International Journal of Neural Systems, Vol. , No. (2017) 1–17

A Pareto Based Ensemble with Feature and Instance Selection for Learning from Multi-Class Imbalanced Datasets

Alberto Fernández
Department of Computer Science and Artificial Intelligence, University of Granada
Granada, 18071, Spain
E-mail: alberto@decsai.ugr.es

Cristobal José Carmona
Department of Civil Engineering, University of Burgos
Burgos, 09006, Spain; and
Leicester School of Pharmacy, De Montfort University
Leicester, LE1 9BH, United Kingdom
E-mail: cjcarmona@ubu.es

María José del Jesus
Deparment of Computer Science, University of Jaén
Jaén, 23071, Spain
E-mail: mjjesus@ujaen.es

Francisco Herrera
Department of Computer Science and Artificial Intelligence, University of Granada
Granada, 18071, Spain; and
Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University (KAU)
Jeddah, 80200, Saudi Arabia
E-mail: herrera@decsai.ugr.es

Imbalanced classification is related to those problems that have an uneven distribution among classes. In addition to the former, when instances are located into overlapped areas, the correct modeling of the problem becomes harder. Current solutions for both issues are often focused on the binary case study, as multi-class datasets require an additional effort to be addressed. In this research, we overcome these problems by carrying out a combination between feature and instance selection. Feature selection will allow simplifying the overlapping areas easing the generation of rules to distinguish among the classes. Selection of instances from all classes will address the imbalance itself by finding the most appropriate class distribution for the learning task, as well as possibly removing noise and difficult borderline examples. For the sake of obtaining an optimal joint set of features and instances, we embedded the searching for both parameters in a Multi-Objective Evolutionary Algorithm, using the C4.5 decision tree as baseline classifier in this wrapper approach. The multi-objective scheme allows taking a double advantage: the search space becomes broader, and we may provide a set of different solutions in order to build an ensemble of classifiers. This proposal has been contrasted versus several state-of-the-art solutions on imbalanced classification showing excellent results in both binary and multi-class problems.

Keywords: Imbalanced Classification, Multi-class, Overlapping, Feature Selection, Instance Selection, Multi-objective Evolutionary Algorithms, Ensembles

1. Introduction

When addressing a classification task, researchers and practitioners often find that some of the classes are harder to recognize than others. As a result, the accuracy obtained in these cases is much lower than for the remaining ones. This issue, known as the problem of "difficult classes",[26] is mainly due to the structure and inner characteristics of the data.[40,56]

We may refer to classification with imbalanced data[8,51] as a particular example of the former. This scenario is shown when learning algorithms face an uneven class distribution such as in medical applications[57] or business failure prediction.[7] Focusing on accuracy and generalization, models are often biased towards majority class examples, so minority ones are more difficult to discriminate. When the number of classes increases, so does the number of boundaries to consider, imposing additional restrictions to the classification algorithm.[19] However, imbