
Evolutionary multiobjective optimization
NSGA-II
High complexity datasets

ABSTRACT

Recently we proposed the use of the Random Linear Oracles classical classifier ensemble (CE) design methodology in a fuzzy environment. It derived fuzzy rule-based CEs obtaining an outstanding performance. Random Oracles introduce an additional diversity into the base classifiers improving the accuracy of the entire CE. Meanwhile, the overproduce-and-choose strategy leads to a good accuracy-complexity trade-off. It is based on the generation of a large number of component classifiers and a subsequent selection of the best cooperating subset of them. The current contribution has a twofold aim: (1) Introduce a new Random Oracles approach into the fuzzy rule-based CEs design; (2) Incorporate an evolutionary multi-objective overproduce-and-choose strategy to our approach analyzing the influence of this additional diversity in the final CE performance (focusing on the accuracy). To do so, firstly, we incorporate the two Random Oracle variants into the fuzzy rule-based CE framework. Then, we use NSGA-II to provide a specific component classifier selection driven by three different criteria. Exhaustive experiments are carried out over 29 UCI and KEEL datasets with high complexity (considering both the number of attributes as well as the number of examples) showing the good performance of the proposed approach.

1. Introduction

Classifier ensembles (CEs), also called multiclassifiers, are well-recognized tools in the machine learning community and more recently in the soft computing community. They are able not only to outperform a single classifier but also to deal with complex and high dimensional classification problems [1].

In a preceding contribution [2], we incorporated Random Linear Oracles (RLOs) [3], a classical CE design methodology, into a previously proposed CE framework [4] to derive fuzzy rule-based classifier ensembles (FRBCEs). Thanks to the additional diversity introduced by RLOs into the robust FURIA-based fuzzy classifiers [5,6], the obtained FRBCEs were able to achieve an outstanding performance in terms of accuracy, outperforming RLO combined with the classical base classifiers.

Nevertheless, the performance of FRBCEs can still be improved. It has been theoretically and empirically shown that smaller ensembles can outperform larger ones [7–9]. Thus, selecting a subset of classifiers is a natural way to follow. In our previous contributions, we used the well known *overproduce-and-choose strategy* [10] (OCS) to reduce the CE dimensionality, while improving its accuracy. OCS is a classifier selection method based on the generation of a large number of component classifiers and a subsequent selection of the best cooperating subset of them.

Therefore, OCS helps to obtain a good accuracy-complexity trade-off in the CE design as well as in many cases it also improves the accuracy of the final CE. In fact, these characteristics were exhibited in [11] for FRBCEs using an OCS strategy based on NSGA-II [12]. NSGA-II, which is a state-of-the-art evolutionary multi-objective (EMO) algorithm [13], generated a set of CE designs with different accuracy-complexity trade-offs in a single run.

In this contribution, we introduce two novel aspects to our FRBCE design methodology in [2] in order to improve the CE accuracy, while reducing its complexity:

* Corresponding author. Tel.: +34 985456545.
E-mail addresses: krzysztof.trawinski@softcomputing.es (K. Trawiński), oscar.cordon@softcomputing.es, ocordon@decsai.ugr.es (O. Cordón), aquirin@gradiant.org (A. Quirin), luciano@uniovi.es (L. Sánchez).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.08.006

1. To keep a high diversity in the set of classifiers as well as high performance, we incorporate a new Random Oracle (RO) approach, namely the Random Spherical Oracle (RSO) [14], into the FRBCE framework. Opposite to RLO, RSO uses an oracle based on a random hypersphere to divide the feature space into two regions in order to feed two subclassifiers, which both compose the final RSO. We expect to improve the performance of the FRBCEs by combining the RSO randomness and its oracle shapes with the "soft boundaries" provided by the FURIA-based component classifier.
2. To reduce the complexity, we design a specific EMO-based OCS strategy for RO-based FRBCEs from our previous proposal in [11]. Since RO is composed of two base classifiers, this approach offers a tremendous advantage over bagging FURIA-based component classifiers because each classifier can be independently selected within each pair component. A higher degree of freedom is achieved during the selection procedure, while still having the potential of drastically reducing the complexity.

On the one hand, we aim to obtain a good accuracy-complexity trade-off when dealing with high complexity datasets. While the main goal in the design of CEs is to obtain an accurate system, the complexity is an interesting secondary objective allowing us to obtain simpler and quicker CEs. On the other hand, we aim to analyze whether the additional diversity induced by ROs is beneficial for the EMO OCS-based FRBCEs. That is, our goal is to check if the OCS-based selection leads to more accurate results when applied on RO-based FURIA fuzzy CEs than on bagging fuzzy CEs thanks to the additional freedom degrees resulting from the RO design. For that purpose, we use a novel NSGA-II design with a three-objective fitness function including an advanced accuracy measure as well as complexity and diversity indices for the component classifier selection. Specifically, we propose a special binary coding for NSGA-II in order to take advantage of the additional degrees of freedom offered by the RO base classifiers, and test two different mutation operator settings to look for the best performance.

To perform the experimental analysis, we carry out exhaustive experiments on 29 high complexity datasets from the UCI machine learning [15] and the KEEL dataset [16] repositories.

This paper is set up as follows. In the next section, the preliminaries required for a good understanding of our work are reviewed. Section 3 presents RLOs, RSOs, both RLO- and RSO-based FRBCEs, and a set of experiments focused on the comparison of different RO-based strategies for the combination of the component classifiers. Then, Section 4 introduces our NSGA-II proposal for RSO component fuzzy classifier selection incorporating a three-objective fitness function and the analysis of the experiments performed. Finally, Section 5 concludes this contribution with some future research lines.

2. Preliminaries

This section explores the current literature related to the generation of a FRBCE. The techniques used to generate CEs and fuzzy CEs are described in Sections 2.1 and 2.2, respectively. Some ways to reduce the size of the ensembles are described in Section 2.3. The use of genetic algorithms (GAs) within the OCS strategy is explored in Section 2.4. Finally, we briefly introduce evolutionary fuzzy systems in Section 2.5.

2.1. Classifier ensembles design methodologies

A CE 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 [1]. 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 [17,18], although CEs can be used with any other type of classifiers (neural networks are also very extended, see for example [19]).

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 [20] 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 induce the individual classifier diversity using some ways different from resampling [21]. Feature selection plays a key role in many of them where each classifier is derived by considering a different subset of the original features [22,23]. *Random subspace* [24], 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 a combination of the two groups, such as *random forests* [25] and more recently *rotation forest* [26] and *fuzzy random forest* [27].

The interested reader is referred to [18,19] for two surveys for the case of decision tree (both) and neural network ensembles (the latter), including exhaustive experimental studies.

2.2. Related work on fuzzy classifier ensembles

Focusing on fuzzy CEs, only a few contributions for bagging fuzzy classifiers have been proposed considering fuzzy neural networks (together with feature selection) [28], neuro-fuzzy systems [29], and fuzzy decision trees [27,30] as component classifier structures.

Especially worth mentioning is the contribution of Bonissone et al. [27]. This approach hybridizes Breiman's idea of random forests [25] with fuzzy decision trees [31]. Such resulting fuzzy random forest combines characteristics of CEs 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 kind of classification problems.

Some advanced Evolutionary Fuzzy System-based contributions should also be remarked. On the one hand, a fuzzy rule-based classifier system (FRBCS) ensemble design technique is proposed in [32] considering feature selection methods based on some niching GA [33] 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 the rule-based ensemble design method using a single- and multiobjective genetic selection process is introduced in [34,35]. 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 [36], where an EMO algorithm generated a Pareto set of FRBCSs with different accuracy-complexity trade-offs to be combined into an ensemble.

2.3. Pruning a set of component classifiers in the classifier ensemble

Typically, an ensemble of classifiers is post-processed in such a way only a subset of them are kept for the final decision. It is a well known fact that the size of this CE is an important issue for its trade-off between accuracy and complexity [18,19] and that most of the error reduction occurs with the first few additional classifiers [7,19]. Furthermore, the selection process also participates in the elimination of the duplicates or the poor-performing classifiers.

While in the first studies on CEs a very small number (around ten) of component classifiers was considered as appropriate to sufficiently reduce the test set prediction error, later research on boosting (that also holds for bagging) suggested that error can be significantly reduced by largely exceeding this number [37]. This has caused the use of very large ensemble sizes (for example comprised by 1000 individual classifiers) in the last few years [18].

Hence, the determination of the optimal size of the ensemble is an important issue for obtaining both the best possible accuracy in the test data set without overfitting it, and a good accuracy-complexity trade-off. In pure bagging and boosting approaches, the optimal ensembles are directly composed of all the individual classifiers generated until a specific stopping point, which is determined according to different means (validation data set errors, likelihood, etc.). For example, an heuristic method to determine the optimal number guided by the *out-of-bag* error is proposed in [18].

However, there is the chance that the optimal ensemble is not comprised by all the component classifiers first generated but on a subset of them carrying a larger degree of disagreement/diversity. This is why different classifier selection methods have been proposed [38]. GAs have been commonly used for this task as we will show in the following subsection.

2.4. Related work on OCS-based genetic selection of classifier ensembles

The selection of a subset of classifiers is commonly done using the OCS strategy [10,39], in which a large set of classifiers is produced and then selected to extract the best performing subset. GAs are a popular technique within this strategy. In the literature, performance, complexity and diversity measures are usually considered as search criteria. Complexity measures are employed to simplify the system, whereas diversity measures are used to avoid overfitting. The reader is referred to [40] for a review on these genetic CE selection approaches.

Among the different genetic OCS methods, we can remark those most related to our current proposal. Oliveira et al. presented in [41] a hierarchical multiobjective GA, performing feature selection at the first level and classifier selection at the second level, which outperforms classical methods for two handwritten recognition problems. The multiobjective GA allows both performance and diversity to be considered for CE selection. Another EMO proposal for classifier selection is introduced in [42]. In that contribution, a comparison of a single-objective GA and the NSGA-II EMO algorithm for 14 different objective functions based on combining the mentioned three families of criteria (12 diversity measures, the training error, and the number of classifiers as a complexity measure) is developed. The authors applied their study on only one dataset, a digit handwritten recognition problem with 10 classes and 118,735 instances. They concluded that the training error is the best criterion for a single GA and a combination of training error and one diversity measure is the best criterion for an EMO algorithm. In [43] a genetic classifier selection method was considered based on a single performance index, either the diversity, including 16 different measures, or the ensemble error. The best results were obtained with the accuracy measure and a specific kind of diversity measures correlated with the error.

Finally, in [11] we proposed to use NSGA-II with five bi-objective fitness functions based on three different optimization criteria (accuracy, complexity, and diversity) for the component classifier selection in bagging FURIA-based fuzzy CEs. A combination between accuracy and diversity criteria showed very promising results.

2.5. Evolutionary fuzzy systems

Fuzzy systems, which are based on fuzzy logic, became popular in the research community, since they have ability to deal with complex, non-linear problems being too difficult for the classical methods [44]. Besides, its capability of knowledge extraction and representation allowed them to become human-comprehensible to some extent (more than classical black-box models) [45,46].

The lack of the automatic extraction of fuzzy systems have attracted the attention of the computational intelligence community to incorporate learning capabilities to these kinds of systems. In consequence, a hybridization of fuzzy systems and GAs has become one of the most popular approaches in this field [47–50]. In general, evolutionary fuzzy systems (EFSs) are fuzzy systems enhanced by a learning procedure coming from evolutionary computation, i.e. considering any evolutionary algorithm (EA).

Fuzzy rule-based systems (FRBSs), which are based on fuzzy "IF–THEN" rules, constitute one of the most important areas of fuzzy logic applications. Designing FRBSs might be seen as a search problem in a solution space of different candidate models by encoding the model into the chromosome, as EAs are well known optimization algorithms capable of searching among large spaces with the aim of finding optimal (usually nearly optimal) solutions.

The generic coding of EAs provides them with a large flexibility to define which FRBS parameters/components are to be designed [49]. For example, the simplest case would be a parameter optimization of the fuzzy membership functions. The complete rule base can also be learned. This capability allowed the field of EFSs to grow over two decades and to still be one of the most important topics in computational intelligence.

In the current contribution, we will relay on the EFS paradigm to define our EMO OCS-based FRBCE design method.

3. Using random oracles to design fuzzy rule-based classifier ensembles

An RO [3,14] is a structured classifier, also defined as a "mini-ensemble", encapsulating the base classifier of the CE. It is composed of two classifiers and an oracle that decides which one to use in each case. Basically, the oracle is a random function whose objective is to randomly split the dataset into two subsets by dividing the feature space into two regions. Each of the two generated regions and the corresponding data subset is assigned to one classifier. Any shape for the decision surface of the function can be applied as far as it divides the training set into two subsets at random.

In a preceding contribution [2], we used RLOs within our CE framework [4] to derive FRBCEs. Thanks to the additional diversity introduced by RLOs into the base classifiers, the obtained FRBCEs were able to achieve an outstanding performance in terms of accuracy.

In the current work, we enhance our previous study by combining RSOs with FRBCEs. In Section 3.1 we recall the RLO approach, while in Section 3.2 we introduce the RSO approach. Section 3.3

describes the RO-based FRBCE framework. Then, the experiments developed to benchmark both approaches and their analysis are shown in Section 3.4. We kindly refer the interested reader to [14,3] for more details regarding ROs.

3.1. Random linear oracle

A RLO uses a randomly generated hyperplane to divide the feature space. To generate a RLO the following procedure was proposed [3]:

- Select randomly a pair of examples from the training set
- Find the line segment between these points, passing through a middle point M
- Calculate the hyperplane perpendicular to the obtained line segment and containing M

3.2. Random spherical oracle

A RSO is based on a hypersphere where one classifier is responsible for the subspace inside of that hypersphere, while the second classifier is in charge of the rest of the feature space (outside of the hypersphere). The generation procedure of RSO is as follows [14]:

- Select randomly at least the half ($\geq 50\%$) of the features
  - Choose randomly an example from the training set to be the center of the hypersphere
  - Calculate distances from the center to E examples from the training set (chosen at random); the median of these distances is the radius of the hypersphere

Notice that, the random feature subset selection is done in order to improve the randomness, thus the diversity of the RSO. Moreover, the method itself is scalable, meaning that it is weakly affected by the number of attributes and not affected at all by the number of examples.

3.3. Random oracles fuzzy classifier ensemble framework

In this subsection, we will detail how the RO-based bagging FRBCEs are designed. To generate RO-based FRBCEs, a normalized dataset is split into two parts, a training set and a test set. The training set is submitted to the bagging procedure in order to provide K individual training sets (*bags*) to train RO (either RLO or RSO) mini-ensembles composed of the oracle and two FURIA [5,6] fuzzy subclassifiers. The oracles randomly split the bags into two parts and feed each FURIA classifier with the data from each half-space. As already said, RLO is based on a randomly generated hyperplane, which serves as a mean to divide the feature space. Alternatively, RSO does so using a random hypersphere. In total, $2 \times K$ FURIA-based fuzzy FRBCSs are generated in every case.

Let us emphasize that during the classification phase, the oracle commits an internal classifier selection, that is to say it decides which FURIA subclassifier makes the final decision for the given example to be further used at the ensemble level (classifier fusion).

Of course, we directly use the fuzzy classification rules generated by the FURIA algorithm. These fuzzy rules R_j^k show a class C_j^k and a certainty degree CF_j^k in the consequent: If x_1^k is A_{j1}^k and … and x_n^k is A_{jn}^k then Class C_j^k with CF_j^k, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, with J being a number of rules and K being a number of component classifiers. The voting-based fuzzy reasoning method is used to take the decision of the individual subclassifier [51,52].

After the training, we get an initial RO-based bagging FRBCE, which is validated using the training and the test errors, as well as a measure of complexity based on the total number of fuzzy rules obtained from the FURIA classifiers. The standard majority voting approach is applied as the classifier fusion method [1,53]: the ensemble class prediction will directly be the most voted class in the ROs output set. In the case of a tie the output class is chosen at random.

The global framework of the RO-based bagging FRBCE approach is presented in Fig. 1.

3.4. Experiments and analysis of results

This subsection is devoted to validate our framework using FURIA as a base classifier in RO-based bagging FRBCEs. Firstly, the experimental setup considered is introduced. Then, RSO-based

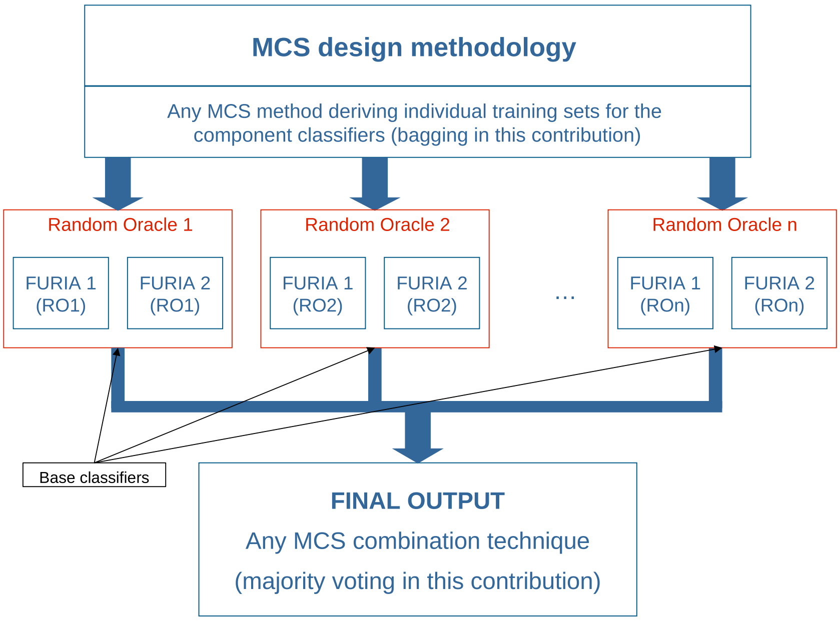


Fig. 1. Our initial framework: after obtaining bootstrapped replicas, the individual component classifiers are derived by RO composed of an oracle and two FURIA-based subclassifiers. The final output is taken by means of the majority voting, an inherent feature of bagging.

bagging FRBCEs are compared with RLO-based bagging FRBCE and bagging FRBCEs in order to show that RSOs have a better influence on the performance of bagging FRBCEs than the other approaches. Furthermore, RSO-based bagging FRBCEs are also compared with classical RSO-based bagging CEs and other kinds of classical CE design methodologies. By doing so, we aim to show that RSO-based bagging FRBCEs are competitive against the state-of-the-art RSO-based bagging CEs using C4.5 [3,14] and Naïve Bayes [14] as the base classifiers as well as *random forests* (RF) [25], when dealing with high complexity datasets, thanks to the use of the FURIA algorithm.

3.4.1. Experimental setup

To evaluate the performance of the RO-based bagging FRBCEs, 29 high dimensional data sets from the UCI machine learning repository [15] and the KEEL dataset repository [16] have been selected (see Table 1). Every attribute is tagged as real, integer, or nominal, denoted by "(R/I/N)" in the table. As it can be seen, the number of features ranges from 7 to 617,[1] while the number of examples does so from 1941 to 58,000. For illustrative purposes, we show in the table a complexity index computed as follows $\frac{\#ex.\times\#attr.}{10,000}$, denoted by "cmpl.".

In order to compare the accuracy of the considered classifiers, we used the Dietterich's 5×2-fold cross-validation (5×2-cv) [54]. The Friedman test and the Iman-Davenport are also used for assessing the statistical significance of the differences between algorithms, while the Holm test is carried out in case of $1\times n$ comparison and the Shaffer test is conducted in case of $n\times n$ comparison [55–57]. The confidence level considered for the null hypothesis rejection of all statistical tests considered is 5%.

We used the WEKA [58] implementations of the base classifiers with the default parameters (i.e. C4.5 is an unpruned tree). All the experiments have been run on an Intel quadri-core i5-2400 3.1 GHz processor with 4 GBytes of memory, under the Linux operating system.

3.4.2. Comparison of RSO-based bagging FRBCEs with RLO-based bagging FRBCEs and bagging FRBCEs

This subsection is devoted to analyze the performance of RSO combined with bagging FRBCEs. We compare them with the bagging FRBCEs approach proposed in [4], a base variant without RO. In order to make a fair comparison, we consider all CEs having a similar complexity based on the total number of rules in the FRBCSs. Notice that, although by embedding ROs into the CE the number of resulting classifiers in the ensemble increases by two (RO includes an oracle and two subclassifiers for each bag), the total number of rules in the FRBCEs does not necessarily have to increase by the same factor (it will be shown below, when analyzing Table 5). Thus, we consider the generated bagging FRBCEs comprised by 100 classifiers and RO-based bagging FRBCEs comprised by 75 classifiers only to achieve a similar complexity in terms of number of fuzzy rules in both ensembles.

The obtained results over the 29 selected datasets are presented in Table 2, that collects the test errors for the three FRBCEs considered. The best result for a given dataset is presented in bold font. The average "Avg." and standard deviation "Std. Dev." values over the 29 datasets are reported at the bottom of the table.

In view of this table, it can be noticed that both RO-based bagging FRBCEs outperform the original bagging FRBCEs considering the overall average test error as well as they also show a lower standard deviation. Taking each individual dataset into account, RLO-based bagging FRBCEs outperform bagging FRBCEs in 23 out of 29 cases (+1 tie), while RSO-based bagging FRBCEs do so in another 24 out of 29 cases (+1 tie).

[1] Notice that, our approach does not support nominal values (due to the use of FURIA as base fuzzy classifier learner, even if our generic framework does so). Thus, for abalone we simply remove this attribute. Besides, bioassay_688red contains 122 Boolean values (out of 126 integer values) considered as integer.

Table 1
Datasets considered.

Dataset	#ex.	#attr.	(R/I/N)	cmpl.	#classes
abalone	4178	8	(7/0/1)	3.3	28
bioassay_688red	27,190	153	(27/126/0)	416.0	2
coil2000	9822	85	(0/85/0)	83.5	2
gas_sensor	13,910	128	(128/0/0)	178.0	7
isolet	7797	617	(617/0/0)	481.1	26
letter	20,000	16	(0/16/0)	32.0	26
magic	19,020	10	(10/0/0)	19.0	2
marketing	6876	13	(0/13/0)	8.9	9
mfeat_fac	2000	216	(0/216/0)	43.2	10
mfeat_fou	2000	76	(76/0/0)	15.2	10
mfeat_kar	2000	64	(64/0/0)	12.8	10
mfeat_zer	2000	47	(47/0/0)	9.4	10
musk2	6598	166	(0/166/0)	109.5	2
optdigits	5620	64	(0/64/0)	36.0	10
pblocks	5474	10	(4/6/0)	5.5	5
pendigits	10,992	16	(0/16/0)	17.6	10
ring_norm	7400	20	(20/0/0)	14.8	2
sat	6436	36	(0/36/0)	23.2	6
segment	2310	19	(19/0/0)	4.4	7
sensor_read_24	5456	24	(24/0/0)	13.1	4
shuttle	58,000	9	(0/9/0)	52.2	7
spambase	4602	57	(57/0/0)	26.2	2
steel_faults	1941	27	(11/16/0)	5.2	7
texture	5500	40	(40/0/0)	22.0	11
thyroid	7200	21	(6/15/0)	15.1	3
two_norm	7400	20	(20/0/0)	14.8	2
waveform_noise	5000	40	(40/0/0)	20.0	3
waveform	5000	21	(21/0/0)	10.5	3
wquality_white	4898	11	(11/0/0)	5.4	7

Table 2
A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging FRBCEs (100 classifiers) in terms of accuracy. FURIA serves as the base classifier in the three approaches.

Dataset	BAG test err.	BAG + RLO test err.	BAG + RSO test err.
abalone	0.7455	0.7452	**0.7480**
bioassay_688red	**0.0090**	**0.0090**	**0.0090**
coil2000	0.0602	**0.0601**	**0.0601**
gas_sensor	0.0086	0.0079	**0.0078**
isolet	0.0774	**0.0691**	0.0700
letter	0.0778	**0.0742**	0.0743
magic	0.1325	0.1314	**0.1299**
marketing	0.6749	0.6673	**0.6671**
mfeat_fac	0.0547	0.0434	**0.0431**
mfeat_fou	0.1992	0.1941	**0.1925**
mfeat_kar	0.0825	**0.0699**	0.0709
mfeat_zer	0.2231	**0.2169**	0.2181
musk2	0.0338	0.0328	**0.0320**
optdigits	0.0324	0.0283	**0.0282**
pblocks	**0.0335**	0.0353	0.0338
pendigits	0.0155	0.0137	**0.0132**
ring_norm	0.0432	0.0438	**0.0315**
sat	0.1013	0.1008	**0.1001**
segment	0.0309	0.0303	**0.0295**
sensor_read_24	**0.0222**	0.0227	0.0233
shuttle	**0.0008**	0.0009	0.0009
spambase	0.0663	0.0651	**0.0639**
steel_faults	0.2371	0.2367	**0.2361**
texture	0.0288	0.0278	**0.0274**
thyroid	**0.0212**	0.0215	0.0218
two_norm	0.0316	**0.0271**	0.0276
waveform_noise	0.1480	0.1461	**0.1457**
waveform	0.1480	**0.1451**	0.1453
wquality_white	0.3908	0.3840	**0.3803**
Avg.	0.1286	0.1259	**0.1252**
Std. Dev.	0.1833	0.1825	0.1829

Please cite this article in press as: K. Trawiński et al., Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?, Knowl. Based Syst. (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.006

Table 3
Average rankings of the Friedman's test.

Algorithm	Ranking
FURIA + BAG + RSO	1.552
FURIA + BAG + RLO	1.828
FURIA + BAG	2.621

Table 4
The adjusted p-values of Shaffer test for the pair-wise comparisons (FURIA is the base classifier in every case).

Comparison	p-value
BAG + RSO vs BAG	+(1.41e−4)
BAG + RLO vs BAG	**+(0.002)**
BAG + RSO vs BAG + RLO	=(0.293)

It seems that RSO-based bagging FRBCEs is the approach worth pointing out as it obtains the lowest overall average test error. In addition, it gets the highest number of the best individual results (17 + 2 ties). Nonetheless, a clear conclusion about which RO-based approach is the best one cannot be drawn as RLO-based bagging FRBCEs are not much inferior in terms of overall average test error.

The average rankings of each CE obtained through the Friedman test are shown in Table 3. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to $3.338533e-5$, which is much lower than the assumed α-value 0.05.

These conclusions are confirmed in Table 4, which presents the adjusted p-values of the Shaffer test comparing all the FRBCE design approaches (the results showing a significant difference are presented in bold font). Both RO-based bagging FRBCEs show significant differences in comparison with bagging FRBCEs. However, the statistical test did not indicate significant differences between RSO- and RLO-based bagging FRBCEs.

Table 5
A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging CEs (100 classifiers) in terms of complexity (number of rules). FURIA serves as the base classifier in the three approaches.

Dataset	BAG # Rules	BAG + RLO # Rules	BAG + RSO # Rules
abalone	**8369.0**	8696.7	9382.8
bioassay_688red	5526.9	**4642.8**	4780.8
coil2000	4331.9	**3804.1**	4002.1
gas_sensor	8628.3	**7091.3**	7310.7
isolet	12215.7	**10523.6**	10828.5
letter	47109.1	**39410.5**	40972.9
magic	13770.8	**13143.0**	14556.9
marketing	**6418.5**	7252.0	7429.1
mfeat_fac	3479.9	**3050.2**	3110.3
mfeat_fou	5483.5	**4711.4**	4886.9
mfeat_kar	4953.3	**4448.4**	4581.0
mfeat_zer	5028.3	**4349.9**	4549.2
musk2	4332.2	**3581.1**	3582.7
optdigits	7167.3	**6352.4**	6511.1
pblocks	3201.7	2877.9	**2816.4**
pendigits	8788.6	**7348.0**	7491.6
ring_norm	7308.9	6205.7	**5961.4**
sat	8454.4	**6956.2**	7109.5
segment	2546.3	**2201.6**	2378.7
sensor_read_24	3430.8	**3340.4**	3428.3
shuttle	1826.2	**1723.8**	1737.5
spambase	3612.9	**3281.9**	4181.1
steel_faults	5467.3	**4799.0**	4857.0
texture	6537.2	**5305.7**	5542.8
thyroid	3299.5	**2831.7**	2959.8
two_norm	6147.5	**4973.3**	5307.8
waveform_noise	7932.6	**6729.9**	6850.6
waveform	8303.0	**7017.3**	7115.0
wquality_white	13429.3	**12134.0**	12564.4
Avg.	7831.1	6854.6	7130.6
Std. Dev.	8144.6	6857.3	7156.8
Avg. (without letter)	6428.3	5691.9	5921.9
Std. Dev. (without letter)	3100.2	2847.3	3030.4

As already mentioned, we use the overall number of rules in the FRBCEs as a measure of the ensemble complexity, while the number of classifiers composing the ensemble was fixed after a preliminary study. Table 5 shows the corresponding values for each of the three FRBCEs.

In the light of this table, it can clearly be noticed that both RO-based bagging FRBCEs obtain a lower complexity than the original bagging FRBCEs in terms of overall average number of rules. RLO-based bagging FRBCEs obtain the lowest overall average number of rules, as well as the lowest individual number of rules in 25 out of 29 cases (even though the difference with RSO-based bagging FRBCEs is very small). Notice that, the overall standard deviation values are very high due to the large number of rules obtained for the letter dataset. Because of that, we also present the overall average number of rules and the overall standard deviation for the 28 remaining datasets at the bottom of this table.

The average rankings of each CE obtained through the Friedman test concerning the number of rules are shown in Table 6. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to $6.578824e-15$.

These conclusions are confirmed in Table 7, which presents the adjusted p-values of the Shaffer test comparing all the FRBCE design approaches (the results showing a significant difference are presented in bold font). Both RO-based bagging FRBCEs show significant differences in comparison with bagging FRBCEs. RLO-based bagging FRBCEs show significant differences in comparison with RSO-based bagging FRBCEs.

Overall, we may conclude that RO-based bagging FRBCEs significantly outperform bagging FRBCEs both in terms of accuracy and complexity. A decision whether to choose RLO or RSO is not straightforward since RLO obtains slightly lower accuracy, but also lower complexity, while RSO does the opposite (slightly higher accuracy at the cost of a small complexity increase). For the purpose of this contribution, which considers obtaining FRBCEs with a good accuracy-complexity trade-off but mainly focusing on accuracy as usual, we will choose the RSO approach for the further comparisons.

3.4.3. Comparison of RSO-based FRBCEs with classical classifier eensembles

In this subsection we compare RSO-based bagging FRBCEs with classical RSO-based bagging CEs using C4.5 [3,14] and Naïve Bayes (NB) [14] as the base classifiers. See Tables A.24 and A.25 in the

Table 6
Average rankings of the Friedman's test.

Algorithm	Ranking
FURIA + BAG + RLO	1.138
FURIA + BAG + RSO	2.069
FURIA + BAG	2.793

Table 7
The adjusted p-values of Shaffer test for the pair-wise comparisons (FURIA is the base classifier in every case).

Comparison	p-value
BAG + RLO vs BAG	**+(8.77e−10)**
BAG + RSO vs BAG	**+(0.006)**
BAG + RLO vs BAG + RSO	**=+(3.92e−4)**

Appendix for the results of both RO-based bagging CEs using C4.5 and NB. It can be noticed that the RSO-based approach is the best performing approach in both cases. We also perform a comparison with the state-of-the-art algorithm, RF [25] using 100 base classifiers.

In this case, a complexity comparison in terms of the number of rules in the base classifiers is rather impossible, because NB is not a rule-based classifier and C4.5 considers tree-based rules. Besides, the number of classifiers is already prefixed to get a similar complexity, so we will not compare the different CEs in terms of this parameter.

Table 8 presents the test results achieved by RSO-based bagging FRBCEs and RSO-based bagging CE using C4.5 and NB, as well as RF over the 29 datasets.

In the light of this table, it can be noticed that RSO-based bagging FRBCEs outperform the other approaches considering the overall average test error. It also obtains the highest number of wins (12 + 2 ties) in the individual datasets. On the opposite, RSO-based bagging CEs based on NB turns out to be the worst choice both considering overall the average test error and the number of individual best results. Let us emphasize that RSO-based bagging FRBCEs obtains a lower overall average test error than RF, which is a very powerful CE algorithm, thus showing the quality of the proposed design. Notice that, RSO-based bagging CEs with C4.5 also outperform RF as a consequence of the use of the RO approach.

The average rankings of each CE obtained through the Friedman test are shown in Table 9. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to $1.074188e{-}6$.

The adjusted p-values of the Holm test comparing RSO-based bagging FRBCEs (the control algorithm) with the rest of the FRBCE design approaches (the results showing a significant difference are presented in bold font) are presented in Table 10. It reveals significant differences in favor of RSO-based bagging FRBCEs when comparing with RSO-based bagging CEs using NB. That is not the case when comparing with RSO-based bagging CEs using C4.5 and RF. We aim to improve even more RSO-based bagging FRBCE accuracy by incorporating the OCS strategy, as it will be shown in the following section.

Concluding, RSO-based bagging FRBCEs not only outperform classical RSO-based bagging CEs using C4.5 and NB, but they also obtain better results in comparison with the state-of-the-art RF algorithm. Thus, we may draw the conclusion that RO-based bagging FRBCEs successfully deal with high complexity datasets, being competitive with classical RO-based bagging CEs.

Table 8
A comparison of RSO-based bagging CEs using FURIA, C4.5 and NB, as well as RF in terms of accuracy.

Dataset	FURIA test err.	C4.5 test err.	NB test err.	RF test err.
abalone	**0.7480**	0.7681	0.7619	0.7536
bioassay_688red	**0.0090**	**0.0090**	0.0152	**0.0090**
coil2000	0.0601	0.0615	0.1847	**0.0597**
gas_sensor	**0.0078**	0.0089	0.2939	0.0092
isolet	**0.0700**	0.0788	0.1246	0.0766
letter	0.0743	**0.0615**	0.2927	0.0701
magic	0.1299	**0.1255**	0.2391	0.1314
marketing	0.6671	0.6735	0.6864	**0.6624**
mfeat_fac	**0.0431**	0.0498	0.0659	0.0475
mfeat_fou	0.1925	0.1902	0.2221	**0.1858**
mfeat_kar	0.0709	0.0818	**0.0593**	0.0597
mfeat_zer	**0.2181**	0.2273	0.2464	0.2330
musk2	0.0320	**0.0271**	0.1107	0.0375
optdigits	0.0282	**0.0276**	0.0709	0.0277
pblocks	0.0338	**0.0327**	0.0706	0.0332
pendigits	**0.0132**	0.0150	0.0864	0.0162
ring_norm	**0.0315**	0.0376	0.0199	0.0587
sat	0.1001	**0.0950**	0.1720	0.1027
segment	**0.0295**	0.0328	0.1180	0.0350
sensor_read_24	0.0233	0.0234	0.3710	**0.0224**
shuttle	**0.0009**	**0.0009**	0.0143	**0.0009**
spambase	0.0639	0.0651	0.1788	**0.0625**
steel_faults	0.2361	**0.2263**	0.3441	0.2517
texture	**0.0274**	0.0334	0.1384	0.0383
thyroid	**0.0218**	0.0222	0.0381	0.0221
two_norm	0.0276	0.0280	**0.0219**	0.0389
waveform_noise	**0.1457**	0.1643	0.1668	0.1556
waveform	**0.1453**	0.1588	0.1534	0.1587
wquality_white	0.3803	**0.3688**	0.5230	0.3864
Avg.	**0.1252**	0.1274	0.1997	0.1292
Std. Dev.	0.1829	0.1852	0.1890	0.1830

4. EMO OCS classifier selection for RSO-based FRBCEs

The second part of this contribution introduces the use of the OCS strategy for classifier selection in RO-based fuzzy CEs. On the one hand, the aim is to refine the accuracy-complexity trade-off in the RO-based bagging FRBCEs (both in RLO and RSO) when dealing with high complexity classification problems. On the other hand, an interesting objective is to study whether the additional diversity induced by ROs is beneficial for the EMO OCS-based component fuzzy classifier selection. Thus, we opted for the state-of-the-art NSGA-II EMO algorithm, which in fact was successively used for OCS with bagging FRBCEs in [11] (considering biobjective fitness functions), in order to generate good quality Pareto set approximations. In this study, we propose a specific design customized to the RO characteristics as well as use a three-objective fitness function including accuracy, complexity, and diversity measures for classifier selection.

Since RO is composed of the pair of the component classifiers, our classifier selection is done at the level of the component classifiers and not the whole pair of classifiers. The adapted coding scheme, which permits that none, one or both FURIA subclassifiers can be selected, is proposed. We also develop a reparation operator, whose objective is to correct the unfeasible solutions.

Fig. 2 shows the final structure of the RO-based bagging FRBCE design methodology including the OCS stage. The two following subsections briefly present the proposed algorithm operation mode and the three evaluation criteria used for the three-objective fitness function.

4.1. The main components of NSGA-II

NSGA-II [12] is based on a Pareto dominance depth approach, where the population is divided into several fronts and the depth of each front shows to which front an individual belongs to. A

Table 9
Average rankings of the Friedman's test.

Algorithm	Ranking
FURIA + BAG + RSO	1.793
C4.5 + BAG + RSO	2.276
RF	2.345
NB + BAG + RSO	3.586

Table 10
The adjusted p-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE (using FURIA) is the control method.

Comparison	p-value
FURIA + BAG + RSO vs C4.5 + BAG + RSO	=(0.207)
FURIA + BAG + RSO vs RF	=(0.207)
FURIA + BAG + RSO vs NB + BAG + RSO	**+(3.69e−7)**

pseudo-dominance rank being assigned to each individual, which is equal to the front number, is a metric used for the selection of an individual.

In our coding scheme, a binary digit corresponds to each gene, i.e. to a single FURIA base classifier composing the RO pair. When the gene takes the value 1, it means that the component classifier belongs to the final ensemble, while when the gene is equal to 0, that classifier is discarded. This representation has a low operational cost, which leads to the high speed of the algorithm.

We have used a generational approach and an elitist replacement strategy. The initial population is composed of randomly generated individuals. To introduce a high amount of diversity, binary tournament is used as selection mechanism. That means that two individuals are randomly picked from the current population and the best one is selected. We have considered the classical two-point crossover and two different mutation operator settings. The first one is the standard bit-flip mutation, while the second is the bit-flip mutation with biased probabilities proposed in [59]. In the latter case, the probability of generating a 0 is higher than the probability of giving a 1. Both crossover and mutation operators are applied with different pre-specified probabilities.

The proposed coding scheme, however, has one drawback. Since the oracle assigns only a half-region of the feature space to each component classifier, NSGA-II might select a subset of classifiers that do not cover the entire feature space. To avoid this problem, at least one pair of component classifiers composing the RO pair is forced to be selected. For that purpose, we developed a reparation operator, which is executed after the genetic operators (mutation and crossover) and is enabled when the said condition is not fulfilled. In this case all the possible combinations containing a single RO pair of component classifiers are generated. Then, the reparation operator evaluates them in the objective space and removes the dominated ones. Finally, one of the non-dominated solutions is selected at random.

4.2. The three used evaluation criteria for three-objective NSGA-II

In this subsection we describe all the optimization criteria considered by NSGA-II for the RO-based FRBCE OCS task. We will utilize measures of three different kinds embedded into the three-objective fitness function:

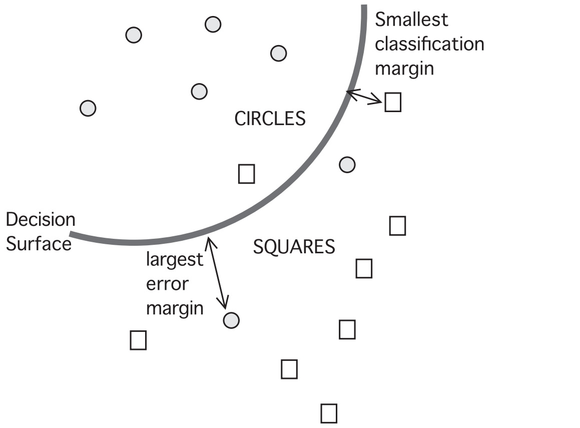


Fig. 3. The accuracy objective of NSGA-II has three components: (a) a quantile of the bootstrap estimation of the training error, (b) the largest distance between a misclassified example and the decision surface, and (c) the smallest distance between a correctly classified example and the decision surface.

- **Accuracy**. We use the accuracy of a selected ensemble, which was proposed in [60], defined by three components (e, m_1, m_2) (see Fig. 3), thus the first objective of NSGA-II is a triplet comprising three real numbers:
  1. Training error e: we compute the error of each ensemble for a large number of bootstrapped resamples of the training set, and use a quantile of the distribution of these errors as the first term of the fitness. This is intended to avoid overfitting when there are outliers in the training set, and also to detect the most robust selections, which are expected to generalize better.
  2. Error margin m_1: the second component of the fitness function depends on the distance between the misclassified examples and their nearest decision surface. Given an example x, we have approximated this value by the difference between the highest and the second highest term of $R^{II}(x)(c)$, and defined that the error margin of an ensemble is the worst (i.e. the highest) value of this difference for any example x in the training set.
  3. Classification margin m_2: the third component depends on the distance between the correctly classified instances and their nearest decision surface, which is approximated as

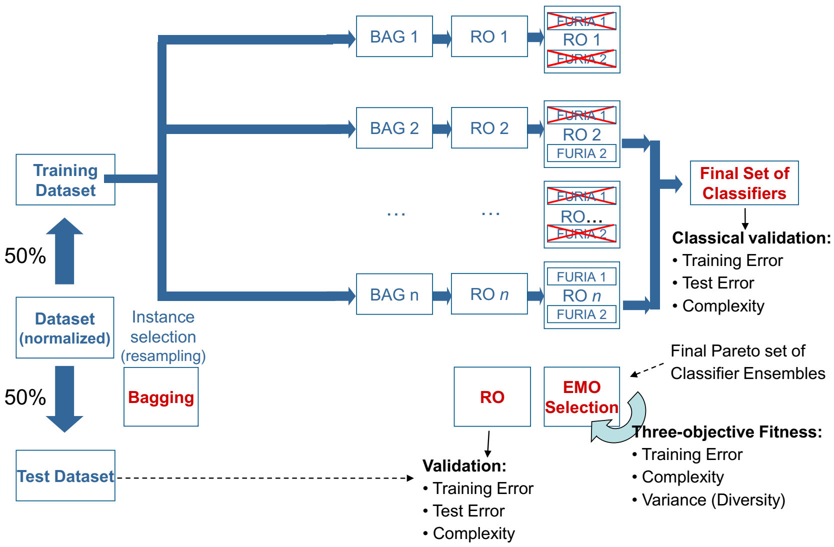


Fig. 2. Our final framework: after obtaining bootstrapped replicas, the component classifiers are derived by the specific RO method (either RLO or RSO) using FURIA as subclassifiers. Then, the OCS takes place by means of NSGA-II with a three-objective fitness function providing a Pareto set of simplified FRBCEs. In every case, the output is obtained using a voting-based method.

before, by the difference between the highest and the second highest terms in $R^{II}(x)(c)$. In this case, however, the margin of an ensemble is the lowest value of this difference for all the examples of the training set; we seek a decision surface with the highest margin.

Given an instance x, let us define the "winner rule" as the rule with the highest activation and the "most promising rule" for classifying this pattern as the rule with the highest activation among those whose consequent is different than that of the winner rule.

The decision surface is formed by the points for which there is a tie between the activations of the winner rule and the most promising rule. In this respect, if an instance is close to the decision surface, the difference between these two activations will be small. If the instance is moved towards the decision surface, this difference will be further decreased. The opposite is also true: if the instance is separated from the decision surface, this difference will increase. As a consequence, we can take this difference as a measure of distance between the instance and the nearest decision surface.

A lexicographical ordering is defined between each two triplets:$(e, m_1, m_2) \prec (e', m_1', m_2') \Leftrightarrow$

$$\begin{cases} (e < e\prime) \\ (e = e\prime) \text{ and } m_1 < m_1\prime \\ (e = e\prime) \text{ and } (m_1 = m_1\prime) \text{ and } (m_2 > m_2\prime) \end{cases} \tag{1}$$

- **Complexity**. The complexity of the ensemble is directly accounted by the number of classifiers in the ensemble:

$$Complx = |E| \tag{2}$$

- **Diversity**. Obtaining a high diversity between classifiers is the base goal to be reached when aiming to achieve performance improvement in CEs. In the last few years, a group of researchers devoted their attention to diversity measures [22,43,61]. In this contribution we consider the difficulty measure θ. This measure is computed as follows. Let $X = \{i/|E|\}_{i \in \{0,...,|E|\}}$ and $X_k \in X$ be the proportion of classifiers classifying correctly the instance x^k. Then, θ is defined as follows:

$$\theta = Var(\{X_1, \ldots, X_k, \ldots, X_m\}) \tag{3}$$

Notice that, the use of fitness functions based on diversity measures has been justified by previous findings in the specialized literature. Diversity measures have been deeply studied in [22,39,42,43,61]. However, the relationship between diversity measures and accuracy is still not clear. In [61], it was showed how the ensemble accuracy and diversity are not as strongly correlated as it could be expected. The authors concluded that accuracy estimation cannot be substituted by diversity during the CE design process. These results were confirmed by Ruta et al. [43] in classifier selection, our current application domain. In the experimental study developed, the authors drew the conclusion that the use of a single-objective function based on a diversity measure does not outperform the direct use of an error rate. Hence, the combined action of both kinds of measures can lead to a better performance in our case.

The idea of using a three-objective evaluation function came from our previous contribution [11] in which we conducted an extensive study of the OCS method on FRBCEs using NSGA-II and five different two-objective fitness functions obtained from four evaluation criteria. The most promising results were obtained by the combination of TE and θ, which led to an accuracy improvement while keeping a low complexity. However, a preliminary study showed us that, when combined with RO-based FRBCEs, a two-objective fitness function is not enough to give the chance to obtain a low number of classifiers. This is why we decided to also include explicitly the complexity as an additional objective.

We should remark that, the three objectives proposed are not directly correlated, hence using them jointly may lead to good accuracy-complexity trade-off solutions, always keeping in mind that accuracy is the most important learning goal.

4.3. The EMO variants for the comparison purpose

To study the influence of the additional diversity induced by the RO approach in the performance of the final FRBCE selected by the specific RO NSGA-II, we compare it with the EMO-based OCS classifier selection of bagging FRBCEs.

For that purpose, we use NSGA-II with the same components as the specific RO NSGA-II, apart from one. Since bagging FRBCEs are composed of the FURIA fuzzy classifiers, a standard binary coding is used. That is to say, a binary gene is assigned to each base classifier. Of course, in this approach there is no need for a reparation operator. From now on, this version of NSGA-II we will call "standard NSGA-II".

Furthermore, a different choice for selecting classifiers among RO-based bagging FRBCEs is possible. That could be performed by selecting directly the whole RO pairs (instead of a single component classifier). That is to say the binary coding assigns a single binary gene to each RO pair. Hence, it is clear that this second variant has a lower number of freedom degrees associated. Besides, the reparation operator is not necessary in this approach. Thus, standard NSGA-II can be applied. All the EMO variants use the three-objective fitness function described in Section 4.2.

We will compare the proposed NSGA-II for RLO- and RSO-based bagging FRBCEs classifier selection with the latter two EMO OCS methods. Table 11 summarizes the seven resulting EMO OCS-based FRBCEs approaches. In order to make the manuscript easy to follow, from now on we will refer to them using the abbreviations presented in this table.

4.4. Experiments and analysis of results

This section reports all the experiments performed on the EMO classifier selection procedure. Firstly, we introduce the experimental setup (Section 4.4.1). Then, in Section 4.4.2, we compare the performance of the seven FRBCE variants in Table 11. A unary multiobjective metric [13] is considered to analyze the results. We also show graphs of the obtained Pareto front approximations. Furthermore, we study some representative individual solutions selected

Table 11
The different variants resulting from the three EMO approaches used for the classifier selection.

Abbreviation	Base classifier	CE methodology	OCS strategy	Mut. type
BAS-BAG	FURIA	Bagging	Standard NSGA-II	Standard
BAS-RLO	RLO (2 × FURIA + oracle)	Bagging + RLO	Standard NSGA-II	Standard
ADV-RLO	RLO (2 × FURIA + oracle)	Bagging + RLO	Specific RO NSGA-II	Standard
ADV-BI-RLO	RLO (2 × FURIA + oracle)	Bagging + RLO	Specific RO NSGA-II	Biased
BAS-RSO	RSO (2 × FURIA + oracle)	Bagging + RSO	Standard NSGA-II	Standard
ADV-RSO	RSO (2 × FURIA + oracle)	Bagging + RSO	Specific RO NSGA-II	Standard
ADV-BI-RSO	RSO (2 × FURIA + oracle)	Bagging + RSO	Specific RO NSGA-II	Biased

from the obtained Pareto sets in Section 4.4.3. Finally, we compare the best EMO-based FRBCE variant with some non-simplified classical CEs in Section 4.4.4.

4.4.1. Experimental setup

To evaluate the performance of the generated EMO OCS-based FRBCEs, we use the same 29 datasets as in Section 3.4 (see Table 1). In order to compare the accuracy of the considered classifiers, we also consider the Dietterich's 5×2-fold cross-validation (5×2-cv) [54].

The RO-based bagging FRBCEs generated are initially comprised by 75 RO pairs (150 base classifiers), while bagging FRBCEs are initially comprised by 100 base classifiers. Both versions of NSGA-II for the component classifier selection work with a population of 200 individuals and runs during 1000 generations. The number of genes is equal to the number of base classifiers in every case but in variants BAS-RLO and BAS-RSO. In these cases, its size is half of the number of classifiers, as the result of the selection of the entire RO pairs. The crossover probability considered is 0.6 and the standard mutation probability is 0.1, while the biased mutation probability is 0.1 from 1 to 0 and 0.033, three times less, from 0 to 1. A different run is developed with each of the variants proposed for each initial FRBCE, thus resulting in 10 different runs per dataset as a consequence of the 5×2-cv procedure. All the experiments have been run in the same computer used in Section 3.4.

To compare the results obtained we computed their values for the two learning goals, namely test error (the primary goal) and complexity, that are supposed to be considered by the designer in order to choose the final fuzzy CE structure.

Notice that, in order to make a fair comparison, we consider the final complexity in terms of the total number of rules instead of the total number of classifiers, since the RO-based FRBCEs produce twice as much classifiers as the only bagging approach and usually their component classifiers are less complex than a standard base classifier (as already explained in Section 3.4.2). In contrast, it is necessary to use the number of classifiers as the optimization criterion (as a EMO objective), since using the total number of rules will lead to a bias in the NSGA-II optimization process. That is, as each classifier has a different number of rules, the ones having a lower number of rules will be promoted. Thus, some of the classifiers will have a higher probability (based on number of rules) of being used in the final ensemble. However, our assumption is that each classifier should be treated in the same way, having the same chance to be selected in the final ensemble. In order to perform a comparison of the Pareto front approximations of the global learning objectives (i.e. CE test accuracy and complexity) we consider one of the most extended multiobjective metrics, the hypervolume ratio (HVR). A detailed explanation of the HVR metric is to be found in [13].

Let us call $\overline{P_i^j}$ the non-dominated solution set returned by NSGA-II using the variant of fitness function i in the j-th run for a specific problem instance; $P_i = \overline{P_i^1} \bigcup \overline{P_i^2} \bigcup \ldots \bigcup \overline{P_i^{10}}$, the union of the solution sets returned by the ten runs obtained from 5x2-cv of algorithm i, and finally $\overline{P_i}$ the set of all non-dominated solutions in the P_i set[2] (aggregated Pareto fronts). As a complement to the analysis of the results obtained in the two different multiobjective metrics, we will provide graphical representations of some of those aggregated Pareto fronts. When graphically represented, these plots offer a valuable visual information, not measurable, but sometimes more useful than numerical values.

[2] Notice that, the pseudo-optimal Pareto front is the fusion of the $\overline{P_i}$ sets generated by every variant of the EMO OCS-based FRBCEs in all the runs developed.

Table 12
Comparison of Pareto fronts using the HVR measure.

	BAS-BAG	BAS-RLO	ADV-RLO	ADV-BI-RLO	BAS-RSO	ADV-RSO	ADV-BI-RSO
aba	0.8248	0.8594	0.6399	**0.8878**	0.8378	0.7305	0.8500
bio	0.8343	0.9073	0.8059	**0.9825**	0.9115	0.9118	0.9678
coi	0.6929	0.7419	0.5687	**0.7548**	0.7251	0.6497	0.6477
gas	0.8590	0.9404	0.6876	**0.9771**	0.9382	0.8435	0.9642
iso	0.8611	0.9118	0.7661	**0.9534**	0.9074	0.8571	0.9155
let	0.9127	0.9477	0.7961	0.9726	0.9626	0.8945	**0.9727**
mag	0.7970	0.8423	0.6444	**0.9061**	0.8433	0.8119	0.8737
mar	0.7214	0.8217	0.6569	**0.8689**	0.8170	0.7994	0.8225
mfa	0.8874	0.9463	0.7886	**0.9763**	0.9439	0.8717	0.9600
mfo	0.8373	0.8838	0.7145	**0.9322**	0.8809	0.8040	0.8931
mka	0.8661	0.9227	0.7643	**0.9631**	0.9091	0.8418	0.9211
mze	0.8041	0.8650	0.6498	**0.9183**	0.8560	0.7702	0.8660
mus	0.7112	0.8098	0.6161	**0.8779**	0.8172	0.7071	0.8122
opt	0.8721	0.9316	0.7662	**0.9669**	0.9322	0.8411	0.9415
pbl	0.7487	0.7794	0.6038	0.7231	0.8052	0.7764	**0.8421**
pen	0.8617	0.9375	0.6873	**0.9752**	0.9419	0.8106	0.9609
rin	0.8187	0.8526	0.6878	0.8803	0.9221	0.8954	**0.9222**
sat	0.8436	0.9219	0.7196	**0.9613**	0.9284	0.8296	0.9468
seg	0.8551	0.9081	0.7621	**0.9358**	0.9080	0.8172	0.8417
sen	0.8597	0.9234	0.6630	**0.9644**	0.9228	0.8043	0.9503
shu	0.9347	0.9192	0.7051	0.9645	0.9176	0.7858	**0.9661**
spa	0.8196	0.8932	0.6805	**0.9343**	0.8690	0.8535	0.9109
ste	0.8206	0.8836	0.6620	**0.9264**	0.8877	0.7998	0.9053
tex	0.8713	0.9308	0.7769	**0.9614**	0.9288	0.8388	0.9444
thy	0.8368	0.9084	0.6804	**0.9560**	0.9025	0.8303	0.9487
two	0.8774	0.9558	0.7478	**0.9814**	0.9392	0.8880	0.9565
wan	0.8566	0.8881	0.7335	**0.9397**	0.8873	0.8400	0.8890
wav	0.8426	0.9033	0.7163	**0.9300**	0.8989	0.8367	0.9192
wqu	0.7914	0.8567	0.6973	**0.9098**	0.8724	0.8119	0.8881
Avg.	0.8317	0.8894	0.7031	**0.9269**	0.8901	0.8191	0.9035
Std. Dev.	0.0618	0.0524	0.0564	0.0681	0.0562	0.0522	0.0608

Table 13
Average rankings of the Friedman's test.

Algorithm	Ranking
ADV-BI-RLO	1.345
ADV-BI-RSO	2.207
BAS-RLO	3.310
BAS-RSO	3.517
BAS-BAG	5.103
ADV-RSO	5.517
ADV-RLO	6.999

Table 14
The adjusted p-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE is the control method (FURIA is the base classifier in every case).

Comparison	p-value
ADV-BI-RLO vs ADV-RLO	**+(1.26e−22)**
ADV-BI-RLO vs ADV-RSO	**+(9.56e−13)**
ADV-BI-RLO vs BAS-BAG	**+(1.38e−10)**
ADV-BI-RLO vs BAS-RSO	**+(3.85e−4)**
ADV-BI-RLO vs BAS-RLO	**+(0.003)**
ADV-BI-RLO vs ADV-BI-RSO	=(0.771)

4.4.2. Comparison of the three EMO classifier selection variants in the two global learning objectives

Table 12 presents the results obtained in the HVR metric. Our analysis of the HVR measure clearly points out the best performing approach for the final learning goal. The ADV-BI-RLO variant, considering the RLO-based bagging FRBCEs with the specific RO NSGA-II method proposed in this contribution and the biased mutation, obtained the highest value in 25 out of 29 cases showing an overall good quality of the Pareto front approximations. Variant ADV-BI-RSO obtained the highest value 4 times, being not much worst than

ADV-BI-RLO. Concerning the average value on the 29 datasets considered, the order of the best algorithms is ADV-BI-RLO, ADV-BI-RSO, BAS-RSO, BAS-RLO, BAS-BAG, ADV-RSO and finally ADV-RLO.

The average rankings of each EMO variant using HVR metric obtained through the Friedman test are shown in Table 13. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to 5.942658e−64.

Table 14 presents the adjusted p-values of the Holm test comparing ADV-BI-RLO (the control algorithm) with the rest of the EMO variants (the results showing a significant difference are presented in bold font). ADV-BI-RLO show significant differences in comparison with all the other variants apart from the ADV-BI-RSO.

The results obtained in the experimentation developed corroborate our initial assumption that the additional diversity provided by the RO approach is beneficial for the FRBCEs designed. The best performing approaches are based on RLO and RSO. Notice that, the HVR metric considered in the comparison measures the overall quality of the Pareto front approximations obtained with respect to the two global learning goals, accuracy and complexity. Hence, the EMO OCS methods performing a stronger classifier selection are promoted. In the next subsection we will show how, when focusing on the main learning goal, test accuracy, a weaker selection is more beneficial and the specific RO EMO OCS method not using the biased mutation allows us to significantly improve the RO-based FRBCE accuracy.

In order to complement the previous analysis and get a deep insight of the results obtained, the aggregated Pareto fronts will be graphically represented for two datasets: letter and sensor_read_24 (see Fig. 4). These figures allow an easy visual comparison of the performance of the different EMO OCS-based FRBCEs variants.

It can be seen that our ADV-BI-RLO and ADV-BI-RSO approaches generate FRBCE designs spreading widely over the Pareto search space. ADV-BI-RLO seems to be obtaining better Pareto front solutions, however ADV-BI-RSO does not perform much worse. Differences with the other approaches are less significant in letter (Fig. 4a) than in sensor_read_24 (Fig. 4c).

In order to make a stronger conclusion, particular solutions are extracted from the Pareto front approximations and analyzed in detail in the next section.

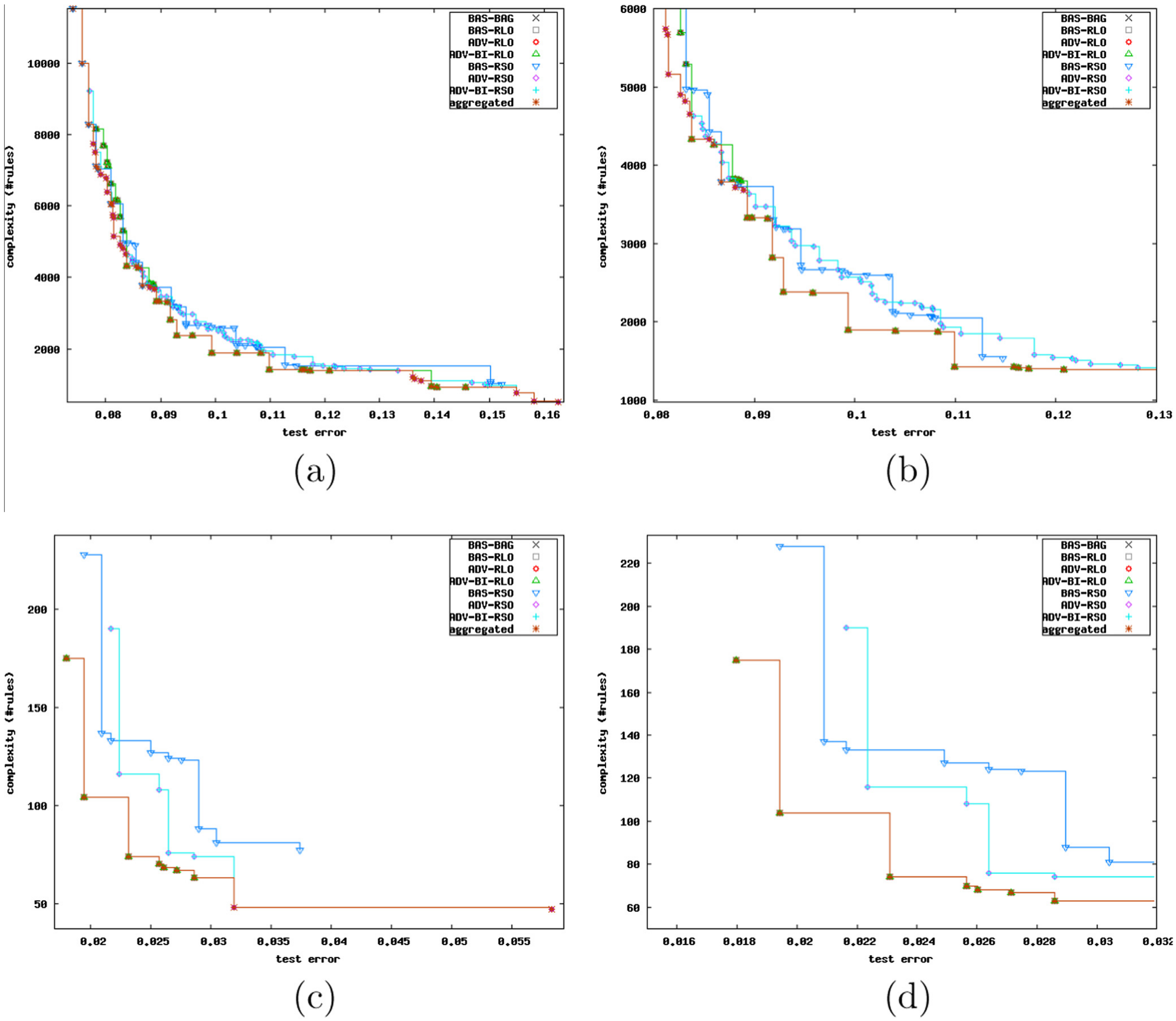


Fig. 4. Graphical representations of the Pareto front approximations obtained from the three EMO approaches for two datasets: (a) letter, (b) letter (zoom), (c) sensor_read_24, and sensor_read_24 (zoom). Objective 1 stands for test error and objective 2 for complexity in terms of the number of rules. The pseudo-optimal Pareto front is also drawn for reference.

Please cite this article in press as: K. Trawiński et al., Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?, Knowl. Based Syst. (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.006

Table 15
A comparison of the averaged performance of the four single solutions selected from the obtained Pareto sets.

		Best complx				Best diversity				Best train				Best trade-off (tra-div-cmpl)			
		Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst
Avg.	BAS-BAG	445	0.1392	0.0694	0.1399	558	0.1371	0.0695	0.1396	2003	0.2173	0.0506	0.1306	647	0.1554	0.0640	0.1317
	BAS-RLO	202	0.0874	0.0983	0.1736	303	0.0842	0.0962	0.1726	1748	0.2176	0.0436	0.1301	404	0.1151	0.0667	0.1381
	ADV-RLO	1058	0.0203	0.0525	0.1314	3044	0.0165	0.0476	0.1268	2663	0.0172	0.0435	0.1267	2078	0.0172	0.0473	0.1235
	ADV-BI-RLO	90	0.0390	0.0916	0.1680	1606	0.0163	0.0474	0.1271	1269	0.0173	0.0426	0.1276	480	0.0185	0.0489	0.1264
	BAS-RSO	205	0.0904	0.0926	0.1688	308	0.0874	0.0905	0.1682	1883	0.2243	0.0420	0.1292	402	0.1201	0.0646	0.1384
	ADV-RSO	587	0.2633	0.0523	0.1307	778	0.1818	0.0514	0.1290	2164	0.6111	0.0409	0.1271	836	0.2228	0.0501	0.1248
	ADV-BI-RSO	115	0.1180	0.1417	0.2114	670	0.0542	0.0628	0.1446	1463	0.3331	0.0392	0.1308	414	0.0624	0.0635	0.1380
Std. Dev.	BAS-BAG	512	0.0987	0.1500	0.1858	687	0.0987	0.1495	0.1852	3505	0.1539	0.1390	0.1832	731	0.1088	0.1478	0.1833
	BAS-RLO	222	0.0668	0.1511	0.1925	496	0.0640	0.1418	0.1881	2976	0.1565	0.1213	0.1821	541	0.0845	0.1396	0.1832
	ADV-RLO	1267	0.0230	0.1319	0.1845	4234	0.0203	0.1274	0.1831	3508	0.0210	0.1206	0.1828	2745	0.0209	0.1276	0.1812
	ADV-BI-RLO	96	0.0395	0.1535	0.1928	2218	0.0202	0.1272	0.1830	1645	0.0211	0.1196	0.1829	542	0.0221	0.1309	0.1818
	BAS-RSO	205	0.0688	0.1472	0.1906	499	0.0662	0.1366	0.1867	3461	0.1622	0.1194	0.1820	454	0.0878	0.1372	0.1837
	ADV-RSO	686	0.1195	0.1303	0.1845	927	0.0871	0.1307	0.1842	3269	0.3958	0.1156	0.1829	989	0.1020	0.1288	0.1818
	ADV-BI-RSO	120	0.0665	0.1462	0.1870	764	0.0483	0.1287	0.1833	2235	0.2822	0.1135	0.1839	519	0.0517	0.1375	0.1838

4.4.3. Analysis and comparison of single solutions extracted from the obtained Pareto front approximations

In this section, our objective is to analyze the final performance of our proposal by imitating the procedure expected to be followed by a human designer in order to select a desired FURIA-based fuzzy CE structure from those available in the obtained accuracy-complexity non-dominated fronts.

From each Pareto front approximation, we have selected four different solutions, the one having the best value in the three objectives that have been optimized: complexity (in terms of number of rules), diversity, and training error; and the one with the best trade-off value among the latter three measures. The trade-off solution is selected as follows: 1000 random weights $w1$, $w2$, $w3 \in [0, 1]$ are computed for each solution, the average value of the aggregation function of three learning goals (complexity, diversity, and training error) $LG1$, $LG2$ and $LG3$ is taken as: $(w1{*}LG1 + w2{*}LG2 + w3{*}LG3)$, and the solution with the highest aggregated value is selected. For each solution we present the values of four different measures, complexity (*Cmpl*), diversity (*Div*), training error (*Tra*), and test error (*Tst*). The average and standard deviation values for each of the four different solutions in the 29 problems are collected in Table 15 (the values for each specific dataset are presented in B, together with the cardinality of each Pareto set approximation).

As our approach involves the joint optimization of three different objectives, in our opinion their mixture is the best combination for the selection of the final solution. Thus, we will focus mostly on the solutions with the best trade-off (cmpl-div-tra) value. From these results we may draw the following conclusions:

- The ADV-RLO, ADV-RSO, and ADV-BI-RLO variants actually outperform the standard BAS-BAG variant considering test accuracy, which shows a good behavior of the approach proposed. However, the ADV-RLO and ADV-RSO variants do so at the cost of obtaining a higher number of rules when looking at the complexity criterion. Notice that, the best complexity solution for ADV-RSO could be alternatively selected from the Pareto front, as it also outperforms BAS-BAG both in terms of test error and complexity (see, for example, the best complexity and diversity solutions for ADV-RSO in Table 15).
- The biased mutation obtains very good results in terms of complexity, as it significantly reduces the number of rules in the final FRBCEs. Considering the best complexity it manages to decrease the number of rules to 90 and 115 for ADV-BI-RLO and ADV-BI-RSO, respectively.
- In general, the proposed NSGA-II approaches with three learning objectives derive good quality solutions, which are widely spread among the Pareto front. They reach both edges acquiring high performance for the two learning goals: accuracy (ADV-RSO and ADV-RLO) and complexity (ADV-BI-RSO and ADV-BI-RLO).
- The best performance in terms of test accuracy was obtained by the ADV-RLO variant, even though it obtained quite weak Pareto front approximations in the previous subsection. This fact is justified by the HVR metric nature, as already mentioned in that section.

To verify the results obtained, we carried the Friedman test on the test error values, whose average rankings are shown in Table 16. The Iman-Davenport test indicates significant differences between the algorithms, as the p-value is equal to $2.816171e{-}38$.

Table 17 presents the adjusted p-values of the Holm test comparing ADV-RLO (the control algorithm) with the rest of the EMO variants (the results showing a significant difference are presented in bold font). ADV-RLO shows significant differences in comparison with all the other variants but the ADV-RSO. No statistical difference between ADV-RSO and ADV-RLO could be expected, since the only difference between these algorithms is the kind of oracle used. The statistical differences indicated by the Holm test also help us to answer the question asked at the beginning of the paper.

Table 16
Average rankings of the Friedman's test.

Algorithm	Ranking
ADV-RLO	1.603
ADV-RSO	2.138
ADV-BI-RLO	3.345
BAS-BAG	3.707
BAS-RSO	5.603
BAS-RLO	5.638
ADV-BI-RSO	5.966

Table 17
The adjusted p-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE is the control method (FURIA is the base classifier in every case).

Comparison	p-value
ADV-RLO vs ADV-BI-RSO	**8.89e−014**
ADV-RLO vs BAS-RLO	**5.73e−012**
ADV-RLO vs BAS-RSO	**7.11e−012**
ADV-RLO vs BAS-BAG	**0.0006**
ADV-RLO vs ADV-BI-RLO	**0.0043**
ADV-RLO vs ADV-RSO	0.3461

Please cite this article in press as: K. Trawiński et al., Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?, Knowl. Based Syst. (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.006

It can be noticed that the additional diversity induced by the ROs to the base classifier is also beneficial for the final accuracy of the FRBCEs designed.

Concluding, the proposed approaches (ADV-RSO and also ADV-RLO) generate very good results in terms of the test accuracy of the final FRBCEs, which is the lowest of all the variants considered. That is confirmed by the statistical tests for the best individual FRBCE design according to the test error. The ADV-RLO accuracy results showed significant differences with respect to all the other variants apart from ADV-RSO. In addition, the latter approach obtains very high complexity reduction. The additional diversity provided by RO actually introduces more degrees of freedom than a standard OCS-based genetic selection. Taking advantage from this fact we have been able to design both highly accurate, which is the main learning goal, and significantly less complex FRBCEs.

4.4.4. Comparison between the EMO-selected FRBCEs and non-selected classical classifier ensembles

In this section we compare the ADV-RLO and ADV-RSO EMO variants, which turned out to be the best performing in Section 4.4.3, with the full original ensemble, RO-based bagging FRBCEs composed of 75 base classifiers. The FRBCE designs considered are those corresponding to the best trade-off values in Table 15. Besides, we perform a comparison with the classical CE approaches RO-based bagging CEs using C4.5 and RF.

Table 18
A comparison of RLO-based bagging CEs using FURIA, C4.5 and NB, as well as RF in terms of accuracy.

Dataset	ADV-RLO test err.	FURIA + BAG + RLO test err.	C4.5 + BAG + RLO test err.	RF test err.
abalone	**0.7425**	0.7452	0.7666	0.7536
bioassay_688red	0.0091	**0.0090**	**0.0090**	**0.0090**
coil2000	0.0603	0.0601	0.0612	**0.0597**
gas_sensor	**0.0075**	0.0079	0.0097	0.0092
isolet	0.0693	**0.0691**	0.0803	0.0766
letter	0.0852	0.0742	**0.1559**	0.0701
magic	0.1285	0.1314	**0.1254**	0.1314
marketing	**0.6620**	0.6673	0.6728	0.6624
mfeat_fac	**0.0427**	0.0434	0.0484	0.0475
mfeat_fou	**0.1825**	0.1941	0.1932	0.1858
mfeat_kar	0.0655	0.0699	0.0766	**0.0597**
mfeat_zer	**0.2108**	0.2169	0.2285	0.2330
musk2	0.0297	0.0328	**0.0271**	0.0375
optdigits1	**0.0270**	0.0283	0.0290	0.0277
pblocks	0.0336	0.0353	**0.0333**	0.0332
pendigits	**0.0127**	0.0137	0.0155	0.0162
ring_norm	**0.0409**	0.0438	0.0558	0.0587
sat	0.0980	0.1008	**0.0953**	0.1027
segment	**0.0263**	0.0303	0.0338	0.0350
sensor_read_24	**0.0218**	0.0227	0.0228	0.0224
shuttle	**0.0006**	0.0009	0.0009	0.0009
spambase	**0.0604**	0.0651	0.0650	0.0625
steel_faults	0.2293	0.2367	**0.2265**	0.2517
texture	**0.0262**	0.0278	0.0348	0.0383
thyroid	0.0218	**0.0215**	0.0222	0.0221
two_norm	**0.0260**	0.0271	0.0266	0.0389
waveform	**0.1422**	0.1461	0.1630	0.1556
waveform1	**0.1430**	0.1451	0.1599	0.1587
wquality_white	0.3762	0.3840	**0.3714**	0.3864
Avg.	**0.1235**	0.1259	0.1314	0.1292
Std. Dev.	0.1812	0.1825	0.1844	0.1830

Table 19
Average rankings of the Friedman's test.

Algorithm	Ranking
ADV-RLO	1.586
FURIA + BAG + RLO	2.534
RF	2.879
C4.5 + BAG + RLO	3.000

Table 20
The adjusted *p*-values of Holm test for the pair-wise comparisons where RLO-based bagging FRBCE (using FURIA) is the control method.

Comparison	*p*-value
ADV-RLO vs C4.5 + BAG + RLO	**9.13e−005**
ADV-RLO vs RF	**2.73e−004**
ADV-RLO vs FURIA + BAG + RLO	**0.0051**

Table 18 reports the test error of the RLO-based CEs on the 29 datasets. It can be clearly noticed that the ADV-RLO FRBCEs obtain the highest accuracy. They outperform the other approaches in 17 out of 29 cases, obtaining the lowest average test error.

The average rankings of each CE obtained through the Friedman test are shown in Table 19. The Iman-Davenport test indicates significant differences between the algorithms as the *p*-value is equal to $6.443102e{-}5$.

The adjusted *p*-values of the Holm test comparing ADV-RLO (the control algorithm) with the rest of the CE design approaches are presented in Table 20 (the results showing a significant difference are presented in bold font). It reveals significant differences in favor of ADV-RLO when comparing with all the big ensembles. Notice that, in addition, ADV-RLO obtains roughly a 70% of the complexity reduction in comparison with FURIA + BAG + RLO.

Meanwhile, Table 21 reports the test error of the RSO-based CEs on the 29 datasets. The ADV-RSO FRBCEs obtain the highest accuracy, as it happened for ADV-RLO in Table 18. They outperform the

Table 21
A comparison of RSO-based bagging CEs using FURIA, C4.5 and NB, as well as RF in terms of accuracy.

Dataset	ADV-RSO test err.	FURIA + BAG + RSO test err.	C4.5 + BAG + RSO test err.	RF test err.
abalone	**0.7425**	0.7480	0.7681	0.7536
bioassay_688red	0.0091	**0.0090**	**0.0090**	**0.0090**
coil2000	0.0601	0.0601	0.0615	**0.0597**
gas_sensor	0.0079	**0.0078**	0.0089	0.0092
isolet	0.0744	**0.0700**	0.0788	0.0766
letter	0.0780	0.0743	**0.0615**	0.0701
magic	0.1294	0.1299	**0.1255**	0.1314
marketing	0.6654	0.6671	0.6735	**0.6624**
mfeat_fac	0.0453	**0.0431**	0.0498	0.0475
mfeat_fou	**0.1887**	0.1925	0.1902	0.1858
mfeat_kar	0.0740	0.0709	0.0818	**0.0597**
mfeat_zer	0.2188	**0.2181**	0.2273	0.2330
musk2	0.0316	0.0320	**0.0271**	0.0375
optdigits1	0.0294	0.0282	**0.0276**	0.0277
pblocks	0.0317	0.0338	**0.0327**	0.0332
pendigits	**0.0132**	**0.0132**	0.0150	0.0162
ring_norm	**0.0314**	0.0315	0.0376	0.0587
sat	0.0994	0.1001	**0.0950**	0.1027
segment	**0.0268**	0.0295	0.0328	0.0350
sensor_read_24	**0.0219**	0.0233	0.0234	0.0224
shuttle	**0.0005**	0.0009	0.0009	0.0009
spambase	**0.0606**	0.0639	0.0651	0.0625
steel_faults	0.2303	0.2361	**0.2263**	0.2517
texture	**0.0280**	0.0274	0.0334	0.0383
thyroid	**0.0214**	0.0218	0.0222	0.0221
two_norm	0.0282	**0.0276**	0.0280	0.0389
waveform	0.1464	**0.1457**	0.1643	0.1556
waveform1	0.1459	**0.1453**	0.1588	0.1587
wquality_white	0.3785	0.3803	**0.3688**	0.3864
Avg.	**0.1248**	0.1252	0.1274	0.1292
Std. Dev.	0.1818	0.1829	0.1852	0.1830

Please cite this article in press as: K. Trawiński et al., Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?, Knowl. Based Syst. (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.006

Table 22
Average rankings of the Friedman's test.

Algorithm	Ranking
ADV-RSO	1.983
FURIA + BAG + RSO	2.259
C4.5 + BAG + RSO	2.793
RF	2.965

Table 23
The adjusted *p*-values of Holm test for the pair-wise comparisons where RSO-based bagging FRBCE (using FURIA) is the control method.

Comparison	*p*-value
ADV-RSO vs C4.5 + BAG + RSO	**0.0112**
ADV-RSO vs RF	**0.0336**
ADV-RSO vs FURIA + BAG + RSO	0.4158

other approaches in 9 (+1 tie) out of 29 cases, obtaining the lowest average test error.

The average rankings of each CE obtained through the Friedman test are shown in Table 22. The Iman-Davenport test indicates significant differences between the algorithms as the *p*-value is equal to $6.443102e{-}5$.

The adjusted *p*-values of the Holm test comparing ADV-RSO (the control algorithm) with the rest of the CE design approaches are presented in Table 23. It reveals significant differences in favor of ADV-RSO when comparing with RF and C4.5 CEs but not with the FURIA + BAG + RSO approach. Even so, ADV-RSO obtains an outstanding complexity reduction keeping a similar (slightly better) accuracy level, showing good accuracy-complexity trade-off. When comparing to its full ensemble FURIA + BAG + RSO, the complexity reduction is almost 90% (exactly 88.3%), which eventually leads to a lower execution time and memory consumption with no performance decrease.

Thus, we can conclude that the proposed approach obtained very promising results, showing the profit resulting from the additional diversity provided by the RO approach.

5. Conclusions and future works

In this contribution, we focused on two aims. Firstly, we introduced a new RSO approach into our previous FRBCE design framework. Secondly, we incorporated the EMO-based OCS strategy to these kinds of classifiers analyzing the influence of the additional RO diversity in the final FRBCEs performance. We used an advanced accuracy measure and proposed a specific binary coding for the RO-based classifier selection. A three-objective fitness function using three different optimization criteria such as accuracy, complexity, and diversity metrics was used. By using the said techniques, we succeed to both obtain a high accuracy level and a good accuracy-complexity trade-off, when dealing with high complexity data.

We carried out exhaustive experiments using 29 high complexity datasets from the UCI and the KEEL repositories. It turned out that our EMO OCS approach for the RO-based FRBCEs provided very promising results.

Among the next steps to be considered, we include the use of different diversity measures in the objective space of the fitness function in order to improve the results obtained. A combination between ROs and the recently proposed FRBCS-based combination method including classifier fusion and classifier selection for a good interpretability-accuracy trade-off [60] is also an interesting future step to be considered. Finally, applying this approach to

Table A.24
A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging FRBCEs (100 classifiers) in terms of accuracy. C4.5 serves as a base classifier in both approaches.

Dataset	BAG	BAG + RLO	BAG + RSO
abalone	0.7676	**0.7666**	0.7681
bioassay_688red	**0.0090**	**0.0090**	**0.0090**
coil2000	0.0619	**0.0612**	0.0615
gas_sensor	0.0151	0.0097	**0.0089**
isolet	0.0929	0.0803	**0.0788**
letter	0.0801	**0.1559**	0.0615
magic	0.1267	**0.1254**	0.1255
marketing	0.6761	**0.6728**	0.6735
mfeat_fac	0.0588	**0.0484**	0.0498
mfeat_fou	0.2017	0.1932	**0.1902**
mfeat_kar	0.0998	**0.0766**	0.0818
mfeat_zer	0.2401	0.2285	**0.2273**
musk2	0.0314	**0.0271**	**0.0271**
optdigits	0.0441	0.0290	**0.0276**
pblocks	0.0339	0.0333	**0.0327**
pendigits	0.0236	0.0155	**0.0150**
ring_norm	0.0601	0.0558	**0.0376**
sat	0.1017	0.0953	**0.0950**
segment	0.0368	0.0338	**0.0328**
sensor_read_24	**0.0226**	0.0228	0.0234
shuttle	0.0010	**0.0009**	**0.0009**
spambase	0.0665	**0.0650**	0.0651
steel_faults	0.2279	0.2265	**0.2263**
texture	0.0399	0.0348	**0.0334**
thyroid	0.0224	**0.0222**	**0.0222**
two_norm	0.0338	**0.0266**	0.0280
waveform_noise	0.1699	**0.1630**	0.1643
waveform	0.1666	0.1599	**0.1588**
wquality_white	0.3739	0.3714	**0.3688**
Avg.	0.1340	0.1314	**0.1274**
Std. Dev.	0.1841	0.1844	0.1852

Table A.25
A comparison of RO-based bagging FRBCEs (75 classifiers) with bagging FRBCEs (100 classifiers) in terms of accuracy. NB serves as a base classifier in both approaches.

Dataset	BAG	BAG + RLO	BAG + RSO
abalone	0.7653	0.7653	**0.7619**
bioassay_688red	0.0222	**0.0143**	0.0152
coil2000	0.2119	**0.1797**	0.1847
gas_sensor	0.4293	0.3064	**0.2939**
isolet	0.1411	0.1279	**0.1246**
letter	0.3578	0.3022	**0.2927**
magic	0.2725	**0.2359**	0.2391
marketing	0.6925	0.6874	**0.6864**
mfeat_fac	0.0762	**0.0658**	0.0659
mfeat_fou	0.2362	0.2251	**0.2221**
mfeat_kar	0.0686	**0.0579**	0.0593
mfeat_zer	0.2660	0.2486	**0.2464**
musk2	0.1577	0.1110	**0.1107**
optdigits	0.0890	**0.0681**	0.0709
pblocks	0.0845	**0.0590**	0.0706
pendigits	0.1421	**0.0816**	0.0864
ring_norm	0.0201	**0.0199**	**0.0199**
sat	0.2038	**0.1706**	0.1720
segment	0.1897	0.1477	**0.1180**
sensor_read_24	0.4661	0.3728	**0.3710**
shuttle	0.0642	0.0176	**0.0143**
spambase	0.2273	0.2191	**0.1788**
steel_faults	0.3735	**0.3397**	0.3441
texture	0.2260	0.1855	**0.1384**
thyroid	0.0390	**0.0375**	0.0381
two_norm	0.0219	**0.0216**	0.0219
waveform_noise	0.2006	**0.1552**	0.1668
waveform	0.1905	**0.1429**	0.1534
wquality_white	0.5535	**0.5164**	0.5230
Avg.	0.2341	0.2029	**0.1997**
Std. Dev.	0.1938	0.1893	0.1890

Table B.26
Statistics of the Pareto front approximations with the global EMO objectives.

		Card.	Best complx				Best diversity				Best train				Best trade-off (tra-div-cmpl)			
			Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst
aba	BAS-BAG	171.6	650	0.1267	0.5781	0.7562	1421	0.1126	0.5770	0.7535	2788	0.1247	0.5153	0.7498	1187	0.1152	0.5679	0.7408
	BAS-RLO	199.1	340	0.1648	0.5911	0.7736	1107	0.1276	0.5527	0.7532	3059	0.1490	0.4393	0.7502	985	0.1305	0.5444	0.7444
	ADV-RLO	152.2	1448	0.0601	0.4950	0.7529	3601	0.0452	0.4712	0.7498	3377	0.0500	0.4355	0.7502	2568	0.0490	0.4727	0.7425
	ADV-BI-RLO	182.1	132	0.1233	0.6161	0.7850	1931	0.0449	0.4717	0.7493	1591	0.0515	0.4307	0.7509	825	0.0521	0.4762	0.7435
	BAS-RSO	199.4	360	0.1740	0.5827	0.7778	1127	0.1336	0.5383	0.7587	3515	0.1561	0.4226	0.7518	1214	0.1356	0.5208	0.7477
	ADV-RSO	121.0	1057	0.2613	0.4807	0.7544	1437	0.1752	0.4792	0.7533	3748	0.5691	0.4035	0.7511	1426	0.2138	0.4713	0.7417
	ADV-BI-RSO	188.1	157	0.2388	0.6107	0.7945	1134	0.1431	0.4869	0.7593	2066	0.4089	0.3926	0.7586	606	0.1655	0.5208	0.7527
bio	BAS-BAG	7.6	222	0.0094	0.0080	0.0092	421	0.0092	0.0088	0.0091	295	0.0098	0.0075	0.0092	244	0.0093	0.0086	0.0091
	BAS-RLO	2.7	135	0.0084	0.0052	0.0102	135	0.0083	0.0053	0.0102	139	0.0086	0.0048	0.0103	135	0.0084	0.0053	0.0102
	ADV-RLO	26.6	256	0.0055	0.0064	0.0095	572	0.0048	0.0077	0.0091	265	0.0054	0.0062	0.0095	386	0.0049	0.0069	0.0091
	ADV-BI-RLO	25.5	48	0.0055	0.0048	0.0103	247	0.0047	0.0075	0.0091	99	0.0052	0.0044	0.0105	130	0.0049	0.0065	0.0092
	BAS-RSO	3.7	130	0.0085	0.0052	0.0101	134	0.0083	0.0055	0.0101	136	0.0088	0.0049	0.0101	132	0.0084	0.0054	0.0100
	ADV-RSO	17.4	131	0.0163	0.0087	0.0091	140	0.0156	0.0088	0.0092	581	0.0629	0.0077	0.0092	139	0.0165	0.0088	0.0091
	ADV-BI-RSO	15.1	62	0.0079	0.1082	0.1105	103	0.0063	0.0859	0.0890	155	0.0141	0.0046	0.0105	72	0.0076	0.0069	0.0099
coi	BAS-BAG	14.0	185	0.0659	0.0569	0.0604	197	0.0638	0.0575	0.0601	753	0.0884	0.0530	0.0613	220	0.0653	0.0575	0.0599
	BAS-RLO	10.0	87	0.0579	0.0576	0.0603	87	0.0576	0.0578	0.0604	239	0.0722	0.0502	0.0628	89	0.0581	0.0576	0.0601
	ADV-RLO	31.8	364	0.0402	0.0541	0.0605	696	0.0365	0.0543	0.0607	546	0.0377	0.0514	0.0615	500	0.0372	0.0535	0.0603
	ADV-BI-RLO	53.4	48	0.0434	0.0557	0.0614	317	0.0363	0.0538	0.0612	133	0.0387	0.0484	0.0628	131	0.0383	0.0547	0.0605
	BAS-RSO	10.6	99	0.0578	0.0566	0.0606	101	0.0576	0.0570	0.0604	207	0.0697	0.0487	0.0629	101	0.0576	0.0570	0.0604
	ADV-RSO	24.9	137	0.1490	0.0556	0.0604	147	0.1444	0.0560	0.0604	773	0.6004	0.0521	0.0609	167	0.1574	0.0561	0.0601
	ADV-BI-RSO	22.4	54	0.0444	0.0544	0.0646	215	0.0379	0.0500	0.0627	240	0.1748	0.0464	0.0636	76	0.0406	0.0550	0.0625
gas	BAS-BAG	15.6	445	0.0346	0.0029	0.0123	457	0.0335	0.0030	0.0116	1475	0.0708	0.0005	0.0091	540	0.0373	0.0022	0.0107
	BAS-RLO	20.1	194	0.0164	0.0080	0.0233	196	0.0153	0.0083	0.0235	1090	0.0625	0.0004	0.0091	297	0.0219	0.0028	0.0122
	ADV-RLO	79.5	986	0.0025	0.0012	0.0093	2477	0.0018	0.0011	0.0081	1925	0.0020	0.0004	0.0081	1763	0.0020	0.0009	0.0074
	ADV-BI-RLO	140.7	89	0.0052	0.0087	0.0234	1280	0.0018	0.0011	0.0079	869	0.0020	0.0002	0.0086	492	0.0020	0.0007	0.0079
	BAS-RSO	21.9	197	0.0168	0.0079	0.0234	201	0.0159	0.0087	0.0243	1034	0.0648	0.0003	0.0097	297	0.0227	0.0033	0.0132
	ADV-RSO	59.0	504	0.3490	0.0017	0.0094	628	0.1964	0.0018	0.0093	1824	0.9081	0.0003	0.0085	711	0.2540	0.0015	0.0079
	ADV-BI-RSO	74.8	111	0.0707	0.1294	0.1437	565	0.0075	0.0030	0.0133	901	0.3685	0.0000	0.0100	286	0.0131	0.0052	0.0155
iso	BAS-BAG	37.5	610	0.2110	0.0158	0.1144	611	0.2109	0.0160	0.1141	1983	0.3795	0.0000	0.0855	1277	0.3115	0.0025	0.0871
	BAS-RLO	52.8	285	0.1222	0.0662	0.1807	288	0.1187	0.0680	0.1817	1675	0.3758	0.0000	0.0833	707	0.2231	0.0115	0.0979
	ADV-RLO	151.0	1703	0.0135	0.0026	0.0840	7113	0.0091	0.0002	0.0701	4020	0.0100	0.0000	0.0729	5031	0.0095	0.0001	0.0685
	ADV-BI-RLO	176.0	135	0.0383	0.0663	0.1779	3821	0.0091	0.0001	0.0705	1723	0.0100	0.0000	0.0741	1336	0.0103	0.0005	0.0722
	BAS-RSO	54.1	292	0.1265	0.0716	0.1901	295	0.1252	0.0720	0.1892	1836	0.3968	0.0000	0.0821	691	0.2236	0.0129	0.1016
	ADV-RSO	103.0	932	0.3177	0.0034	0.0857	1443	0.1663	0.0010	0.0780	1864	0.2089	0.0000	0.0773	1525	0.1925	0.0008	0.0747
	ADV-BI-RSO	160.7	171	0.1641	0.0973	0.2097	834	0.0764	0.0208	0.1189	2378	0.1301	0.0000	0.0824	995	0.0854	0.0069	0.0930
let	BAS-BAG	51.3	2932	0.1964	0.0227	0.1002	2932	0.1964	0.0227	0.1002	19,513	0.3465	0.0048	0.0802	4461	0.2317	0.0140	0.0886
	BAS-RLO	101.8	1244	0.0885	0.2491	0.3173	1244	0.0885	0.2491	0.3173	16,292	0.3311	0.0112	0.0856	3399	0.1538	0.0328	0.1096
	ADV-RLO	158.7	7199	0.0167	0.0172	0.0921	23,327	0.0135	0.0186	0.0913	19,721	0.0146	0.0113	0.0845	15,632	0.0141	0.0145	0.0851
	ADV-BI-RLO	182.9	543	0.0328	0.0599	0.1453	12,150	0.0135	0.0194	0.0913	9135	0.0148	0.0112	0.0850	4290	0.0148	0.0147	0.0870
	BAS-RSO	76.3	1130	0.1198	0.0721	0.1604	1140	0.1197	0.0727	0.1619	18,892	0.3989	0.0029	0.0755	2755	0.1982	0.0217	0.0991
	ADV-RSO	177.1	3845	0.4772	0.0112	0.0856	5177	0.2888	0.0084	0.0808	18,040	1.1457	0.0028	0.0761	5672	0.3277	0.0071	0.0778
	ADV-BI-RSO	194.5	676	0.1825	0.1700	0.2476	4218	0.0601	0.0180	0.0967	12,403	0.8690	0.0029	0.0777	3773	0.0728	0.0186	0.0958
mag	BAS-BAG	12.2	674	0.1631	0.1047	0.1388	674	0.1631	0.1047	0.1388	2853	0.2238	0.0869	0.1319	937	0.1727	0.0951	0.1331
	BAS-RLO	10.8	414	0.1199	0.1045	0.1458	422	0.1194	0.1050	0.1464	2130	0.2057	0.0732	0.1305	681	0.1395	0.0813	0.1346
	ADV-RLO	24.6	1646	0.0499	0.0833	0.1325	3327	0.0432	0.0769	0.1298	2747	0.0438	0.0732	0.1298	2699	0.0440	0.0754	0.1285
	ADV-BI-RLO	66.0	162	0.0772	0.0992	0.1461	1596	0.0427	0.0750	0.1300	1023	0.0443	0.0707	0.1307	731	0.0461	0.0732	0.1305
	BAS-RSO	9.8	467	0.1178	0.1003	0.1455	469	0.1172	0.1005	0.1461	2355	0.2087	0.0703	0.1299	690	0.1393	0.0790	0.1349
	ADV-RSO	65.8	794	0.3533	0.0827	0.1312	894	0.2779	0.0833	0.1304	4008	1.3268	0.0688	0.1298	1150	0.3861	0.0798	0.1288
	ADV-BI-RSO	111.4	265	0.1370	0.1101	0.1558	809	0.0542	0.0991	0.1474	2258	1.0684	0.0620	0.1311	863	0.0820	0.0799	0.1345
mar	BAS-BAG	200.0	609	0.1713	0.6036	0.6815	2786	0.1590	0.6016	0.6789	2302	0.1672	0.5678	0.6734	1562	0.1648	0.5837	0.6695
	BAS-RLO	199.4	357	0.1898	0.5958	0.6928	2446	0.1659	0.5542	0.6730	2908	0.1744	0.5063	0.6678	1356	0.1715	0.5509	0.6629
	ADV-RLO	130.9	1239	0.1066	0.5398	0.6757	2867	0.0972	0.5259	0.6691	2971	0.1002	0.5046	0.6665	2054	0.0993	0.5263	0.6620
	ADV-BI-RLO	175.1	98	0.1649	0.6111	0.6959	1508	0.0968	0.5245	0.6693	1403	0.0998	0.5015	0.6682	627	0.1023	0.5331	0.6627
	BAS-RSO	199.7	359	0.1873	0.5962	0.6949	2508	0.1657	0.5534	0.6742	2796	0.1745	0.5070	0.6648	1300	0.1696	0.5595	0.6673
	ADV-RSO	111.5	545	0.4481	0.5417	0.6739	746	0.3510	0.5451	0.6723	2847	1.0419	0.4955	0.6672	812	0.4130	0.5352	0.6648
	ADV-BI-RSO	174.9	108	0.2088	0.6265	0.7098	1000	0.1585	0.5360	0.6766	1649	0.7167	0.4885	0.6718	532	0.1966	0.5586	0.6743
mfa	BAS-BAG	32.9	196	0.1428	0.0107	0.0675	203	0.1406	0.0105	0.0672	390	0.2045	0.0000	0.0593	310	0.1832	0.0022	0.0550
	BAS-RLO	34.7	83	0.0862	0.0436	0.1225	86	0.0814	0.0454	0.1197	285	0.2014	0.0000	0.0599	182	0.1375	0.0100	0.0615
	ADV-RLO	83.2	468	0.0103	0.0024	0.0540	1145	0.0068	0.0002	0.0451	1037	0.0075	0.0000	0.0451	978	0.0071	0.0002	0.0423
	ADV-BI-RLO	106.7	35	0.0248	0.0439	0.1224	662	0.0067	0.0003	0.0468	252	0.0099	0.0000	0.0598	270	0.0077	0.0003	0.0453
	BAS-RSO	33.2	86	0.0841	0.0416	0.1173	87	0.0816	0.0433	0.1159	296	0.2056	0.0000	0.0616	197	0.1459	0.0086	0.0617
	ADV-RSO	71.0	281	0.2191	0.0030	0.0517	423	0.1612	0.0012	0.0486	453	0.1703	0.0000	0.0489	431	0.1707	0.0011	0.0453
	ADV-BI-RSO	92.9	46	0.0673	0.1759	0.2432	387	0.0441	0.0013	0.0596	472	0.1125	0.0000	0.0575	229	0.0487	0.0062	0.0618
mfo	BAS-BAG	33.3	291	0.2851	0.0347	0.2241	297	0.2831	0.0361	0.2242	1197	0.4456	0.0000	0.2013	388	0.3170	0.0231	0.2050
	BAS-RLO	40.6	122	0.1635	0.0986	0.2781	123	0.1605	0.1013	0.2795	948	0.4431	0.0000	0.1996	231	0.2337	0.0431	0.2160
	ADV-RLO	119.1	777	0.0224	0.0089	0.2018	2224	0.0162	0.0029	0.1917	2186	0.0166	0.0000	0.1923	1525	0.0175	0.0024	0.1822
	ADV-BI-RLO	160.6	61	0.0526	0.0990	0.2940	1183	0.0161	0.0025	0.1914	1223	0.0164	0.0000	0.1919	552	0.0178	0.0024	0.1853
	BAS-RSO	40.8	127	0.1674	0.1038	0.2836	131	0.1645	0.1091	0.2918	967	0.4494	0.0000	0.2026	220	0.2287	0.0462	0.2253
	ADV-RSO	85.3	487	0.3466	0.0113	0.2030	586	0.2638	0.0100	0.1995	1376	0.7376	0.0000	0.1968	608	0.3109	0.0070	0.1887
	ADV-BI-RSO	138.4	74	0.1629	0.1825	0.3534	524	0.0924	0.0319	0.2269	854	0.2934	0.0000	0.2088	314	0.1095	0.0313	0.2084

Table B.27
Statistics of the Pareto front approximations with the global EMO objectives.

		Card.	Best complx				Best diversity				Best train				Best trade-off (tra-div-cmpl)			
			Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst
mka	BAS-BAG	37.2	268	0.2258	0.0179	0.1168	268	0.2258	0.0179	0.1168	564	0.3160	0.0000	0.0947	500	0.2958	0.0044	0.0887
	BAS-RLO	38.7	118	0.1299	0.0809	0.2131	119	0.1290	0.0815	0.2145	460	0.3212	0.0000	0.0956	307	0.2427	0.0087	0.1001
	ADV-RLO	105.0	758	0.0144	0.0011	0.0837	2115	0.0100	0.0001	0.0700	2335	0.0102	0.0000	0.0701	1819	0.0104	0.0000	0.0648
	ADV-BI-RLO	125.1	56	0.0388	0.0680	0.1947	1183	0.0099	0.0001	0.0714	997	0.0102	0.0000	0.0712	490	0.0112	0.0001	0.0694
	BAS-RSO	40.4	126	0.1400	0.0773	0.2064	128	0.1368	0.0810	0.2086	500	0.3352	0.0000	0.0984	307	0.2472	0.0105	0.1055
	ADV-RSO	78.5	454	0.2982	0.0021	0.0900	697	0.2128	0.0010	0.0796	709	0.2188	0.0000	0.0796	670	0.2433	0.0005	0.0740
	ADV-BI-RSO	102.7	78	0.1569	0.1492	0.2608	586	0.0686	0.0000	0.0957	665	0.1351	0.0000	0.0972	409	0.0703	0.0047	0.0988
mze	BAS-BAG	31.1	279	0.3024	0.0786	0.2453	288	0.2998	0.0773	0.2419	1368	0.4530	0.0344	0.2287	344	0.3170	0.0705	0.2324
	BAS-RLO	42.2	114	0.1916	0.1436	0.3021	116	0.1896	0.1465	0.3048	1269	0.4737	0.0307	0.2218	219	0.2582	0.0825	0.2430
	ADV-RLO	115.8	760	0.0365	0.0434	0.2256	1937	0.0307	0.0367	0.2187	1837	0.0321	0.0304	0.2186	1482	0.0317	0.0350	0.2116
	ADV-BI-RLO	157.8	54	0.0763	0.1430	0.3085	1054	0.0306	0.0359	0.2205	875	0.0318	0.0298	0.2195	439	0.0327	0.0371	0.2146
	BAS-RSO	43.8	127	0.2014	0.1487	0.3068	131	0.1998	0.1515	0.3132	1360	0.4894	0.0299	0.2232	257	0.2823	0.0775	0.2471
	ADV-RSO	108.2	443	0.3937	0.0483	0.2309	537	0.2819	0.0452	0.2295	1414	0.7910	0.0298	0.2245	658	0.3243	0.0425	0.2177
	ADV-BI-RSO	159.8	80	0.2034	0.1815	0.3393	299	0.1299	0.0946	0.2709	999	0.5235	0.0290	0.2247	285	0.1540	0.0754	0.2391
mus	BAS-BAG	19.1	218	0.0609	0.0126	0.0396	219	0.0596	0.0134	0.0411	1061	0.1261	0.0012	0.0342	301	0.0676	0.0078	0.0330
	BAS-RLO	21.4	99	0.0312	0.0228	0.0486	99	0.0311	0.0229	0.0487	867	0.1203	0.0008	0.0322	155	0.0426	0.0080	0.0325
	ADV-RLO	51.7	478	0.0053	0.0054	0.0345	982	0.0041	0.0027	0.0320	1061	0.0044	0.0007	0.0313	782	0.0043	0.0023	0.0294
	ADV-BI-RLO	98.8	38	0.0162	0.0192	0.0447	512	0.0040	0.0026	0.0322	479	0.0042	0.0005	0.0297	158	0.0049	0.0030	0.0293
	BAS-RSO	22.3	99	0.0309	0.0245	0.0512	99	0.0309	0.0245	0.0512	908	0.1233	0.0005	0.0321	177	0.0456	0.0081	0.0317
	ADV-RSO	81.9	284	0.0704	0.0068	0.0339	353	0.0527	0.0063	0.0343	1276	0.3221	0.0006	0.0331	422	0.0811	0.0051	0.0315
	ADV-BI-RSO	115.1	59	0.0368	0.0256	0.0502	172	0.0195	0.0104	0.0380	683	0.1603	0.0001	0.0328	214	0.0312	0.0075	0.0328
opt	BAS-BAG	25.5	420	0.1069	0.0061	0.0426	428	0.1064	0.0057	0.0432	898	0.1655	0.0000	0.0367	686	0.1401	0.0014	0.0351
	BAS-RLO	37.0	168	0.0595	0.0323	0.0863	169	0.0589	0.0331	0.0896	745	0.1753	0.0000	0.0351	404	0.1103	0.0050	0.0404
	ADV-RLO	93.9	1013	0.0072	0.0015	0.0334	3006	0.0049	0.0002	0.0284	3005	0.0049	0.0000	0.0284	2255	0.0052	0.0003	0.0270
	ADV-BI-RLO	126.6	89	0.0172	0.0299	0.0851	1514	0.0049	0.0001	0.0290	1519	0.0049	0.0000	0.0292	593	0.0056	0.0001	0.0290
	BAS-RSO	36.3	171	0.0621	0.0331	0.0876	171	0.0610	0.0338	0.0881	795	0.1864	0.0000	0.0370	396	0.1104	0.0063	0.0421
	ADV-RSO	77.5	669	0.2247	0.0010	0.0347	908	0.1673	0.0007	0.0325	954	0.1833	0.0000	0.0332	1007	0.2193	0.0004	0.0293
	ADV-BI-RSO	106.5	112	0.0954	0.0584	0.1076	814	0.0305	0.0026	0.0409	998	0.1120	0.0000	0.0364	509	0.0349	0.0043	0.0406
pbl	BAS-BAG	16.8	174	0.0497	0.0191	0.0355	199	0.0481	0.0192	0.0343	652	0.0740	0.0117	0.0331	217	0.0501	0.0174	0.0329
	BAS-RLO	17.1	79	0.0306	0.0233	0.0409	80	0.0300	0.0241	0.0417	602	0.0760	0.0105	0.0351	119	0.0369	0.0169	0.0339
	ADV-RLO	44.2	329	0.0116	0.0152	0.0357	606	0.0097	0.0129	0.0345	733	0.0103	0.0105	0.0347	516	0.0099	0.0122	0.0336
	ADV-BI-RLO	78.6	37	0.0165	0.0234	0.0429	292	0.0094	0.0128	0.0357	267	0.0101	0.0096	0.0355	153	0.0101	0.0121	0.0339
	BAS-RSO	15.3	74	0.0311	0.0245	0.0408	76	0.0304	0.0249	0.0416	479	0.0703	0.0105	0.0349	130	0.0370	0.0171	0.0334
	ADV-RSO	52.1	151	0.2513	0.0154	0.0338	172	0.1606	0.0157	0.0336	851	1.3417	0.0107	0.0328	234	0.3057	0.0146	0.0317
	ADV-BI-RSO	60.5	44	0.0430	0.0298	0.0417	317	0.0077	0.0125	0.0342	367	0.5747	0.0086	0.0338	135	0.0154	0.0175	0.0330
pen	BAS-BAG	20.5	467	0.0511	0.0038	0.0196	467	0.0511	0.0038	0.0196	1620	0.1050	0.0001	0.0163	563	0.0565	0.0024	0.0171
	BAS-RLO	27.1	195	0.0254	0.0135	0.0378	196	0.0250	0.0138	0.0372	1284	0.1089	0.0001	0.0157	308	0.0359	0.0044	0.0213
	ADV-RLO	106.5	1110	0.0035	0.0013	0.0154	2860	0.0025	0.0007	0.0141	2274	0.0027	0.0001	0.0144	1986	0.0027	0.0007	0.0127
	ADV-BI-RLO	172.3	103	0.0075	0.0123	0.0374	1535	0.0025	0.0007	0.0140	1443	0.0027	0.0000	0.0145	667	0.0028	0.0006	0.0136
	BAS-RSO	24.4	194	0.0259	0.0136	0.0377	196	0.0254	0.0138	0.0379	1290	0.1101	0.0001	0.0148	347	0.0400	0.0039	0.0210
	ADV-RSO	83.7	691	0.2230	0.0013	0.0151	961	0.0939	0.0013	0.0147	1833	0.2488	0.0001	0.0144	985	0.1158	0.0010	0.0132
	ADV-BI-RSO	92.4	119	0.1054	0.0516	0.0737	651	0.0121	0.0020	0.0183	1108	0.1260	0.0000	0.0155	354	0.0131	0.0044	0.0211
rin	BAS-BAG	20.8	383	0.1114	0.0080	0.0543	383	0.1114	0.0080	0.0543	985	0.1945	0.0000	0.0454	635	0.1511	0.0021	0.0435
	BAS-RLO	29.8	168	0.0595	0.0170	0.0666	171	0.0593	0.0184	0.0693	864	0.1935	0.0000	0.0472	376	0.1105	0.0044	0.0484
	ADV-RLO	91.2	1052	0.0061	0.0014	0.0466	3254	0.0043	0.0002	0.0434	3189	0.0043	0.0000	0.0437	2281	0.0045	0.0001	0.0409
	ADV-BI-RLO	135.5	86	0.0151	0.0158	0.0674	1749	0.0042	0.0002	0.0437	1726	0.0043	0.0000	0.0433	640	0.0049	0.0003	0.0425
	BAS-RSO	31.2	149	0.0561	0.0213	0.0659	154	0.0546	0.0227	0.0677	799	0.1918	0.0000	0.0352	382	0.1137	0.0037	0.0387
	ADV-RSO	72.7	352	0.3508	0.0024	0.0392	474	0.2111	0.0015	0.0376	1591	0.8206	0.0000	0.0328	772	0.2955	0.0007	0.0310
	ADV-BI-RSO	130.3	103	0.0557	0.0435	0.0893	659	0.0251	0.0148	0.0609	1047	0.1181	0.0000	0.0395	456	0.0597	0.0043	0.0432
sat	BAS-BAG	28.7	511	0.1603	0.0266	0.1064	513	0.1585	0.0272	0.1071	2579	0.2727	0.0087	0.1013	620	0.1691	0.0247	0.1031
	BAS-RLO	38.1	201	0.0870	0.0518	0.1286	202	0.0858	0.0526	0.1297	2248	0.2690	0.0058	0.1031	392	0.1305	0.0268	0.1066
	ADV-RLO	85.2	990	0.0161	0.0154	0.1029	2734	0.0128	0.0088	0.1008	2642	0.0133	0.0057	0.1009	1875	0.0134	0.0091	0.0979
	ADV-BI-RLO	145.9	92	0.0315	0.0503	0.1281	1428	0.0128	0.0092	0.1000	1344	0.0132	0.0052	0.1007	641	0.0139	0.0096	0.0978
	BAS-RSO	39.7	195	0.0847	0.0531	0.1287	197	0.0838	0.0540	0.1308	2178	0.2689	0.0055	0.1008	397	0.1300	0.0258	0.1070
	ADV-RSO	121.0	597	0.2356	0.0159	0.1042	743	0.1751	0.0143	0.1029	2566	0.7476	0.0050	0.1008	843	0.2120	0.0146	0.0993
	ADV-BI-RSO	167.1	122	0.1613	0.1016	0.1684	762	0.0508	0.0282	0.1139	1598	0.5394	0.0043	0.1019	455	0.0636	0.0248	0.1053
seg	BAS-BAG	26.6	140	0.0690	0.0082	0.0364	153	0.0640	0.0084	0.0373	453	0.1074	0.0000	0.0305	222	0.0771	0.0041	0.0282
	BAS-RLO	23.4	61	0.0352	0.0182	0.0520	70	0.0335	0.0160	0.0491	355	0.1092	0.0000	0.0311	119	0.0506	0.0052	0.0306
	ADV-RLO	44.3	285	0.0054	0.0030	0.0333	534	0.0038	0.0004	0.0299	535	0.0038	0.0000	0.0293	463	0.0042	0.0004	0.0262
	ADV-BI-RLO	59.2	24	0.0110	0.0169	0.0486	237	0.0037	0.0004	0.0294	243	0.0037	0.0000	0.0293	111	0.0043	0.0008	0.0278
	BAS-RSO	25.1	65	0.0337	0.0168	0.0474	70	0.0318	0.0171	0.0467	298	0.1071	0.0000	0.0308	118	0.0540	0.0055	0.0308
	ADV-RSO	69.5	214	0.1674	0.0032	0.0307	274	0.1309	0.0031	0.0298	477	0.3429	0.0000	0.0289	277	0.1711	0.0025	0.0267
	ADV-BI-RSO	69.7	36	0.0571	0.0762	0.1028	110	0.0196	0.0086	0.0363	327	0.1109	0.0000	0.0310	122	0.0272	0.0053	0.0299
sen	BAS-BAG	24.0	188	0.0398	0.0069	0.0249	203	0.0378	0.0074	0.0242	786	0.0759	0.0013	0.0227	266	0.0436	0.0057	0.0216
	BAS-RLO	22.9	90	0.0240	0.0154	0.0383	92	0.0235	0.0155	0.0387	745	0.0921	0.0009	0.0235	147	0.0309	0.0077	0.0235
	ADV-RLO	48.0	440	0.0046	0.0039	0.0239	895	0.0035	0.0023	0.0236	922	0.0038	0.0008	0.0244	636	0.0038	0.0026	0.0218
	ADV-BI-RLO	98.4	41	0.0094	0.0140	0.0392	401	0.0034	0.0022	0.0239	411	0.0036	0.0006	0.0242	187	0.0038	0.0028	0.0221
	BAS-RSO	22.4	92	0.0245	0.0164	0.0402	98	0.0244	0.0151	0.0381	676	0.0921	0.0008	0.0238	150	0.0323	0.0070	0.0245
	ADV-RSO	67.4	257	0.2262	0.0043	0.0239	389	0.1435	0.0052	0.0233	895	0.5728	0.0006	0.0231	339	0.1798	0.0043	0.0220
	ADV-BI-RSO	82.3	52	0.0669	0.0628	0.0800	142	0.0121	0.0207	0.0395	550	0.2289	0.0002	0.0250	162	0.0127	0.0099	0.0267

Table B.28
Statistics of the Pareto front approximations with the global EMO objectives.

		Card.	Best complx				Best diversity				Best train				Best trade-off (tra-div-cmpl)			
			Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst	Cmpl	Div	Tra	Tst
shu	BAS-BAG	3.9	38	0.0003	0.0003	0.0006	38	0.0003	0.0003	0.0006	71	0.0006	0.0002	0.0006	40	0.0003	0.0003	0.0006
	BAS-RLO	8.8	46	0.0005	0.0004	0.0008	51	0.0005	0.0004	0.0008	139	0.0011	0.0002	0.0006	65	0.0006	0.0004	0.0006
	ADV-RLO	37.0	155	0.0003	0.0003	0.0007	291	0.0002	0.0002	0.0006	322	0.0002	0.0002	0.0006	207	0.0002	0.0002	0.0006
	ADV-BI-RLO	24.7	22	0.0003	0.0004	0.0009	119	0.0002	0.0002	0.0007	124	0.0002	0.0002	0.0006	56	0.0002	0.0002	0.0006
	BAS-RSO	8.4	47	0.0005	0.0004	0.0008	53	0.0005	0.0004	0.0008	119	0.0010	0.0002	0.0006	68	0.0006	0.0003	0.0005
	ADV-RSO	33.3	114	0.1293	0.0003	0.0006	135	0.0907	0.0003	0.0006	294	0.5660	0.0002	0.0006	131	0.1256	0.0003	0.0005
	ADV-BI-RSO	27.8	20	0.0800	0.0184	0.0185	76	0.0001	0.0030	0.0035	103	0.0866	0.0002	0.0007	42	0.0104	0.0004	0.0008
spa	BAS-BAG	13.9	192	0.1007	0.0427	0.0680	199	0.0995	0.0426	0.0677	691	0.1551	0.0304	0.0642	223	0.1032	0.0397	0.0657
	BAS-RLO	14.0	96	0.0659	0.0479	0.0808	105	0.0641	0.0493	0.0830	569	0.1517	0.0253	0.0633	146	0.0776	0.0366	0.0643
	ADV-RLO	30.1	408	0.0227	0.0324	0.0655	648	0.0196	0.0280	0.0630	642	0.0204	0.0250	0.0629	526	0.0202	0.0281	0.0604
	ADV-BI-RLO	58.5	39	0.0317	0.0472	0.0830	338	0.0193	0.0280	0.0638	282	0.0201	0.0235	0.0636	180	0.0203	0.0261	0.0612
	BAS-RSO	13.5	116	0.0638	0.0490	0.0859	117	0.0631	0.0500	0.0865	720	0.1584	0.0203	0.0618	182	0.0793	0.0319	0.0687
	ADV-RSO	52.9	170	0.2824	0.0282	0.0638	205	0.2539	0.0297	0.0651	867	1.2860	0.0190	0.0619	239	0.3106	0.0273	0.0606
	ADV-BI-RSO	67.4	63	0.0982	0.0728	0.1081	456	0.0457	0.0302	0.0725	630	0.5219	0.0149	0.0634	202	0.0504	0.0310	0.0685
ste	BAS-BAG	34.2	301	0.3127	0.0749	0.2567	303	0.3118	0.0750	0.2560	1617	0.4945	0.0155	0.2399	331	0.3208	0.0704	0.2484
	BAS-RLO	42.3	130	0.1845	0.1237	0.3147	130	0.1818	0.1241	0.3088	1547	0.4968	0.0111	0.2405	283	0.2729	0.0679	0.2491
	ADV-RLO	76.8	758	0.0315	0.0335	0.2447	1870	0.0243	0.0189	0.2389	1797	0.0256	0.0099	0.2395	1432	0.0253	0.0176	0.2288
	ADV-BI-RLO	135.5	62	0.0689	0.1282	0.3186	1075	0.0241	0.0188	0.2371	928	0.0253	0.0089	0.2380	408	0.0269	0.0197	0.2313
	BAS-RSO	43.9	136	0.1933	0.1248	0.3052	138	0.1924	0.1263	0.3034	1607	0.5102	0.0099	0.2376	239	0.2586	0.0722	0.2538
	ADV-RSO	94.4	440	0.2478	0.0366	0.2437	621	0.1832	0.0314	0.2392	1648	0.4641	0.0087	0.2391	611	0.2077	0.0349	0.2305
	ADV-BI-RSO	142.4	89	0.1803	0.1967	0.3565	579	0.1197	0.0626	0.2669	1105	0.3620	0.0069	0.2413	343	0.1257	0.0590	0.2446
tex	BAS-BAG	25.8	373	0.0965	0.0050	0.0393	381	0.0951	0.0050	0.0390	813	0.1418	0.0000	0.0323	611	0.1217	0.0016	0.0312
	BAS-RLO	30.1	141	0.0444	0.0250	0.0713	142	0.0436	0.0255	0.0718	668	0.1386	0.0000	0.0336	280	0.0731	0.0060	0.0382
	ADV-RLO	89.5	767	0.0059	0.0017	0.0324	2107	0.0041	0.0003	0.0282	2152	0.0041	0.0000	0.0282	1606	0.0044	0.0003	0.0259
	ADV-BI-RLO	119.1	60	0.0128	0.0214	0.0662	1227	0.0041	0.0003	0.0280	1151	0.0041	0.0000	0.0282	494	0.0045	0.0004	0.0277
	BAS-RSO	30.6	147	0.0476	0.0274	0.0721	148	0.0467	0.0279	0.0724	659	0.1420	0.0000	0.0318	315	0.0794	0.0057	0.0381
	ADV-RSO	72.8	532	0.2566	0.0015	0.0312	685	0.1038	0.0015	0.0306	880	0.1514	0.0000	0.0302	749	0.1531	0.0006	0.0277
	ADV-BI-RSO	95.2	91	0.1112	0.0831	0.1195	439	0.0256	0.0062	0.0429	940	0.1051	0.0000	0.0326	333	0.0318	0.0060	0.0392
thy	BAS-BAG	10.6	154	0.0279	0.0129	0.0222	167	0.0275	0.0124	0.0228	443	0.0441	0.0077	0.0221	167	0.0283	0.0123	0.0218
	BAS-RLO	11.4	78	0.0201	0.0109	0.0273	80	0.0193	0.0116	0.0279	251	0.0399	0.0064	0.0242	89	0.0212	0.0110	0.0262
	ADV-RLO	32.3	310	0.0072	0.0089	0.0224	549	0.0061	0.0085	0.0230	449	0.0065	0.0068	0.0226	418	0.0064	0.0088	0.0215
	ADV-BI-RLO	55.8	32	0.0099	0.0111	0.0285	287	0.0059	0.0086	0.0228	193	0.0064	0.0056	0.0233	146	0.0063	0.0073	0.0222
	BAS-RSO	10.6	85	0.0193	0.0102	0.0286	87	0.0191	0.0105	0.0283	287	0.0418	0.0057	0.0232	119	0.0235	0.0099	0.0250
	ADV-RSO	44.0	162	0.0838	0.0115	0.0221	184	0.0707	0.0111	0.0221	697	0.3476	0.0068	0.0222	205	0.0893	0.0106	0.0213
	ADV-BI-RSO	44.0	35	0.0254	0.1369	0.1477	107	0.0127	0.0259	0.0403	263	0.1771	0.0037	0.0238	62	0.0172	0.0140	0.0269
two	BAS-BAG	25.5	337	0.0954	0.0059	0.0415	339	0.0932	0.0065	0.0417	913	0.1692	0.0000	0.0352	488	0.1175	0.0026	0.0353
	BAS-RLO	29.2	121	0.0409	0.0233	0.0611	122	0.0405	0.0235	0.0621	664	0.1472	0.0000	0.0323	273	0.0740	0.0048	0.0344
	ADV-RLO	104.9	836	0.0044	0.0010	0.0317	2481	0.0031	0.0003	0.0274	2415	0.0032	0.0000	0.0277	1932	0.0033	0.0001	0.0259
	ADV-BI-RLO	137.0	65	0.0129	0.0220	0.0626	1269	0.0031	0.0003	0.0274	1209	0.0031	0.0000	0.0276	442	0.0038	0.0002	0.0274
	BAS-RSO	28.5	143	0.0467	0.0246	0.0704	144	0.0463	0.0261	0.0714	711	0.1605	0.0000	0.0348	309	0.0858	0.0064	0.0393
	ADV-RSO	86.1	365	0.1834	0.0022	0.0320	517	0.1221	0.0014	0.0312	1039	0.2921	0.0000	0.0298	577	0.1554	0.0013	0.0282
	ADV-BI-RSO	114.0	81	0.0962	0.0578	0.0989	409	0.0264	0.0166	0.0525	835	0.1124	0.0000	0.0338	372	0.0305	0.0053	0.0390
wan	BAS-BAG	39.1	421	0.2212	0.0236	0.1623	427	0.2188	0.0247	0.1629	1877	0.4111	0.0000	0.1499	539	0.2497	0.0168	0.1528
	BAS-RLO	51.0	177	0.1154	0.0660	0.2012	183	0.1132	0.0670	0.2044	1622	0.4110	0.0000	0.1495	318	0.1748	0.0290	0.1644
	ADV-RLO	131.5	1139	0.0140	0.0064	0.1530	4143	0.0098	0.0006	0.1454	3684	0.0101	0.0000	0.1466	2737	0.0105	0.0009	0.1418
	ADV-BI-RLO	168.5	86	0.0374	0.0676	0.2008	2180	0.0098	0.0009	0.1462	1761	0.0101	0.0000	0.1472	797	0.0112	0.0012	0.1438
	BAS-RSO	50.8	180	0.1197	0.0696	0.2047	181	0.1188	0.0718	0.2049	1641	0.4151	0.0000	0.1499	325	0.1754	0.0303	0.1657
	ADV-RSO	100.8	604	0.2645	0.0062	0.1527	775	0.1913	0.0036	0.1519	1562	0.3660	0.0000	0.1488	826	0.2236	0.0037	0.1462
	ADV-BI-RSO	131.2	120	0.1425	0.0930	0.2183	717	0.0581	0.0110	0.1596	1751	0.2104	0.0000	0.1526	522	0.0590	0.0162	0.1586
wav	BAS-BAG	38.7	489	0.2258	0.0272	0.1614	489	0.2258	0.0272	0.1614	3009	0.4285	0.0010	0.1492	644	0.2548	0.0201	0.1530
	BAS-RLO	54.9	192	0.1138	0.0650	0.1957	193	0.1133	0.0654	0.1970	2777	0.4390	0.0002	0.1473	389	0.1851	0.0280	0.1593
	ADV-RLO	132.0	1132	0.0156	0.0112	0.1534	3694	0.0115	0.0030	0.1473	3561	0.0119	0.0002	0.1464	2566	0.0122	0.0023	0.1425
	ADV-BI-RLO	168.8	111	0.0348	0.0590	0.1908	1991	0.0115	0.0028	0.1482	1904	0.0118	0.0002	0.1486	829	0.0127	0.0044	0.1455
	BAS-RSO	53.5	203	0.1202	0.0640	0.1959	203	0.1202	0.0640	0.1959	2474	0.4348	0.0003	0.1474	354	0.1760	0.0326	0.1602
	ADV-RSO	102.1	614	0.2568	0.0084	0.1512	851	0.1824	0.0078	0.1503	2392	0.5965	0.0002	0.1482	788	0.2153	0.0072	0.1459
	ADV-BI-RSO	135.5	115	0.1717	0.1009	0.2133	839	0.0562	0.0119	0.1552	1842	0.3425	0.0002	0.1523	550	0.0626	0.0200	0.1552
wqu	BAS-BAG	36.1	731	0.3716	0.1938	0.4184	731	0.3707	0.1949	0.4186	4130	0.5045	0.1192	0.3900	950	0.3933	0.1768	0.4049
	BAS-RLO	51.6	326	0.2589	0.2489	0.4619	328	0.2584	0.2519	0.4606	4263	0.5223	0.0883	0.3824	605	0.3277	0.1707	0.4154
	ADV-RLO	97.6	1883	0.0495	0.1252	0.4002	6230	0.0378	0.0962	0.3843	4880	0.0390	0.0873	0.3825	4418	0.0394	0.0939	0.3753
	ADV-BI-RLO	161.3	167	0.1140	0.2435	0.4637	3476	0.0377	0.0952	0.3850	2483	0.0391	0.0851	0.3842	1384	0.0412	0.0945	0.3786
	BAS-RSO	53.8	344	0.2595	0.2469	0.4565	344	0.2591	0.2473	0.4566	5065	0.5339	0.0778	0.3784	717	0.3457	0.1566	0.4075
	ADV-RSO	143.9	1201	0.5524	0.1212	0.3930	1461	0.4049	0.1142	0.3896	5307	1.2924	0.0735	0.3770	1717	0.4914	0.1068	0.3785
	ADV-BI-RSO	189.8	201	0.2518	0.3037	0.5042	1518	0.1709	0.1252	0.4014	3239	0.9560	0.0731	0.3826	917	0.2183	0.1447	0.4003

class imbalanced problems with an appropriate classifier [62] is a very challenging future work.

Acknowledgment

We would like to thank Ludmila I. Kuncheva and Juan José Rodríguez for their kind help with answering all the questions regarding ROs and sharing their source code with us.

This work was supported by the Spanish Ministerio de Economía y Competitividad under Projects TIN2012-38525-C02-01 and TIN2011-24302, including funding from the European Regional Development Fund.

Appendix A. Bagging CEs and RO-based bagging CEs using C4.5 and NB

To make this contribution self-contained we provide the results of bagging CEs and both (RLO and RSO) RO-based bagging CEs using C4.5 and NB (see Tables A.24 and A.25 respectively) as mentioned in Section 3.4.3.

Appendix B. Statistics of the Pareto front approximations with the global objectives

Tables B.26, B.27, and B.28 present the values of three global learning objectives, training error (*Tra*), test error (*Tst*), and complexity (*Cmpl*) considering each solution for each dataset. The cardinality of each Pareto set approximation (*Card.*) is also shown in the three tables.

References

[1] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley, 2004.
[2] K. Trawiński, O. Cordón, A. Quirin, L. Sánchez, Random oracles fuzzy rule-based multiclassifiers for high complexity datasets, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Hyderabad, India, 2013, in press.
[3] L.I. Kuncheva, J.J. Rodríguez, Classifier ensembles with a random linear oracle, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 500–508.
[4] K. Trawiński, O. Cordón, A. Quirin, On designing fuzzy rule-based multiclassification systems by combining FURIA with bagging and feature selection, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 19 (2011) 589–633.
[5] J.C. Hühn, E. Hüllermeier, FURIA: an algorithm for unordered fuzzy rule induction, Data Mining and Knowledge Discovery 19 (2009) 293–319.
[6] J.C. Hühn, E. Hüllermeier, An analysis of the FURIA algorithm for fuzzy rule induction, in: Advances in Machine Learning I, 2010, pp. 321–344.
[7] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[8] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics 28 (1998) 417–425.
[9] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial Intelligence 137 (2002) 239–263.
[10] D. Partridge, W. Yates, Engineering multiversion neural-net systems, Neural Computation 8 (1996) 869–893.
[11] K. Trawiński, A. Quirin, O. Cordón, A study on the use of multi-objective genetic algorithms for classifier selection in FURIA-based fuzzy multiclassifiers, International Journal of Computational Intelligence Systems 5 (2012) 231–253.
[12] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[13] C. Coello, G. Lamont, D.V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, second ed., Springer, 2007.
[14] J.J. Rodríguez, L.I. Kuncheva, Naïve bayes ensembles with a random oracle, Lecture Notes in Computer Science, 4472, Springer-Verlag, 2007, pp. 450–458.
[15] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[16] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2011) 255–287.
[17] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Machine Learning 40 (2000) 139–157.
[18] R. Banfield, L. Hall, K. Bowyer, W. Kegelmeyer, A comparison of decision tree ensemble creation techniques, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 173–180.
[19] D. Optiz, R. Maclin, Popular ensemble methods: an empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.
[20] R. Schapire, The strength of weak learnability, Machine Learning 5 (1990) 197–227.
[21] Z. Zhou, Ensembling local learners through multimodal perturbation, IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics 35 (2005) 725–735.
[22] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in search strategies for ensemble feature selection, Information Fusion 6 (2005) 83–98.
[23] L. Xu, A. Krzyzak, C. Suen, Methods of combining multiple classifiers and their application to handwriting recognition, IEEE Transactions on Systems, Man, and Cybernetics 22 (1992) 418–435.
[24] T. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844.
[25] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.
[26] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1619–1630.
[27] P.P. Bonissone, J.M. Cadenas, M.C. Garrido, R.A. Díaz-Valladares, A fuzzy random forest, International Journal of Approximate Reasoning 51 (2010) 729–747.
[28] H. Takahashi, H. Honda, Lymphoma prognostication from expression profiling using a combination method of boosting and projective adaptive resonance theory, Journal of Chemical Engineering of Japan 39 (2006) 767–771.
[29] J. Canul-Reich, L. Shoemaker, L.O. Hall, Ensembles of fuzzy classifiers, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), London, 2007, pp. 1–6.
[30] C. Marsala, Data mining with ensembles of fuzzy decision trees, in: IEEE Symposium on Computational Intelligence and Data Mining, Nashville (USA), 2009, pp. 348–354.
[31] C.Z. Janikow, Fuzzy decision trees: issues and methods, IEEE Transactions on Systems, Man, and Cybernetics, Part B 28 (1998) 1–14.
[32] J. Aguilera, M. Chica, M. del Jesus, F. Herrera, Niching genetic feature selection algorithms applied to the design of fuzzy rule based classification systems, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), London, 2007, pp. 1794–1799.
[33] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, 1989.
[34] Y. Nojima, H. Ishibuchi, Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion, in: International Conference on Hybrid Intelligent Systems and Conference on Neuro-Computing and Evolving Intelligence, 2006, 4 pp (CD-ROM).
[35] Y. Nojima, H. Ishibuchi, Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design, International Journal of Hybrid Intelligent Systems 4 (2007) 157–169.
[36] H. Ishibuchi, Y. Nojima, Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers, International Journal of Hybrid Intelligent Systems 3 (2006) 129–145.
[37] R. Schapire, Y. Freund, P. Bartlett, W. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, in: International Conference on Machine Learning, Nashville, 1997, pp. 322–330.
[38] B. Dasarathy, B. Sheela, A composite classifier system design: concepts and methodology, Proceedings of IEEE 67 (1979) 708–713.
[39] E. Dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognition 41 (2008) 2993–3009.
[40] O. Cordón, A. Quirin, Comparing two genetic overproduce-and-choose strategies for fuzzy rule-based multiclassification systems generated by bagging and mutual information-based feature selection, International Journal of Hybrid Intelligent Systems 7 (2010) 45–64.
[41] L. Oliveira, M. Morita, R. Sabourin, F. Bortolozzi, Multi-objective genetic algorithms to create ensemble of classifiers, Lecture Notes in Computer Science 3410 (2005) 592–606.
[42] E. Dos Santos, R. Sabourin, P. Maupin, Single and multi-objective genetic algorithms for the selection of ensemble of classifiers, in: International Joint Conference on Neural Networks (IJCNN), Vancouver, 2006, pp. 3070–3077.
[43] D. Ruta, B. Gabrys, Classifier selection for majority voting, Information Fusion 6 (2005) 63–81.
[44] R.R. Yager, D.P. Filev, Essentials of Fuzzy Modeling and Control, Wiley Interscience, New York NY, USA, 1994.
[45] J. Casillas, O. Cordon, F. Herrera, L. Magdalena, Interpretability Issues in Fuzzy Modeling, Springer-Verlag, Berlin, Heidelberg, 2003.
[46] J.M. Alonso, L. Magdalena, G. González-Rodríguez, Looking for a good fuzzy system interpretability index: an experimental approach, International Journal of Approximate Reasoning 51 (2009) 115–134.
[47] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, 2001.
[48] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (2004) 5–31.
[49] F. Herrera, Genetic fuzzy systems: taxonomy current research trends and prospects, Evolutionary Intelligence 1 (2008) 27–46.

Please cite this article in press as: K. Trawiński et al., Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?, Knowl. Based Syst. (2013), http://dx.doi.org/10.1016/j.knosys.2013.08.006

[50] O. Cordón, A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems, International Journal of Approximate Reasoning 52 (2011) 894–913.
[51] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, Fuzzy Sets and Systems 103 (1999) 223–238.
[52] O. Cordón, M. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1999) 21–45.
[53] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 226–238.
[54] T. Dietterich, Approximate statistical test for comparing supervised classification learning algorithms, Neural Computation 10 (1998) 1895–1923.
[55] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[56] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[57] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (2010) 2044–2064.
[58] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems), Morgan Kaufman Publishers Inc., 2005.
[59] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling With Linguistic Information Granules, Springer, 2005.
[60] K. Trawiński, O. Cordón, L. Sánchez, A. Quirin, A genetic fuzzy linguistic combination method for fuzzy rule-based multiclassifiers, IEEE Transactions on Fuzzy Systems (2013). in press.
[61] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2003) 181–207.
[62] V. López, A. Fernández, M.J. del Jesus, F. Herrera, A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets, Knowledge-Based Systems 38 (2013) 85–104. Special Issue on Advances in Fuzzy Knowledge Systems: Theory and Application..

5 A Multiclassifier Approach for Topology-based WiFi Indoor Localization

The work carried out for the real-world problem is described in:

- K. Trawiński, J. M. Alonso, and N. Hernandez. A Multiclassifier Approach for Topology-based WiFi Indoor Localization, Soft Computing, vol. 17, no 10, pp. 1817-1831, 2013. DOI 10.1007/s00500-013-1019-5.
  - State: Published.
  - Impact Factor (JCR 2012): 1.124.
  - Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE. Order: 63/115. Q3.

Soft Comput (2013) 17:1817–1831
DOI 10.1007/s00500-013-1019-5

METHODOLOGIES AND APPLICATION

A multiclassifier approach for topology-based WiFi indoor localization

Krzysztof Trawiński · Jose M. Alonso · Noelia Hernández

Published online: 26 February 2013

Abstract People localization is required for many novel applications like for instance proactive care for the elders or people suffering degenerative dementia such as Alzheimer's disease. This paper introduces a new system for people localization in indoor environments. It is based on a topology-based WiFi signal strength fingerprint approach. Accordingly, it is a robust, cheap, ubiquitous and non-intrusive system which does require neither the installation of extra hardware nor prior knowledge about the structure of the environment under consideration. The well-known curse of dimensionality critically emerges when dealing with complex environments. The localization task turns into a high dimensional classification task. Therefore, the core of the proposed framework is a fuzzy rule-based multiclassification system, using standard methodologies for the component classifier generation such as bagging and random subspace, along with fuzzy logic to deal with the huge uncertainty that is characteristic of WiFi signals. Achieved results in two real environments are encouraging, since they clearly overcome those ones provided by the well-known nearest neighbor fingerprint matching algorithm, which is usually considered as a baseline for WiFi localization.

Keywords WiFi localization · Classifier ensembles · Bagging · Random subspace · Fuzzy rule-based multiclassification systems

1 Introduction

Recently, there has been a wide proliferation of smartphones. They can be seen as small computers equipped with GSM/UMTS, GPS, WiFi, blootooth, infrared, accelerometers, cameras, and so on (Palmer et al. 2012). As a result, everyday there are more and more novel applications which are able of exploiting successfully the localization capabilities of smartphones. Telecommunication companies are interested in providing personalized advertisements and/or context-aware information services. Thus, for instance once user location is discovered he/she can be guided towards the closest restaurant/shop which better fits his/her preferences. In addition, in the context of social networks, one user may be interested in receiving some kind of a notification when some friends were close to him/her. On the other hand, there are also challenging medical applications where people localization is required, such as proactive care of elders or disabled people.

Of course, several ethical and controversial issues usually arise when dealing with localization applications, mainly regarding tasks related to tracking people (Paul et al. 2012). Users want to receive useful information related to their current location but they also want to preserve their own privacy. Therefore, systems based on video cameras are usually rejected. All in all, localization

Communicated by G. Acampora.

K. Trawiński · J. M. Alonso (✉)
European Centre for Soft Computing, 33600 Mieres, Asturias, Spain
e-mail: jose.alonso@softcomputing.es

K. Trawiński
e-mail: krzysztof.trawinski@softcomputing.es

N. Hernández
RobeSafe Research Group, Department of Electronics, University of Alcalá, 28871 Alcalá de Henares, Madrid, Spain
e-mail: noelia.hernandez@depeca.uah.es

systems are demanded to be non-intrusive, ubiquitous and cheap (Garcia-Valverde et al. 2012).

In the case of outdoor environments, there is a wide range of applications based on GPS (Enge and Misra 1999). Most of them are related to navigation assistance, e.g., an intelligent program is able to find out the optimal route (according to the user's preferences) in a predefined map between the current location and the desired destination (Chiang and Huang 2008). Unfortunately, GPS does not work properly in indoor environments where many interesting applications arise.

Regarding indoor localization, there are lots of related proposals which are based on different technologies: infrared, computer vision, ultrasound, laser, radio frequency, cellular communication, etc. Anyway, the so-called signal strength approaches are very attractive because they can be applied to wireless networks without needing an additional specific hardware (Elnahrawy et al. 2004). Moreover, WiFi localization systems arise as the most promising choice thanks to their quickly growing degree of the coverage. Nowadays, there are WiFi Access Points (APs) in most of the public buildings like hospitals, libraries, universities, museums, etc. In addition, measuring the WiFi signal strength level (without transmitting-receiving any data) is free of charge even for private WiFi networks. In fact, some works have already presented experiments showing the suitablity of combining both GPS (outdoor environments) and WiFi (indoor environments) with the aim of designing global and ubiquitous localization systems (Gallagher et al. 2010).

This work focuses on people localization in indoor environments. That is the reason why we have opted for a topology-based localization system. Actually, we have enhanced the framework for WiFi localization sketched in (Menendez et al. 2011). It was firstly introduced in (Alonso et al. 2009) and then successfully integrated in the people positioning module presented in (Alvarez-Alvarez et al. 2010). Such a system is based on fingerprinting, i.e., matching the current signal strength measures with those ones previously stored into a database that represents a WiFi map of the experimental environment. Then, the database is split into training and test datasets. They are used to generate and validate an intelligent classifier able to estimate the closest reference location (one out of the predefined set of significant locations considered when building the WiFi map) to the actual user location according to the measured signal strength.

Notice that, the main novelty of this work comes from the fact that the new system is implemented in the form of a multiclassification system (MCS) instead of adopting a single-classifier strategy like we had considered in our previous works, with the aim of producing a system as accurate as possible. Furthermore, the proposal has been tested and validated into two different real-world environments: (1) A small trial scenario considering only one corridor at the European Centre for Soft Computing (ECSC) premises; and (2) a much more complex scenario considering the second and the third floors of the Polytechnic School at the University of Alcalá (UAH).

The rest of the manuscript is organized as follows. The next section gives an overview on some preliminary works, regarding both WiFi localization systems and MCSs. Then, Sect. 3 describes thoroughly the new proposed MCS-based WiFi indoor localization system. Section 4 presents the experimental framework along with a detailed analysis of the achieved results. Finally, Sect. 5 highlights the main conclusions but also points out some challenging future works.

2 Preliminaries

2.1 WiFi localization and soft computing

WiFi localization systems make use of 802.11 b/g network infrastructure to estimate the position of a receiver. Thus, the location of a person can be discovered by pointing out the estimated location of one of its electronic devices, for instance a smartphone equipped with a WiFi interface.

WiFi technology works at 2.4 GHz, which is close to the water resonant frequency, therefore the received signal strength (RSS) from each AP visible in the surrounding environment directly depends on the distance between the emitter and the receiver, but it also depends on the presence of obstacles and/or people placed between them. Notice that, human body absorbs part of the electromagnetic waves it is exposed to (Bahillo et al. 2009). Thus, people may become a non-negligible source of interference, dimming RSS.

RSS is affected by many variations, namely temporal, small-scale and large-scale variations (Youssef and Agrawala 2003). They introduce a lot of uncertainty in the system that is difficult to deal with. For instance, small-scale variations take place when one electronic device equipped with WiFi moves in a small distance (under the wavelength $\lambda = 12.5$ cm). In such circumstances, RSS can strongly vary up to 10 dBm what would be equivalent to move around 6 m in the context of large-scale variations (Alonso et al. 2011). As a result, it is not straightforward to estimate the correct device location. Uncertainty must be properly handled in order to determine if the observed changes in RSS are due to small-scale variations (only a few centimeters away) or they are due to large-scale variations (a few meters away).

WiFi localization systems exploit the path loss propagation model due to large-scale variations of WiFi signal to determine how close the receiver is to a certain AP. Unfortunately, in indoor environments, RSS is also strongly affected by reflection, refraction and diffraction, what is commonly known as the multipath effect (Elnahrawy et al. 2004). As a result, RSS becomes a complex function of the distance that dynamically changes with time. Therefore, formalizing propagation models adapted to the specific characteristics of each indoor environment, in each time instant, is not reliable.

Soft Computing is usually defined as a family of techniques (Fuzzy Logic, Neuro-computing, Probabilistic Reasoning, Evolutionary Computation, and their hybridizations), well suited for coping with imprecision and uncertainty (Magdalena 2008). Thus, an extensive research has been done on wireless localization based on Soft Computing techniques over the last decade. For instance, Soft Computing techniques are able to deal properly with small-scale variations in wireless sensors (Alonso et al. 2011).

Yun et al. (2009) proposed a Soft Computing based localization system for outdoor environments. To start with, they generated a genetic fuzzy system for individual localization where the edge weights of each anchor node were firstly modeled by a fuzzy system and then optimized by a genetic algorithm. Later, they generated a neural network for overall localization. Achieved results were promising and they proved the suitability of considering Soft Computing approaches to deal with the wireless localization.

Nerguizian et al. (2004) proposed the use of neural networks and fingerprinting to deal with the well-known multipath effect in indoor environments. They addressed mobile robot location based on a distance-based (also known as metric or Cartesian) approach, which is normally adopted in Robotics where localization is made in a low abstraction level with the aim of estimating X–Y coordinates. The network learning was done off-line but it could become computational costly (and even unfeasible) for very large environments. More recently, Outemzabet et al. presented an alternative location system also based on neural networks and fingerprinting. The main novelty arose from the fact that the estimated X–Y position was enhanced first with Kalman filtering (Outemzabet and Nerguizian 2008) and later with particle filtering and a low-cost sensor (Outemzabet and Nerguizian 2008).

Dharne et al. (2006) advocated for the use of fuzzy logic. They proposed a fuzzy rule-based system able to yield good results thanks to the use of a grid-based map describing the environment under consideration. Moreover, they reduced the computational cost by taking into account only significant grid-points. Hence, they followed a topology-based approach instead of a distance-based one. Topology-based systems carry out a more human friendly symbolic localization, which is made in a higher abstraction level. Their goal is not to find out the exact X–Y coordinates but to give an approximate position (e.g., at the room level) with a high confidence. This approach is preferred when dealing with most applications supported by people localization. Notice that, fuzzy logic is especially useful to handle problems where the available information is vague, which is the typical situation when working with WiFi signal strength sensors (Astrain et al. 2006). Fuzzy systems have proved to be effective for topology-based WiFi indoor localization in the context of people localization in ambient intelligent environments (Garcia-Valverde et al. 2012) but also in the context of Robotics (Herrero-Pereza et al. 2010).

Finally, topology-based indoor localization has already been addressed as a classification problem through Soft Computing techniques (Alonso et al. 2011), however a need for improving the accuracy (while maintaining a low execution time) in this kind of applications requires an advanced tool being able to cope with these challenges. Thus, in the current contribution, the MCS approach, which is a well known to obtain high performance (better than a single classifier) (Kuncheva 2004), will be applied. Up to our knowledge, no work has been done related to MCSs in the context of topology-based WiFi fingerprint indoor localization. Notice that, the well-known curse of dimensionality critically emerges when dealing with complex real-world environments. In consequence, the localization problem turns into a high dimensional classification task. Such kind of problem only can be addressed effectively through an MCS approach.

2.2 Multiclassification systems

In the last decade, MCSs, also called multiclassifiers or classifier ensembles, have arisen as very powerful tools when dealing with complex, high dimensional classification problems, because they are able to yield higher performance than any of their single classifiers (Kuncheva 2004). As a result, this research topic has become especially active inside the machine learning community in general, and the Soft Computing community in particular, considering decision trees (machine learning) or neural networks (Soft Computing) to generate the component classifiers. The interested reader is referred to (Banfield et al. 2007; Optiz and Maclin 1999) where he/she can find two surveys for the case of decision trees (both) and neural network ensembles (the latter), including exhaustive experimental studies. More recently, some works have also considered the use of fuzzy classifiers (Soft Computing) (Bonissone et al. 2010; Trawiński et al. 2011).

An individual classifier must provide different patterns of generalization in order to obtain a diverse set of classifiers composing a highly accurate ensemble (Kuncheva 2004). Otherwise, the ensemble would be composed of the same (or similar classifiers) and it would be only as accurate as the best single classifier. Thus, generating diverse component classifiers is fundamental to obtain highly accurate MCSs (Tsymbal et al. 2005). There are different ways to face the design of classifier ensembles.

On the one hand, there is a classical group of approaches considering data resampling with the aim of generating different training sets to derive each individual classifier. Firstly, in the bagging approach (Breiman 1996), the base classifiers are independently learnt from previously resampled training sets ("bags"), which are randomly selected with replacement from the original training data set. Secondly, boosting methods (Schapire 1990) 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, this resampling process gives a higher selection probability to the incorrectly predicted examples by the previous classifiers.

On the other hand, a second group is comprised by a more diverse set of approaches which induct the individual classifier diversity through some alternative ways, different from resampling (Zhou 2005). Feature selection plays a key role in many of them where each classifier is derived by considering different subsets of the original features (Tsymbal et al. 2005). Random subspace (Ho 1998), where each feature subset is randomly generated, is likely to be the most representative method of this kind.

Interestingly, it turns out that a combination between bagging and feature selection yields a generic approach well suited for designing robust and accurate MCSs, no matter the chosen classifier learning method (Panov and Džeroski 2007). We have already tested the combination of bagging and feature selection in the form of fuzzy MCS composition designs (Trawiński et al. 2011). All in all, we drew the conclusion that bagging combined with a base homogeneous fuzzy classifier, which directly incorporated the feature selection ability, was a very powerful tool for dealing with high dimensional classification problems.

3 An MCS-based framework for scalable WiFi indoor localization

This section details the proposed framework for topology-based indoor localization. The main goal of this work is to obtain a scalable and accurate localization system, which can estimate the closest reference location to the actual user location using RSS in a relatively short time. Our proposal is based on an MCS approach. Two different methodologies, bagging (Breiman 1996) and bagging combined with random subspace (Panov and Džeroski 2007), are exploited to design the final MCS-based localization system. First, the base classifiers are learnt off-line from a fingerprint database previously generated, and then the MCS-based framework is run on-line.

3.1 The base classifiers

Two types of base classifiers are considered in order to derive the component classifiers: (1) A simple decision tree, J48G (Webb 1999); and (2) a more advanced Soft Computing algorithm called Fuzzy Unordered Rule Induction Algorithm (FURIA) (Hühn and Hüllermeier 2009).

3.1.1 J48G

J48G is a version of the well-known standard C4.5 (called J48 in Weka[1] (Witten et al. 2011)) decision tree, extended by means of grafting. As explained in (Webb 1999), grafting is a post-processing algorithm applied to an already generated decision tree, which aims at reducing prediction error. It starts with looking for those regions in the feature space which are either empty or they only include misclassified examples. Then, grafting searches for an alternative branch (e.g. generated from the ancestor to the leaf related to the identified region), which is added to the current tree, only in the case that its support for classification of that region is stronger than the already generated one by the initial decision tree generation method.

In particular, J48G considers grafting based on all-tests-but-one-partition (ATBOP), a metric to estimate the accuracy of a potential new leaf, initially proposed by Quinlan (1991). This algorithm assigns one set of training data only to each leaf of the initial decision tree (the set of examples that fails no more than one test on the path to the leaf). Thanks to grafting based on ATBOP, J48G is able to reduce the tree complexity, thus speeding up the induction process while also increasing accuracy.

The interested reader is kindly referred to (Webb 1999) for a full description of J48G.

3.1.2 FURIA

FURIA (Hühn and Hüllermeier 2009) is an extended and enhanced version of the state-of-the-art rule learning algorithm called RIPPER (Cohen 1995), keeping its main advantages such as simplicity and comprehensibility but also introducing new features. We would like to highlight three main extensions of RIPPER provided by FURIA:

1. It defines fuzzy rules instead of crisp ones. The final form of a rule R_j is the following:

[1] We use the implementation of J48G provided by Weka, a software tool for data mining which is freely available at http://www.cs.waikato.ac.nz/ml/weka/.

R_j : **If** x_1 is A_{j1} AND ... AND x_n is A_{jn}
Then Class is C_j with CD_j, $j = 1, 2, \ldots, c$

c is the number of classes. The consequent part points out a class C_j along with its related certainty degree CD_j. Given an example $x = \{x_1, \ldots, x_n\}$, the certainty degree is defined as follows:

$$CD_j = \frac{2\frac{D_T^{C_j}}{D_T} + \sum_{x \in D_T^{C_j}} \mu_r^{C_j}(x)}{2 + \sum_{x \in D_T} \mu_r^{C_j}(x)} \qquad (1)$$

where D_T and $D_T^{C_j}$ stand for the training set and a subset of the training set related to the class C_j respectively. In this approach, each fuzzy rule makes a vote for its consequent class. The vote strength of a rule is calculated as the product of its firing degree $\mu_r^{C_j}(x)$ and its certainty degree CD_j. Hence, the fuzzy reasoning method used is the so-called voting-based method (Cordón et al. 1999).

2. It uses unordered rule sets instead of rule lists. This change omits a bias caused by the default class rule, which is applied whenever there is an uncovered example detected. Unfortunately, the unordered rule set introduces one crucial drawback too, there might appear some cases when given examples are not covered.
3. It proposes a novel rule stretching method in order to manage uncovered examples. To deal with such undesired situations, each rule can be dynamically generalized by removing some of its antecedents. The information measure is proposed to find out which rule to "stretch" for each specific case.

Let us emphasize that FURIA inherits an internal feature selection algorithm from RIPPER. This characteristic in combination with the "soft boundaries" provided by fuzzy rules makes FURIA well-endowed with the ability to deal with noisy, complex, and non-linear high dimensional classification problems.

The interested reader is kindly referred to (Hühn and Hüllermeier 2009) for a full description of FURIA.

3.2 The MCS design approach

In this work we will consider two standard MCS methodologies: bagging (Breiman 1996) and a combination of bagging with random subspace (Ho 1998), as proposed in (Panov and Džeroski 2007) (both methodologies were already used in (Trawiński et al. 2011), as well as random subspace only, however the latter did not bring any performance improvement).

The term bagging is an acronym of bootstrap aggregation and refers to the first successful method proposed to generate MCSs (Breiman 1996). This approach was originally designed for decision tree-based classifiers. However, it represents a very generic approach (its applicability fits to any type of base model either for classification or regression problems). The core of 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. Then, the final output of the model is the most frequent value, called voting, of the learners considered. As a result, bagging is the most effective approach when dealing with unstable classifiers, which means a small change in the training set can cause a significant change in the final model. Furthermore, bagging provides another main advantage, namely it makes feasible parallel and independent learning of the classifiers among the ensemble. In consequence, it is time efficient (due to its inherent parallelism) and quite accurate (Trawiński et al. 2011). In this contribution, the bags are generated with the same size as the original training set, as commonly done.

As said before, random subspace (Ho 1998) is a generic methodology to induce diversity in the generation of the base classifiers. In this approach, a set of features is randomly selected from the original dataset. It is also a well-known approach in the multiclassifiers research field for feature selection (Bonissone et al. 2010; Breiman 2001; Dietterich 2000). Additionally, in Panov and Džeroski 2007 it was shown that the combination between bagging and random subspace results in a general design procedure, usually leading to good MCS designs (regardless the classifier structure considered).

A flow of our design is as follows. The training set is submitted to an instance selection procedure, and (optionally) to a feature selection procedure, in order to provide individual training sets (bags) to train the base classifiers (in off-line mode).

The combination of classifier members within the ensemble (on-line mode), is made by the so-called classifier fusion method (Woods et al. 1997), which aggregates the results provided by the set of component classifiers to calculate the final output, assuming that all classifiers are trained over the entire feature space. The Decision Profile (DP) represents the outputs of all the classifiers in the ensemble (Kuncheva 2001; Kuncheva et al. 2001):

$$DP(\mathbf{x}) = \begin{pmatrix} D_1(\mathbf{x}) \\ \vdots \\ D_L(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} d_{1,1}(\mathbf{x}) & \cdots & d_{1,c}(\mathbf{x}) \\ \vdots & & \vdots \\ d_{L,1}(\mathbf{x}) & \cdots & d_{L,c}(\mathbf{x}) \end{pmatrix} \qquad (2)$$

where c is the number of classes; L is the number of classifiers; and $d_{i,j}(\mathbf{x})$ are the confidence degrees for the classes given an example $\mathbf{x}$. Considering L classifiers, the combined output is usually computed by an algebraic function (Kittler et al. 1998; Kuncheva 2002) such as maximum, minimum, product, mean, median, etc.

3.3 The proposal of an MCS-based framework to deal with noisy WiFi signals

To deal with the inherent noise that characterizes the WiFi signal in indoor environments, we propose an elaborated framework encapsulating an MCS in order to improve the robustness of the whole system. Figure 1 depicts a global schema of the proposed framework, which is made up of the three following phases:

- *Phase1—Classification process of each classifier component*. In this phase the classification task of each MCS components is carried out. Each classifier for each instance from 1 to N outputs confidence degrees d_{ij}^{m} for each class. Thus, the N matrices, namely DPs, are generated to be provided as the input required for the Phase 2.
- *Phase2— Filtering (Aggregation 1)*. The filtering phase takes place at the classifier output level. The confidence degrees d_{ij}^{m} of N instances are aggregated for each classifier d_{ij}^{*}. The aggregation is done by means of one of the (algebraic) functions mentioned in Sect. 3.2. Then, the aggregated DP of the MCS is provided as an output. Notice that, the filtering follows the "moving average" fashion, in every step DP of the next example is included ($x_{N+1}^{'}$ in the example from Fig. 1) in the aggregation of DPs, while excluding the first DP appeared in the given period of time ($x_{1}^{'}$ in the example from Fig. 1).
- *Phase3—Classifier fusion (Aggregation 2)*. In the last phase, a second aggregation is performed. The aggregated DP is combined by means of one of the abovementioned algebraic functions (mean, median, etc.). As a result, the outputs of all the individual classifiers d_{ij}^{*} are merged into one final decision c'.

It is worth noting that, the framework described above is only applicable for the on-line execution mode of our WiFi location system, while the core of the MCS is trained in an off-line mode, starting from a fingerprint database previously generated. Of course, the off-line learning process is not included in the figure.

4 Experimentation

4.1 Experimental setup

In this contribution we present results achieved in two different real-world experimental environments. In the first experiment, just for illustrative purposes, data were gathered in a simple scenario considering only one corridor of the ECSC premises, where all analyzed locations are placed in a straight line. In the second scenario, which is much more realistic and hence complicated, data were gathered in the second and third floors of the Polytechnic School at the UAH. In both cases, experimental data were gathered under usual working conditions.

4.1.1 Scenario 1. A simple but highly illustrative case

The first experimental environment is shown in Fig. 2. It consists of an indoor corridor, about 35 m long, where 13 different positions (Pi), in the distance of 2 m each, were

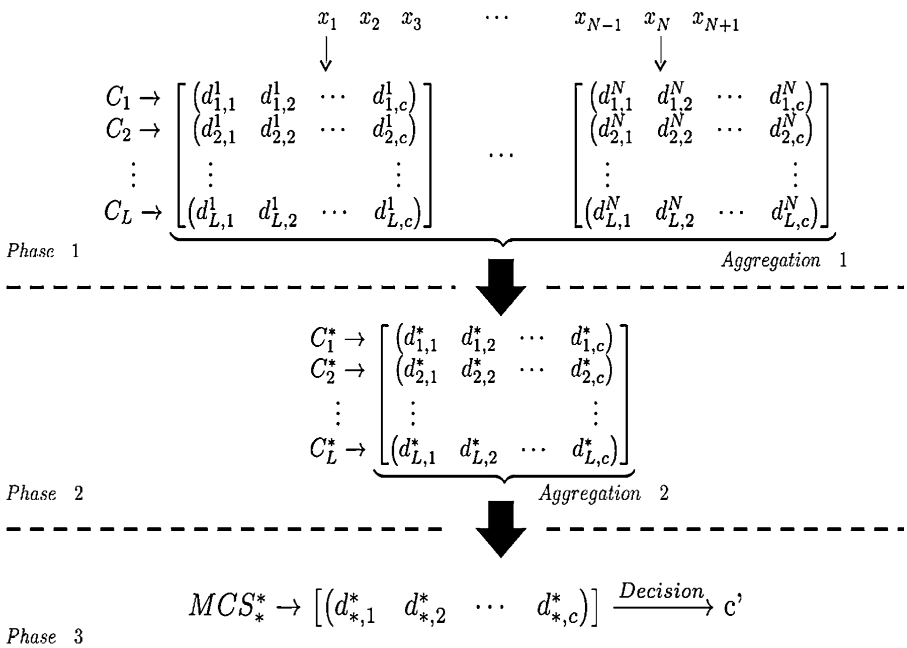


Fig. 1 The proposed framework to tackle with noisy WiFi signals

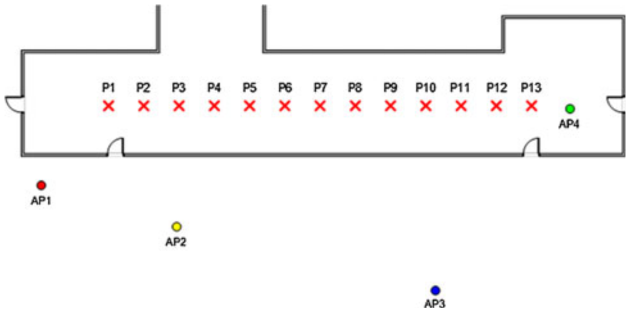


Fig. 2 ECSC test-bed environment (Scenario 1). Four APs and thirteen positions, Pi, (separated by 2 m distance) in a straight line

identified with the aim of being properly recognized by the proposed localization system. The signal strength from 4 APs were measured at each position. These APs were chosen due to their visibility in the entire test-bed environment, exhibiting good patterns of the signal transmission. Although their exact locations are not known, their approximate positions are plotted in the figure.

Data acquisition was made as follows. At each position, Pi, 600 samples of the signal level from each AP were measured and saved in a file. This process was carried out during five different days at different hours. Thus, we obtained a dataset containing 39,000 signal samples. Notice that, since we considered only 13 different positions placed in the same part of the building, namely a corridor, it was feasible to obtain so many samples per position. Considering the WiFi localization task as a classification problem, we built a dataset containing 39000 instances with 4 features. Each set of RSS coming at once from all visible APs corresponds to an instance, whereas the signal from each AP defines a feature of that instance.

The data distribution along the five days for each position and for each AP is shown in Fig. 3. It is represented in the form of boxplots. It can be easily observed how the signal obtained varies for each AP differently, when changing the position. For instance, in positions P1 and P2 the highest power level corresponds to AP1, but in positions P12 and P13 the highest level comes from AP4, due to its proximity. Unfortunately, the variation of RSS is not lineal with physical distance.

Furthermore, in this work, we take an advantage of the time-dependent characteristics of the data, namely the signals of APs are obtained in a consecutive order for each location. In our scenario, the user stops for a few seconds to

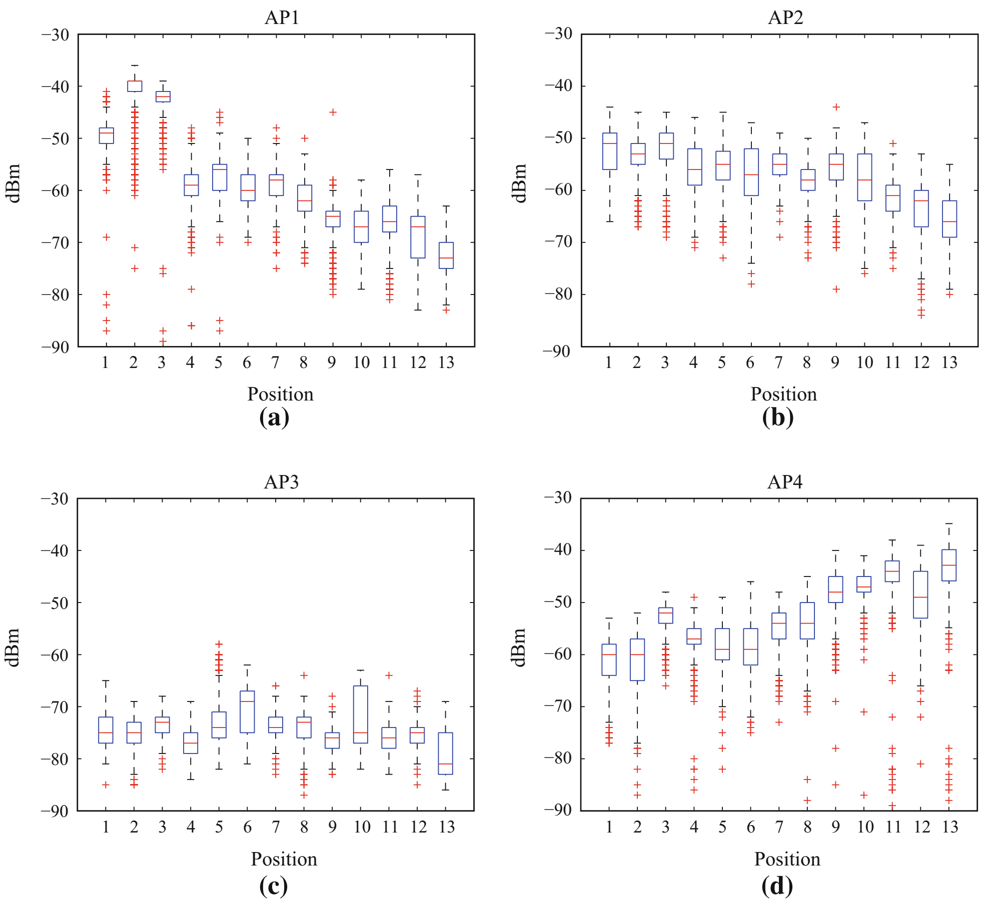


Fig. 3 Data distribution (five different days) for all positions and access points: AP1 (**a**); AP2 (**b**); AP3 (**c**); and AP4 (**d**)

acquire several consecutive WiFi measures with the aim of getting better estimation of its current position. In order to avoid any loss or distortion of the data, for the identification of a given position we consider a consecutive block of samples of size 300 (half of measured samples of each Pi during each day).

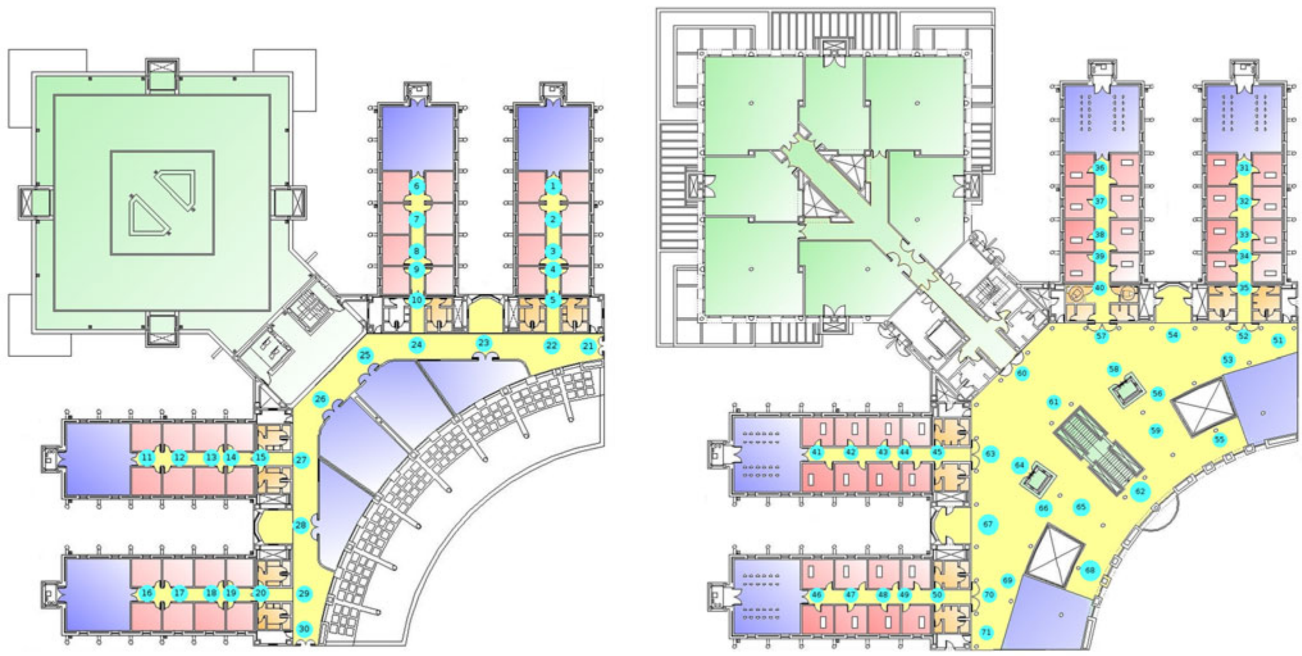
Thus, we have defined a special kind of cross validation inspired by the Diettrich's cross-validation (Dietterich 2000) with the aim of considering the time-dependent characteristics of the data. The measurements for each location and for each experimentation day were divided into 2 blocks of 300 subsequent samples. As we collected data among 5 different days, we got 10 blocks for each location. Blocks for each day are uniformly assigned to the training and test sets in a way that always one block from a given day is assigned to training set as well as to test set. It results with 32 different combinations of the blocks, which can be seen as a particular adaptation of the 16×2 cross-validation. The averaged results are reported.

4.1.2 Scenario 2. A realistic high dimensional case

The second experiment was carried out in the second and the third floors on the west sector of the Polytechnic School at the UAH (see Fig. 4). The experimental environment covers a total surface of 2,400 m^2 with over 100 APs. We have considered 71 significant positions (30 in the third floor and 41 in the second floor), each placed several meters (between 2.5 and 9.5 m) apart from the nearest neighbor position to be recognized.

It is important to remark that contrary to the case of the previous scenario where a few APs were selected for localization purposes, in the second scenario over one-hundred APs were actually deployed with the aim of providing a good Internet access to the largest number of students. Notice that, data coming from all detected wireless devices were considered as APs by our localization system, with no prior knowledge about neither their relevance (they may be just a laptop or smartphone instead of a real AP) nor their physical location. This is a very important issue, since data acquisition is very quick but the number of APs is huge and it may differ among the days of the data acquisition. Moreover, no data pre-processing is required, the off-line learning stage is in charge of discovering the most significant APs for localization purposes.

Data acquisition for training and test was made in two independent weeks. In fact, collecting data for all the 71 positions was made in discontinuous periods of time, at different hours, all along each week considering both mornings and afternoons. At each position, Pi, 60 samples of the signal level from each visible AP were measured and saved in a file. Thus, we built two datasets (one per week) containing 4,260 signal samples each. During the first week we detected 143 different APs, while in the second week there were only 134 APs. Of course, during the test stage only those APs that were also visible in the training



(a) Second floor **(b)** Third floor

Fig. 4 UAH test-bed environment (Scenario 2). One-hundred APs (which actual location is unknown) and seventy-one different positions (represented by *circles*): 30 positions in the second floor (**a**) and 41 positions in the third floor (**b**) stage are taken into account. Again, like in the case of Scenario 1, the WiFi localization task is addressed as a classification problem. Thus, we built a dataset containing 8,520 instances (all sets of RSS coming at once from all visible APs) with 143 (134 in the case of looking at the second experimental week) features (all visible APs).

In a first experiment, the entire dataset obtained from the first week was taken as a training set, while the entire dataset obtained during the second week was considered as a test set. Then, we inverted the procedure, the dataset related to the second week was assigned to a training set and the dataset corresponding to the first week was taken as a test set. Due to the large instability of the random subspace algorithm, we repeat the whole procedure 10 times with different seeds (20 in total). Of course, average of both is reported as the final result.

Notice that, due to the large number of positions analyzed in the different floors of UAH, obtaining such a huge number of data samples like in the previous scenario was not feasible because it would have become very costly and time consuming, thus we opted for reducing the number of collected instances from 600 to 60 at each position. As a result, the total number of instances drops from 39,000 to 8,520. However, the complexity, and scalability, of the new scenario increases significantly in terms of the number of positions (from 13 to 71) and the number of APs (from 4 to over 100), yielding a much more difficult high dimensional classification problem.

4.2 Analysis of the results

In our framework, the localization problem is defined in terms of a classification problem with:

- 4 features considering the 4 visible APs and 13 classes (one per position to be identified) for the first scenario.
- 143 (134 in the case of looking at the second experimental week) features considering all visible APs and 71 classes (one per position to be identified) for Scenario 2.

Our design (J48G MCSs, FURIA MCSs) is composed of 10 classifiers (because in some preliminary trials on Scenario 1, we observed that considering a larger number of classifiers was not yielding a significant increase of accuracy for the analyzed problem), while random subspace (RS) selects a subset of features containing 25, 50, or 75 % of the initial feature set. It is compared with the state-of-the-art classification technique k-Nearest Neighbor (k-NN) (Cover and Hart 1967). For each test sample, k-NN algorithm calculates the Euclidean distances between it and every training sample in the database. Then, it ranks them and takes the k smallest ones. A majority vote between selected samples decides which class the test sample belongs to. We have chosen the standard value $k = 1$, as it is done in (Gallagher et al. 2010). Notice that, 1-NN is usually considered as base line for comparisons since the Bahl's pioneer paper (Bahl and Padmanabhan 2000) where a WiFi localization system based on the use of the Nearest Neighbor algorithm with fingerprinting (a priori radio map) was first proposed. Such a system comprised the same two main stages, namely off-line and on-line stages, as in our proposal. The main difference comes up with the fact that our off-line stage includes not only the generation of the fingerprint database (what was called training stage by Bahl) but also the learning of classifier models. Of course, in the case of 1-NN there is not any explicit classifier model. On the contrary, each new WiFi measure is directly compared against the previously stored radio map in order to determine the right location during the estimation stage.

In addition, we chose mean as the most common aggregation method in both stages (Aggr1 and Aggr2). Notice that the second aggregation stage (Aggr2) only takes place in the case of the designed MCSs, where the 10 individual classifier outputs are fused. The final decision is done using the maximum activation degree. Notice that, with the structure of basic k-NN method Aggr2 makes no sense.

Several values of the block size N were selected, i.e. 1, 4, 7, and 10 corresponding to around 1, 4, 7, and 10 s respectively because our WiFi acquisition frequency was 1 Hz. It is worth mentioning that time is strongly affected by the number of WiFi measures used for a single evaluation, since the time for collecting data samples (in the range of seconds) is much longer than the time required for inferring the current position (in the range of milliseconds).

4.2.1 Results in a simple case (Scenario 1)

In this section we analyze the results obtained for the first scenario. The goal is to check the proposed framework in the context of a rather simple case as the one defined in the selected corridor of the ECSC premises. Table 1 reports the achieved results in terms of accuracy for all the selected block sizes.

The first column presents all the algorithms used, namely 1-NN, J48G, FURIA and several variants of J48G MCSs as well as FURIA MCSs. These variants include bagging (Bag), bagging and RS with 25 % (Bag + RS (25 %)), bagging and RS with 50 % (Bag + RS (50 %)), and bagging and RS with 75 % (Bag + RS (75 %)). In total, 11 algorithms are evaluated in the experiments. The next four columns present accuracy results obtained for block size equal to 1 (N1), 4 (N4), 7 (N7), and 10 (N10) respectively. The best result for each given Ni is highlighted in bold font.

Table 1 Accuracy results for different classification and aggregation methods in Scenario 1 (ECSC environment)

Algorithm	N1	N4	N7	N10
1-NN	**0.893**	0.915	0.929	0.936
J48G	0.836	0.882	0.908	0.923
FURIA	0.758	0.791	0.820	0.846
J48G MCSs				
Bag	0.863	**0.916**	**0.932**	**0.940**
Bag + RS (75 %)	0.838	0.898	0.919	0.931
Bag + RS (50 %)	0.762	0.832	0.860	0.877
Bag + RS (25 %)	0.630	0.692	0.725	0.749
FURIA MCSs				
Bag	0.814	0.868	0.894	0.912
Bag + RS (75 %)	0.744	0.799	0.825	0.844
Bag + RS (50 %)	0.645	0.695	0.718	0.734
Bag + RS (25 %)	0.494	0.530	0.549	0.561

In the light of this table, it can be noticed that "J48G MCS Bag" (N10) outperforms the other approaches in overall (0.940). However, "FURIA MCS Bag" (N10) as well as 1-NN (N10) are able to produce very good results too, 0.912 and 0.936 respectively. As expected, most of the evaluated algorithms were able to achieve very high accuracy (over 84 % in 8 out of 11 algorithms, regarding N10). Notice that 1-NN outperforms both single classifiers, J48G as well as FURIA, no matter the selected block size (Ni).

There are some additional interesting issues to be remarked:

- Both J48G MCSs and (especially) FURIA MCSs get more instable when combined with Bag + RS than with Bag only. "FURIA MCSs Bag + RS (25 %)" obtains the worst result (0.494) for N1.
- MCSs obtain better accuracy results, when increasing the number of features selected by RS (look at the percentage in brackets).
- Considering the block size parameter (Ni), it seems to be somehow correlated with the accuracy. The larger the block size is, the higher accuracy is obtained.

Obviously, feature selection has a negative impact in the results reported for this scenario where the whole training dataset only includes four features. Hence, considering a subset of features is useless in the case of dealing with simple test-bed environments.

Unfortunately, considering different combinations of parameters, it is hard to point out a single one. Even though "J48G MCSs Bag" seems to be the best algorithm in many cases, all results are so similar that we cannot draw significant conclusions. Anyway, we can appreciate how the proposed framework achieves very good results no matter the selected classification method. This fact is due to the inherent simplicity of the analyzed scenario.

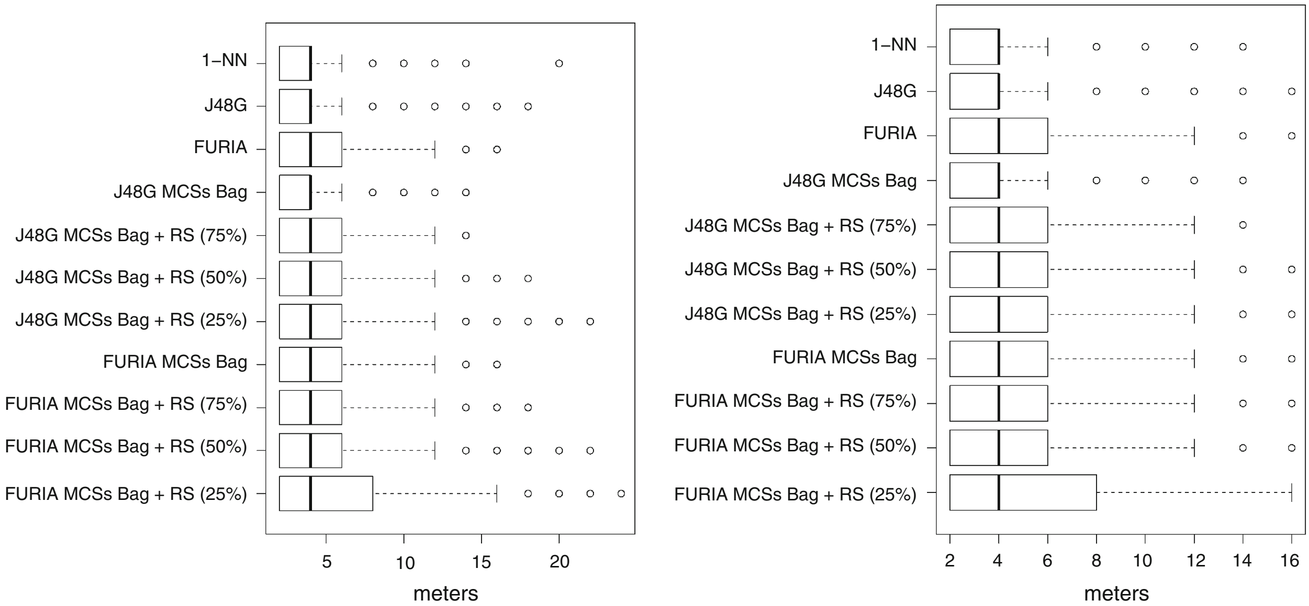
With the aim of making a deeper analysis regarding the behavior of the classifiers used, in addition to the classification accuracy, the distance between the real position and the estimated one, which is also an important issue in the WiFi localization application, is calculated (only when misclassification takes place). The dispersion of these results for each algorithm used is shown in Fig. 5 by means of boxplots (the possible outliers are represented by circles). On the right side (Fig. 5b) a zoom of the boxplots from the left side (Fig. 5a) is presented.

In the view of this figure, it can be noticed that the median for all the algorithms is the same (4 m). Furthermore, 1-NN, J48G, and "J48G MCSs Bag" seem to be providing the lowest error distances, however "J48G MCSs Bag" yields the outliers with the smallest distance value out of the three approaches. On the other hand, "FURIA MCSs Bag + RS (25 %)" provides the worst results, since it obtains the highest value of the upper quartile (8 m) and the farthest outliers (up to 24 m).

Anyway, trying to have a fair global view, both Table 1 and Fig. 5 should be considered together. Unfortunately, although it seems that "J48G MCSs Bag" is the algorithm worth pointing, since it outperforms all the others taking into account both accuracy and error distance distribution, still clear and sound conclusions cannot be drawn. This is due to the simplicity of this first illustrative scenario where all algorithms (even the basic 1-NN) provide really good results.

Finally, Table 2 reports the average on-line evaluation time obtained for the entire test set by the algorithms considered (corresponding to the execution of the schema detailed in Fig. 1). It is constructed in the same fashion as the previous table. The first column presents all the algorithms evaluated, while the next four columns present the reported execution time for each block size (Ni).

Regarding this table, it can be noticed that all types of MCSs reported quite similar and small on-line evaluation time. In contrast, in the case of 1-NN the on-line evaluation time is roughly 7 times longer than the MCS-based methods. We have also included J48G and FURIA just for comparison purpose. As expected, they achieve the shortest execution times. Notice that, the reported time refers to the total computational time, because in the case of 1-NN there is no learning stage, i.e., classification is carried out on the fly without an explicit model previously generated. Of course, a localization system requires high accuracy, however it also requires a low online execution time to provide the user a quick response and this is the main drawback of 1-NN.



(a) Block size N10 **(b)** Block size N10 (zoomed)

Fig. 5 Reported results (considering block size N10) in terms of error distances (in meters), paying attention to the misclassified positions in Scenario 1 (ECSC test-bed environment)

Table 2 Execution time (measured in seconds) for different classification and aggregation methods during the on-line stage in Scenario 1 (ECSC test-bed environment)

Algorithm	N1	N4	N7	N10
1-NN	15.650	15.806	16.437	14.944
J48G	0.311	0.326	0.326	0.332
FURIA	0.938	1.123	1.082	0.929
J48G MCSs				
Bag	2.268	2.304	2.329	2.290
Bag + RS (75 %)	2.321	2.210	2.251	2.310
Bag + RS (50 %)	2.142	2.212	2.219	2.248
Bag + RS (25 %)	2.955	2.301	2.348	2.407
FURIA MCSs				
Bag	2.780	3.550	3.736	3.311
Bag + RS (75 %)	2.868	3.597	3.759	3.322
Bag + RS (50 %)	2.943	3.660	3.776	3.337
Bag + RS (25 %)	2.876	3.653	3.808	3.370

4.2.2 *Results in a realistic case (Scenario 2)*

This section presents the results obtained for the second scenario. It represents a much more complex (and realistic) case. It actually becomes a quite hard classification problem. This state is confirmed in Table 3, which is built in the same way as Table 1. It summarizes the achieved results in terms of accuracy. The best result for each given Ni is highlighted in bold font.

Table 3 Accuracy results for different classification and aggregation methods in Scenario 2 (UAH environment)

Algorithm	N1	N4	N7	N10
1-NN	0.490	0.501	0.523	0.536
J48G	0.566	0.578	0.586	0.589
FURIA	0.534	0.564	0.581	0.595
J48G MCSs				
Bag	0.644	0.679	0.690	0.694
Bag + RS (75 %)	0.657	0.697	0.711	0.720
Bag + RS (50 %)	0.697	0.735	0.749	0.757
Bag + RS (25 %)	0.731	0.776	0.789	0.797
FURIA MCSs				
Bag	0.624	0.667	0.680	0.688
Bag + RS (75 %)	0.675	0.715	0.726	0.734
Bag + RS (50 %)	0.723	0.769	0.785	0.794
Bag + RS (25 %)	**0.733**	**0.790**	**0.803**	**0.809**

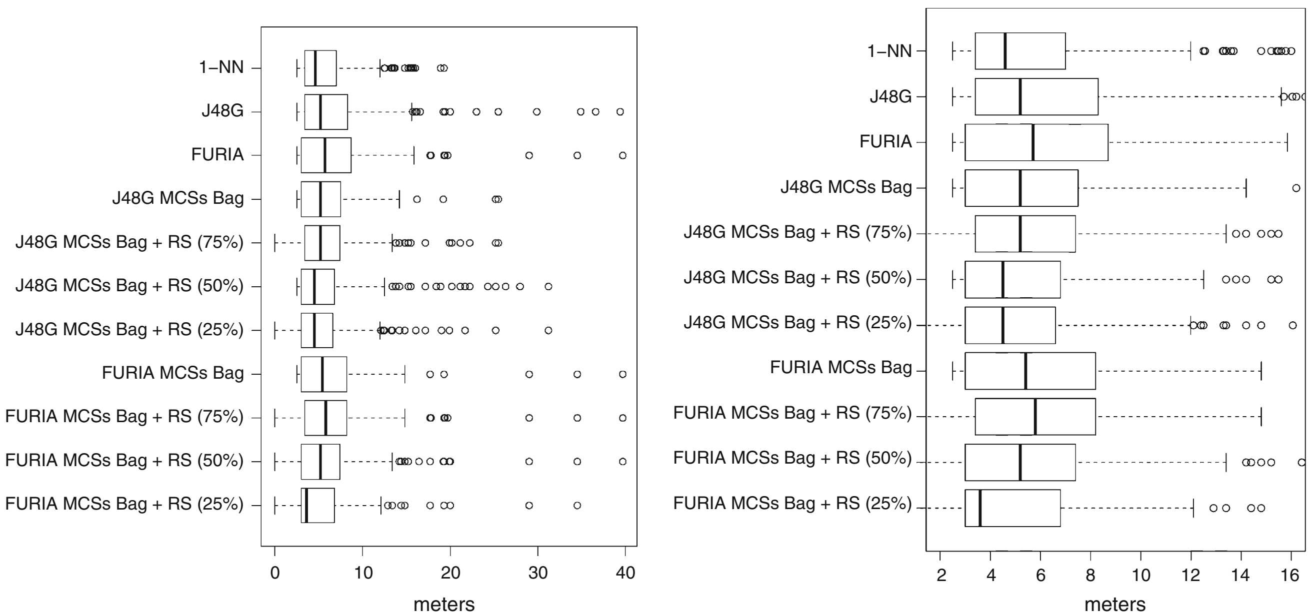
Looking carefully at this table, it can be noticed that reported accuracy significantly decreases in comparison with the results reported for the first scenario (look at Table 1). It is also worth mentioning how 1-NN obtains the worst results (around 0.5 what means that only one out of each two positions is correctly estimated). Thus, we observe how the performance of 1-NN drops dramatically in the case of dealing with complex and realistic environments. Both single classifiers, J48G and FURIA also obtain much worse results than those reported in Scenario 1, however they both outperform 1-NN no matter the block size (Ni). On the other hand, the best result (0.809) is obtained by "FURIA MCSs Bag and RS (25 %)" for N10. Moreover, "FURIA MCSs Bag and RS (25 %)" outperforms the other algorithms for all block sizes (Ni). From these facts, it can be confirmed the need of adopting the MCS-based approach in order to deal properly with high dimensional problems like those arising from complex environments like Scenario 2. Moreover, fuzzy methods like FURIA exhibit all their potential in the context of very noisy problems where classical methods do not perform so well. This is due to the characteristics of the fuzzy rules generated by FURIA. On the one hand, thanks to RS (25 %) FURIA handles much smaller number of features to be further filtered by its own internal feature selection algorithm in order to consider only the most representative features during the rule generation process. On the other hand, the "soft boundaries" determined by the fuzzy rule base fit well this complex classification problem.

In addition, there are some interesting conclusions derived from this table that deserve to be highlighted:

- Both J48G MCSs and FURIA MCSs perform better when combined with Bag + RS than with Bag only. "FURIA MCSs Bag + RS" outperform "J48G MCSs Bag + RS" in all cases. However, "J48G MCSs Bag" slightly outperform "FURIA MCSs Bag".
- MCSs obtain the highest accuracy when decreasing the number of features selected by RS (the smaller percentage of selected features, the more accurate results are reported). This fact is probably due to the so noisy and redundant signals measured from some APs in this complex scenario. Remind that in this scenario the number of handled APs is huge (over one-hundred).
- Considering the block size parameter (Ni), like we observed in the previous scenario, it is correlated with the accuracy. The larger the block size is, the higher accuracy is obtained (however, already for N7 good results are reported).

Like in the previous scenario, we can get a deep insight in the evaluation of the behavior of the classifiers used. Apart from paying attention to the classification accuracy, in the WiFi localization application, we should also take care of the error distance regarding the real position and the estimated one, which is considered only when misclassification takes place. Figure 6 depicts a dispersion of the the error distance (in meters) for each algorithm evaluated by means of boxplots (the possible outliers are represented by circles).

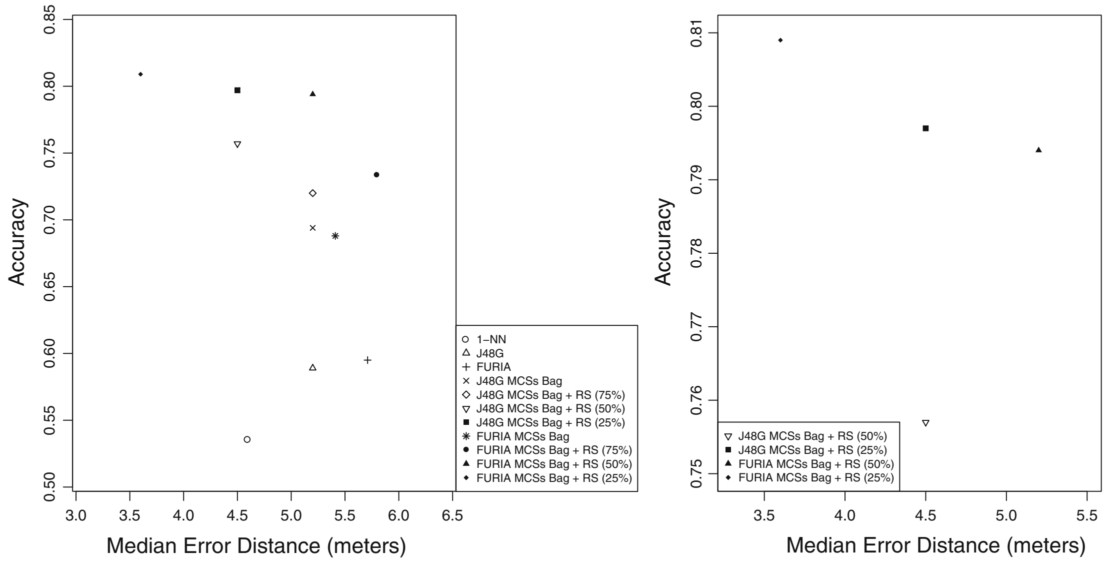
We have focused only on N10 because it yields the highest accuracy results. On the right side (Fig. 6b) a zoom of the boxplots from the left side (Fig. 6a) is presented. In the light of this figure, it can be noticed that the best values are obtained by "FURIA MCSs Bag + RS (25 %)" and "J48G MCSs Bag + RS (25 %)". The former does so for the lower quartile and the median, whereas the latter achieves it for the lower and the upper quartile but also other MCSs combined with Bag + RS such as "J48G MCSs Bag + RS (50 %)" or "FURIA MCSs Bag + RS (50 %)" perform fairly well. On the other hand, several



(a) Block size N10

(b) Block size N10 (zoomed)

Fig. 6 Reported results (considering block size N10) in terms of error distances (in meters) for the misclassified positions in Scenario 2 (UAH test-bed environment)



(a) Accuracy vs Median Error Distance

(b) Accuracy vs Median Error Distance (zoomed)

Fig. 7 Comparison of algorithms used in Scenario 2 (UAH test-bed environment). Accuracy (y-axis) versus Median Error Distance (x-axis)

outliers with high distance value (up to 40 m) are obtained by FURIA MCSs (e.g., "FURIA MCSs Bag", "FURIA MCSs Bag + RS (50 %)", and "FURIA MCSs Bag + RS (75 %)"), as well as J48G and FURIA.

Only looking carefully at both Table 3 and Fig. 6, it is possible to have a fair global view of the goodness of the reported results. In order to make this task easier, Fig. 7a presents the median values of the error distances represented in Fig. 6 (x-axis) against the accuracy values reported in Table 3 (y-axis), for all the eleven algorithms evaluated in the experiments. Notice that, the best solution would lie in the upper-left corner having the highest accuracy and the smallest median. Thus, to have a better insight we have also presented a zoom of that part in Fig. 7b. In the view of this figure, it can be clearly seen that the best solutions (they outperform all the remaining ones taking into account both classification rate and median error distance) are obtained by MCSs combined with Bag + RS. It is worth pointing "FURIA MCS Bag and RS (25 %)", since it outperforms all the other approaches considering both measures. By contrast the basic 1-NN does not achieve especially good behavior. It is placed in the lower part of the figure since it produces very low accuracy, but it does not report the most outstanding median value either. All the MCSs combined with Bag + RS clearly outperform 1-NN, when considering accuracy and in half of the cases with respect to median error distance. Thus, we can draw a clear conclusion. The proposed MCS-based approach strongly outperforms the standard 1-NN. Depending on the type of MCS selected different objectives could be achieved.

Finally, Table 4 summarizes the average on-line evaluation time (corresponding to the execution of the schema detailed in Fig. 1) reported by all the analyzed algorithms when dealing with the entire test set. This table is constructed in the same fashion as Table 2, regarding several block sizes (Ni) for each evaluated algorithm. In the view of the reported run-times, it can be noticed that all evaluated MCSs were able to get much smaller on-line evaluation time than 1-NN. The reported execution time is roughly 8 times smaller with MCSs than with 1-NN. Moreover, if we make a comparison between execution

Table 4 Execution time (measured in seconds) for different classification and aggregation methods during the on-line stage in Scenario 2 (UAH test-bed environment)

Algorithm	N1	N4	N7	N10
1-NN	17.633	17.849	18.677	16.067
J48G	0.008	0.007	0.007	0.007
FURIA	0.010	0.008	0.010	0.010
J48G MCSs				
Bag	1.780	1.732	1.772	1.790
Bag + RS (75 %)	1.819	1.849	1.836	1.996
Bag + RS (50 %)	1.800	1.750	1.782	1.810
Bag + RS (25 %)	1.841	1.772	1.741	1.784
FURIA MCSs				
Bag	1.801	1.810	1.768	1.783
Bag + RS (75 %)	1.833	1.744	1.783	1.814
Bag + RS (50 %)	1.820	1.773	1.796	1.805
Bag + RS (25 %)	1.819	1.771	1.803	1.842

times reported for both scenarios (Table 2 vs Table 4), we can see how 1-NN requires more time when the complexity of the test-bed environment increases (Scenario 2), while simultaneously the execution time reported by MCSs slightly decreases. This fact is due to 1-NN is a lazy method where all computational effort is made during the on-line execution stage while MCS-based methods are eager methods, i.e., they spend most of the time during the off-line learning stage but they are fast and extremely efficient in the on-line execution stage.

5 Conclusions and future works

In this study, we proposed a framework in which both bagging and the combination of bagging with random subspace are applied to train J48G-based MCSs (fast decision trees) and FURIA-based MCSs (fuzzy rule-based classifiers) devoted to deal with the WiFi localization problem, that is faced as a high dimensional classification problem. By using the above mentioned techniques, we aimed to obtain a WiFi localization system which provides high accuracy with a reasonable on-line computational time.

We have conducted a comprehensive experiment on two real test-bed environments, a simple scenario composed of 39,000 instances, 4 features, and 13 classes (locations) and a more realistic one (also more complex) composed of 8,520 instances, 143 (134) features, and 71 classes (locations). It turned out that the designed MCSs were able to outperform the accuracy of the state-of-the-art nearest neighbor algorithm with a lower execution time in both analyzed scenarios. Especially big accuracy differences were reported for the second scenario. Thus, our approach is very promising for tackling with complex and realistic environments.

Of course, further research should be performed. One of the next steps we will consider in the future is to incorporate some other advanced techniques, like random (Breiman 2001) and rotation forest (Rodriguez et al. 2006), to generate MCSs. This is likely to yield even more accurate systems. Another interesting research line to follow is to apply our MCS-based framework into the context of a novel hierarchical WiFi localization approach that we have recently sketched (Hernández et al. 2012). Finally, we would like to explore the chance of extending the basic nearest neighbor algorithm in order to integrate it in our framework in combination with the other evaluated component classifiers J48G and FURIA.

Acknowledgments This work has been partly supported by the Spanish Ministry of Economy and Competitiveness under INFANTREE project (JCI-2011-09839), ABSYNTHE project (TIN2011-29824-C02-01 and TIN2011-29824-C02-02), and the European Centre for Soft Computing (ECSC) located at Mieres (Asturias, Spain).

References

Alonso JM, Ocaña M, Hernández N, Herranz F, Llamazares A, Sotelo MA, Bergasa LM, Magdalena L (2011) Enhanced WiFi localization system based on soft computing techniques to deal with small-scale variations in wireless sensors. Appl Soft Comput 11(8):4677–4691

Alonso JM, Ocaña M, Sotelo MA, Bergasa LM, Magdalena L (2009) WiFi localization system using fuzzy rule-based classification. Lecture Notes in Computer Science, Comput Aid Syst Theory-EUROCAST09 5717:383–390

Alvarez-Alvarez A, Alonso JM, Trivino G, Hernández N, Herranz F, Llamazares A, Ocaña M (2010) Human activity recognition applying computational intelligence techniques for fusing information related to WiFi positioning and body posture. In: Proceedings of the IEEE international conference on Fuzzy systems, pp 1881–1885

Astrain JJ, Villadangos J, Garitagoitia JR, González de Mendívil JR, Cholvi V (2006) Fuzzy location and tracking on wireless networks. In: Proceedings of the 4th ACM international workshop on mobility management and wireless access, pp 84–91

Bahillo A, Lorenzo RM, Mazuelas S, Fernandez P, Abril EJ (2009) Assessment of the shadow caused by the human body on the personal RF dosimeters reading in multipath environments. In: Biomedical Engineering, pp 133–144

Bahl P, Padmanabhan V (2000) RADAR: an in-building RF-based user location and tracking system. In: Proceedings of the IEEE computer and communications societies, pp 775–784

Banfield RE, Hall LO, Bowyer KW, Kegelmeyer WP (2007) A comparison of decision tree ensemble creation techniques. IEEE Trans Pattern Anal Mach Intell 29(1):173–180

Bonissone PP, Cadenas JM, Garrido MC, Díaz-Valladares RA (2010) A fuzzy random forest. Int J Approx Reason 51(7):729–747

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Chiang KW, Huang YW (2008) An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications. Appl Soft Comput 8(1):722–733

Cohen WW (1995) Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning, Morgan Kaufmann, pp 115–123

Cordón O, del Jesus MJ, Herrera F (1999) A proposal on reasoning methods in fuzzy rule-based classification systems. Int J Approx Reason 20:21–45

Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13:21–27

Dharne AG, Lee J, Jayasuriya S (2006) Using fuzzy logic for localization in mobile sensor networks: simulations and experiments. In: Proceedings of the American control conference. IEEE, pp 2066–2071

Dietterich TG (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 40(2):139–157

Elnahrawy E, Li X, Martin RP (2004) The limits of localization using signal strength: a comparative study. In: First annual IEEE communications society conference on sensor Ad Hoc communications and networks, pp 406–414

Enge P, Misra P (1999) Special issue on GPS: the global positioning system. In: Proceedings of the IEEE, pp 3–172

Gallagher TJ, Li B, Dempster AG, Rizos C (2010) A sector-based campus-wide indoor positioning system. In: IEEE international conference on indoor positioning and indoor navigation, pp 1–8

Garcia-Valverde T, Garcia-Sola A, Gomez-Skarmeta A, Botia JA, Hagras H, Dooley J, Callaghan V (2012) An adaptive learning fuzzy logic system for indoor localisation using Wi-Fi in ambient intelligent environments. In: Proceedings of the IEEE World congress on computational intelligence, pp 25–32

Hernández N, Alonso JM, Magro M, Ocaña M (2012) Hierarchical WiFi localization system. In: international workshop on perception in robotics, IEEE intelligent vehicles symposium, pp P21.1–P21.6

Herrero-Pereza D, Martinez-Barbera H, LeBlanc K, Saffiotti A (2010) Fuzzy uncertainty modeling for grid based localization of mobile robots. Int J Approx Reason 51:912–932

Ho T (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844

Hühn JC, Hüllermeier E (2009) FURIA: an algorithm for unordered fuzzy rule induction. Data Min Knowl Disc 19(3):293–319

Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20:226–239

Kuncheva LI (2001) Using measures of similarity and inclusion for multiple classifier fusion by decision templates. Fuzzy Sets Syst 122:401–407

Kuncheva LI (2002) A theoretical study on six classifier fusion strategies. IEEE Trans Pattern Anal Mach Intell 24:281–286

Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley, New Jersey

Kuncheva LI, Bezdek JC, Duin RPW (2001) Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recogn Lett 34:299–314

Magdalena L (2008) What is soft computing? revisiting possible answers. In: 8th International FLINS conference on computational intelligence in decision and control, World Scientific, Singapore, pp 3–10

Menendez P, Campomanes C, Trawiński K, Alonso JM (2011) Topology-based indoor localization by means of WiFi fingerprinting with a computational intelligent classifier. In: Proceedings of the 11th IEEE international conference on intelligent system design and applications, pp 1020–1025

Nerguizian C, Belkhous S, Azzouz A, Nerguizian V, Saad M (2004) Mobile robot geolocation with received signal strength (RSS) fingerprinting technique and neural networks. In: IEEE international conference on industrial technology, pp 1183–1185

Optiz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198

Outemzabet S, Nerguizian C (2008) Accuracy enhancement of an indoor ANN-based fingerprinting location system using Kalman filtering. In: 19th IEEE international symposium on personal, indoor and mobile radio communications, pp 1–5

Outemzabet S, Nerguizian C (2008) Accuracy enhancement of an indoor ANN-based fingerprinting location system using particle filtering and a low-cost sensor. In: IEEE international conference on vehicular technology, pp 2750–2754

Palmer N, Kemp R, Kielmann T, Bal H (2012) The case for smartphones as an urgent computing client platform. Procedia Comput Sci 9:1667–1676

Panov P, Džeroski S (2007) Combining bagging and random subspaces to create better ensembles. In: Proceedings of the 7th international conference on intelligent data analysis. Springer, Berlin, pp 118–129

Paul R, Aguirre E, Garcia-Silvente M, Muñoz-Salinas R (2012) A new fuzzy based algorithm for solving stereo vagueness in detecting and tracking people. Int J Approx Reason 53(4): 693–708

Quinlan JR (1991) Improved estimates for the accuracy of small disjuncts. Mach Learn 6(1):93–98

Rodriguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28(10):1619–1630

Schapire R (1990) The strength of weak learnability. Mach Learn 5(2):197–227

Trawiński K, Cordón O, Quirin A (2011) On designing fuzzy rule-based multiclassification systems by combining FURIA with bagging and feature selection. Int J Uncertain Fuzziness Knowl Based Syst 19(4):589–633

Tsymbal A, Pechenizkiy M, Cunningham P (2005) Diversity in search strategies for ensemble feature selection. Inform Fusion 6(1): 83–98

Webb GI (1999) Decision tree grafting from the all-tests-but-one partition. In: Sixteenth international joint conference on artificial intelligence. Morgan Kaufmann, pp 702–707

Witten IH, Frank E, Hall MA (2011) Data Mining: practical machine learning tools and techniques. 3rd edn, Morgan Kaufmann, San Francisco

Woods K, Kegelmeyer WP, Bowyer K (1997) Combination of multiple classifiers using local accuracy estimates. IEEE Trans Pattern Anal Mach Intell 19:405–410

Youssef M, Agrawala A (2003) Small-scale compensation for wlan location determination systems. In: Proceedings of the ACM workshop on wireless security, pp 11–20

Yun S, Lee J, Chung W, Kim E, Kim S (2009) A soft computing approach to localization in wireless sensor networks. Expert Syst Appl 36(4):7552–7561

Zhou ZH (2005) Ensembling local learners through multimodal perturbation. IEEE Trans Syst Man Cybern Part B Cybern 35(4):725–735

Bibliography

[AC03] M. C. C. Abreu and A. M. P. Canuto. An experimental study on the importance of the choice of the ensemble members in fuzzy combination methods. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 723–728, Rio de Janeiro (Brazil), 2003.

[ACdJH07] J. J. Aguilera, M. Chica, M. J. del Jesus, and F. Herrera. Niching genetic feature selection algorithms applied to the design of fuzzy rule based classification systems. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1794–1799, London (UK), 2007.

[AFFL+11] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.

[AM10] J. M. Alonso and L. Magdalena. Are fuzzy systems as interpretable (readable and understandable) as the fuzzy community usually claims? In Garceta, editor, *Third Workshop on Fuzzy Logic and Soft Computing (LFSC2010) Spanish Congress on Computer Science (CEDI2010)*, pages 475–482, Valencia (Spain), September 2010.

[AMGR09] J. M. Alonso, L. Magdalena, and G. González-Rodríguez. Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning*, 51:115–134, 2009.

[AOS+09] J. M. Alonso, M. Ocaña, M. A. Sotelo, L. M. Bergasa, and L. Magdalena. WiFi localization system using fuzzy rule-based classification. *Lecture Notes in Computer Science, Computer Aided System Theory - EUROCAST09*, 5717:383–390, 2009.

[Bat94] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[BC94] R. Battiti and A. M. Colla. Democracy in neural nets: voting schemes for classification. *Neural Networks*, 7(4):691–707, 1994.

[BCGDV10] 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):729–747, 2010.

[BGTM10] P. Bulacio, S. Guillaume, E. Tapia, and L. Magdalena. A selection approach for scalable fuzzy integral combination. *Information Fusion*, 11(2):208–213, 2010.

[BHBK07] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180, 2007.

[BK99] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999.

[BM98] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. `http://archive.ics.uci.edu/ml`.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[CCHE03] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (Eds.). *Interpretability Issues in Fuzzy Modeling.* Springer Verlag, Berlin Heidelberg, 2003.

[CdJH99] O. Cordón, M. J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20:21–45, 1999.

[CGH+04] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems*, 141(1):5–31, 2004.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21– 27, 1967.

[CH83] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology.* Elsevier Science, New York, 1983.

[CHHM01] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases.* World Scientific, 2001.

[CHZ03] O. Cordón, F. Herrera, and I. Zwir. A hierarchical knowledge-based environment for linguistic modeling: Models and iterative methodology. *Fuzzy Sets and Systems*, 138(2):307–341, 2003.

[CJA04] N. V. Chawla, N. Japkowicz, and A.Kolcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6:1–6, 2004.

[CK95] S.-B. Cho and J. H. Kim. Multiple network fusion using fuzzy logic. *IEEE Transactions on Neural Networks*, 6(2):497–501, 1995.

[CLV07] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd Edition.* Springer, 2007.

[Coh95] W. W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[Cor11] O. Cordón. A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning*, 52(6):894–913, 2011.

[CQ10] 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. *International Journal of Hybrid Intelligent Systems*, 7(1):45–64, 2010.

[CQS08] 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 *Eighth International Conference on Hybrid Intelligent Systems (HIS)*, pages 549–554, Barcelona, 2008.

[CRSH07] J. Canul-Reich, L. Shoemaker, and L. O. Hall. Ensembles of fuzzy classifiers. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, London, 2007.

[Die98] T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[Die00] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[DP97] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[DS79] B. V. Dasarathy and B. V. Sheela. A composite classifier system design: Concepts and methodology. *Proceedings of IEEE*, 67(5):708–713, 1979.

[DSM06] E. M. Dos Santos, R. Sabourin, and P. Maupin. Single and multi-objective genetic algorithms for the selection of ensemble of classifiers. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3070–3077, Vancouver, 2006.

[DSM08] 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):2993–3009, 2008.

[DVA09] N. Dimililer, E. Varoglu, and H. Altincay. Classifier subset selection for biomedical named entity recognition. *Applied Intelligence*, 31:267–282, 2009.

[FR95] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[FY89] D. Filev and R. R. Yager. A generalized defuzzification method under badd distributions. *International Journal of Intelligent Systems*, 6:687–697, 1989.

[GF95] A. E. Gegov and P. M. Frank. Hierarchical fuzzy control of multivariable systems. *Fuzzy Sets and Systems*, 72:299–310, 1995.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989.

[GR01] G. Giacinto and F. Roli. Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, 34(9):1879–1881, 2001.

[GR06] B. Gabrys and D. Ruta. Genetic algorithms in classifier fusion. *Applied Soft Computing*, 6(4):337–347, 2006.

[Gra95] M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 89:279–298, 1995.

[Her08] F. Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46, 2008.

[HH09] J. C. Hühn and E. Hüllermeier. FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.

[HH10] J. C. Hühn and E. Hüllermeier. An analysis of the FURIA algorithm for fuzzy rule induction. In *Advances in Machine Learning I*, pages 321–344. 2010.

[HHJ06] Y. M. Huang, C. M. Hung, and H. C. Jiau. Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications*, 7(4):720–747, 2006.

[HLnS13] D. Hernández-Lobato, G. Martínez-Mu noz, and A. Suárez. How large should ensembles of classifiers be? *Pattern Recognition*, 46(5):1323 – 1336, 2013.

[Ho98] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[IN06] H. Ishibuchi and Y. Nojima. Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers. *International Journal of Hybrid Intelligent Systems*, 3(3):129–145, 2006.

[INM99] H. Ishibuchi, T. Nakashima, and T. Morisawa. Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*, 103(2):223–238, 1999.

[INN05] H. Ishibuchi, T. Nakashima, and M. Nii. *Classification and Modeling With Linguistic Information Granules.* Springer, 2005.

[Jan98] C. Z. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(1):1–14, 1998.

[KBD01] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[KHDM98] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–238, 1998.

[KMH06] M. Kim, S. Min, and I. Han. An evolutionary approach to the combination of multiple classifiers to predict a stock price index. *Expert Systems with Applications*, 31:241–247, 2006.

[KR07] L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508, 2007.

[Kun00] L. I. Kuncheva. *Fuzzy Classifier Design*, volume 49 of *Studies in Fuzziness and Soft Computing.* Springer, 2000.

[Kun01a] L. I. Kuncheva. Combining classifiers: Soft computing solutions. In S.K. Pal and A. Pal, editors, *Pattern Recognition. From Classical to Modern Approaches*, chapter 15, pages 427–452. World Scientific, Singapore, 2001.

[Kun01b] L. I. Kuncheva. Using measures of similarity and inclusion for multiple classifier fusion by decision templates. *Fuzzy Sets and Systems*, 122:401–407, 2001.

[Kun02] L. I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:281–286, 2002.

[Kun03] L. I. Kuncheva. "Fuzzy" versus "nonfuzzy" in combining classifiers designed by boosting. *IEEE Transactions on Fuzzy Systems*, 11(6):729–741, 2003.

[Kun04] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

[KW03] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[LC07] Y.-H. Liu and Y.-T. Chen. Face recognition using total margin-based adaptive fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 18(1):178–192, 2007.

[LS97] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics*, 27:553–568, 1997.

[LY97] Y. Lu and F. Yamaoka. Fuzzy integration of classification results. *Pattern Recognition*, 30(11):1877–1891, 1997.

[Mar09] C. Marsala. Data mining with ensembles of fuzzy decision trees. In *IEEE Symposium on Computational Intelligence and Data Mining*, pages 348–354, Nashville (USA), 2009.

[MHZ+08] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2-3):427–436, 2008.

[NI06] Y. Nojima and H. Ishibuchi. Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion. In *International Conference on Hybrid Intelligent Systems and Conference on Neuro-Computing and Evolving Intelligence*, 2006. CD-ROM, 4 pages.

[NI07] Y. Nojima and H. Ishibuchi. Genetic rule selection with a multi-classifier coding scheme for ensemble classifier design. *International Journal of Hybrid Intelligent Systems*, 4(3):157–169, 2007.

[OM99] D. Optiz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[PD07] P. Panov and S. Džeroski. Combining bagging and random subspaces to create better ensembles. In *IDA'07: Proceedings of the 7th international conference on Intelligent data analysis*, pages 118–129, Berlin, Heidelberg, 2007. Springer-Verlag.

[PK06] W. Pedrycz and K. C. Kwak. Boosting of granular models. *Fuzzy Sets and Systems*, 157(22):2934–2953, 2006.

[PY96] D. Partridge and W. B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.

[Qui93] J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RG05] D. Ruta and B. Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1):63–81, 2005.

[RK07] J. J. Rodríguez and L. I. Kuncheva. Naïve bayes ensembles with a random oracle. In *Lecture Notes in Computer Science*, volume 4472, pages 450–458. Springer-Verlag, 2007.

[RKA06] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

[Sch90] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[SCQT10] L. Sánchez, O. Cordón, A. Quirin, and K. Trawinski. Introducing a genetic fuzzy linguistic combination method for bagging fuzzy rule-based multiclassification. In *Fourth International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*, pages 75–80, 2010.

[SS99] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[SS00] A. J. C. Sharkey and N. E. Sharkey. The *test and select* approach to ensemble combination. In *International Workshop on Multiclassifier Systems*, pages 30–44, Cagliari, 2000.

[Ste05] 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, editors, *Issues in the Representation and Processing of Uncertain and Imprecise Information*, pages 337–354. Akademicka Oficyna Wydawnicza EXIT, Warsaw, 2005.

[Sug74] M. Sugeno. *Theory of fuzzy integrals and its applications.* PhD thesis, Tokio Institute of Technology, 1974.

[SW49] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illlinois Press, 1949.

[TAH13] K. Trawiński, J. M. Alonso, and N. Hernandez. A multiclassifier approach for topology-based wifi indoor localization. *Soft Computing*, 17(10):1817–1831, 2013.

[TCQ11a] K. Trawiński, O. Cordón, and A. Quirin. A first study on a fuzzy rule-based multiclassification system framework combining furia with bagging and feature selection. In *In Proceedings of the World Conference on Soft Computing (WConSC)*, pages 167–175, San Francisco (USA), 2011.

[TCQ11b] K. Trawiński, O. Cordón, and A. Quirin. On designing fuzzy rule-based multiclassification systems by combining FURIA with bagging and feature selection. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 19(4):589–633, 2011.

[TCQ13] K. Trawiński, O. Cordón, and A. Quirin. On applying random oracles to fuzzy rule-based classifier ensembles for high complexity datasets. In *Proceedings of the Eigth International Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT). In press*, Milan (Italy), 2013.

[TCQS13a] K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez. Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity? *Knowledge-Based Systems. In press*, 2013.

[TCQS13b] K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez. Random oracles fuzzy rule-based multiclassifiers for high complexity datasets. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, Hyderabad (India), 2013.

[TCSQ13] K. Trawiński, O. Cordón, L. Sánchez, and A. Quirin. A genetic fuzzy linguistic combination method for fuzzy rule-based multiclassifiers. *IEEE Transactions on Fuzzy Systems*, 21(5):950–965, 2013.

[TG13] A. Tsakonas and B. Gabrys. A fuzzy evolutionary framework for combining ensembles. *Applied Soft Computing*, 13(4):1800 – 1812, 2013.

[TH06] H. Takahashi and H. Honda. Lymphoma prognostication from expression profiling using a combination method of boosting and projective adaptive resonance theory. *Journal of Chemical Engineering of Japan*, 39(7):767–771, 2006.

[Tor02] V. Torra. A review of the construction of hierarchical fuzzy systems. *International Journal of Intelligent Systems*, 17(5):531–543, 2002.

[TPC05] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.

[TQC12] 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. *International Journal of Computational Intelligence Systems*, 5(2):231–253, 2012.

[VLM$^+$99] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis. Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters*, 20(4):429–444, 1999.

[Web99] G. I. Webb. Decision tree grafting from the all-tests-but-one partition. In *Sixteenth International Joint Conference on Artificial Intelligence*, pages 702–707. Morgan Kaufmann, 1999.

[WKB97] K. Woods, W. P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.

[Wol92] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[XKS92] 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):418–435, 1992.

[Yag98] R. R. Yager. On the construction of hierarchical fuzzy systems model. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 28(1):55–66, 1998.

[YF94] R. R. Yager and D. P. Filev. *Essentials of fuzzy modeling and control.* Wiley-Interscience, New York, NY, USA, 1994.

[Zho05] Z. H. Zhou. Ensembling local learners through multimodal perturbation. *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):725–735, 2005.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3:257–271, 1999.

[ZWT02] Z.-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: many could be better than all. *Artificial Intelligence*, 137(1-2):239–263, 2002.

[ZWY04] X. Zhu, X. Wu, and Y. Yang. Dynamic selection for effective mining from noisy data streams. In *Proceedings of Fourth IEEE International Conference on Data Mining (ICDM)*, pages 305–312, Brighton, 2004.

[ZZ80] H. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets and Systems*, 4(1):37–51, 1980.

UNIVERSIDAD DE OVIEDO
Vicerrectorado de Internacionalización y Postgrado

CENTRO INTERNACIONAL DE POSTGRADO
CAMPUS DE EXCELENCIA INTERNACIONAL

FOR- MAT-VOA-035

ACEPTACIÓN COAUTORES PRESENTACIÓN TRABAJOS FORMANDO PARTE DE TESIS DOCTORAL COMO COMPENDIO DE PUBLICACIONES

1.- Datos personales del coautor		
Apellidos: Alonso Moral	Nombre: José María	
DNI/Pasaporte/NIE	Teléfono 985456545	Correo electrónico jose.alonso@softcomputing.es

2.- Publicaciones que formarán parte de la tesis y de las que es coautor
"A Multiclassifier Approach for Topology-based WiFi Indoor Localization", Soft Computing, In press, 2013. DOI:10.1007/s00500-013-1019-5

ACEPTACIÓN:

Acepto que las publicaciones anteriores formen parte de la tesis doctoral titulada "Design of Fuzzy Rule-based Ensembles based on FURIA, Diversity Induction and Evolutionary Algorithms"

Y elaborada por D. Krzysztof Trawinski

Mieres, 04.09.2013

Firma

UNIVERSIDAD DE OVIEDO
Vicerrectorado de Internacionalización y Postgrado

CENTRO INTERNACIONAL DE POSTGRADO
CAMPUS DE EXCELENCIA INTERNACIONAL

ACEPTACIÓN COAUTORES PRESENTACIÓN TRABAJOS FORMANDO PARTE DE TESIS DOCTORAL COMO COMPENDIO DE PUBLICACIONES

1.- Datos personales del coautor		
Apellidos: Hernández Parra	Nombre: Noelia	
DNI/Pasaporte/NIE 46839863H	Teléfono 918856807	Correo electrónico noelia.hernandez@depeca.uah.es

2.- Publicaciones que formarán parte de la tesis y de las que es coautor
"A Multiclassifier Approach for Topology-based WiFi Indoor Localization", Soft Computing, In press, 2013.

FOR-MAT-VOA-035

ACEPTACIÓN:
Acepto que las publicaciones anteriores formen parte de la tesis doctoral titulada "Design of Fuzzy Rule-based Ensembles based on FURIA, Diversity Induction and Evolutionary Algorithms" Y elaborada por D. Krzysztof Trawinski Madrid, 04.09.2013 Firma

RENUNCIA COAUTORES PRESENTACIÓN TRABAJOS FORMANDO PARTE DE TESIS DOCTORAL

1.- Datos personales del coautor		
Apellidos: Hernández Parra	Nombre: Noelia	
DNI/Pasaporte/NIE 46839863H	Teléfono 918856807	Correo electrónico noelia.hernandez@depeca.uah.es

2.- Tesis Doctoral
Título: Design of Fuzzy Rule-based Ensembles based on FURIA, Diversity Induction and Evolutionary Algorithms
Autor: Krzysztof Trawinski
Programa de doctorado: Ingeniería informática

2.- Publicaciones que formarán parte de la tesis y de las que es coautor
"A Multiclassifier Approach for Topology-based WiFi Indoor Localization", Soft Computing, In press, 2013.

RENUNCIA:
Renuncio a que las publicaciones anteriores sean presentadas como parte de otra tesis doctoral presentada como compendio de publicaciones. Madrid, 04.09.2013 Firma

FOR- MAT-VOA-036

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Diseño de métodos de combinación de clasificadores basados en reglas difusas usando FURIA, inducción de diversidad y algoritmos evolutivos	Inglés: Design of fuzzy rule-based ensembles using FURIA, diversity induction and evolutionary algorithms

2.- Autor	
Nombre: Krzysztof Trawiński	DNI/Pasaporte/NIE: Y0599931-J
Programa de Doctorado: Ingeniería Informática	
Órgano responsable: Comisión Académica del Programa de Doctorado	

RESUMEN (en español)

Los sistemas basados en reglas difusas han demostrado una alta capacidad de extracción y representación del conocimiento a la hora de modelar problemas de clasificación complejos y no lineales. Sin embargo, cuando se aplican a conjuntos de datos de alta complejidad, es decir con un gran número de variables y/o ejemplos, sufren la denominada “maldición de las dimensiones” (*curse of dimensionality*). Los métodos de combinación de clasificadores han demostrado ser una buena técnica para afrontar este tipo de problemas.

En esta tesis doctoral se propone un marco global basado en el enfoque de los métodos de combinación de clasificadores que permite a los sistemas basados en reglas difusas manejar conjuntos de datos de alta complejidad evitando el problema anterior. Para conseguir afrontar este objetivo, el marco de trabajo propuesto incorpora distintos métodos de combinación de clasificadores y considera algoritmos evolutivos para diseñar métodos de combinación de clasificadores basados en reglas difusas. Su estructura se basa en dos etapas: 1) Diseño de métodos de combinación de clasificadores basados en reglas difusas a partir de enfoques clásicos y avanzados, y 2) Diseño de nuevos métodos de selección y fusión de clasificadores base usando algoritmos evolutivos. Este enfoque permite diseñar varios métodos específicos de combinación de clasificadores basados en reglas difusas que permiten la mejora de la precisión en los resultados y la obtención de un buen equilibrio entre precisión y complejidad. Se han realizado experimentos exhaustivos con varios conjuntos de datos de alta complejidad (en lo que respecta al número de atributos y al número de ejemplos) procedentes de los repositorios UCI y KEEL que han demostrado el buen comportamiento de los métodos propuestos.

Además, se ha aplicado con éxito uno de los diseños concretos de combinación de clasificadores basados en reglas difusas a un problema real consistente en la localización en interiores utilizando topología WiFi. Esta tarea se corresponde con un problema de clasificación de alta dimensionalidad cuando se trata de un entorno complejo, que presenta la dificultad adicional de la incertidumbre asociada debido a la naturaleza de las señales Wi-Fi.

RESUMEN (en Inglés)

Fuzzy rule-based systems have shown a high capability of knowledge extraction and representation when modeling complex, non-linear classification problems. However, they suffer from the so-called curse of dimensionality when applied to high complexity datasets, which consist of a large number of variables and/or examples. Classifier ensembles have shown to be a good approach to deal with this kinds of problems.

In this PhD dissertation, we propose a classifier ensemble-based global framework allowing fuzzy rule-based systems to deal with high dimensional datasets avoiding the curse of dimensionality. Having this goal in mind, the proposed framework incorporates several classifier ensemble methodologies as well as evolutionary algorithms to design fuzzy rule-based classifier ensembles. The proposed framework follows a two-stage structure: 1) fuzzy rule-based classifier ensemble design from classical and advanced classifier ensemble design approaches, and 2) novel designs of evolutionary component classifier combination. By using our methodology, different fuzzy rule-based classifier ensembles can be designed dealing with several aspects such as the improvement of the performance in terms of accuracy and the obtaining a good accuracy-complexity trade-off. Exhaustive experiments carried out over several UCI and KEEL datasets with high complexity (considering both the number of attributes as well as the number of examples) have shown the good performance of the proposed classifier ensemble-based global framework.

Besides, one of the specific fuzzy rule-based classifier ensemble design approaches obtained from the proposed framework has been successfully applied to a real-world problem. It consists of topology-based WiFi indoor localization, which turns into a high dimensional classification problem when dealing with a complex environment. The complexity of this task is also characterized by the huge inherent uncertainty coming from the nature of WiFi signals.

SR. DIRECTOR DE DEPARTAMENTO DE INFORMÁTICA
SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN INGENIERIA INFORMÁTICA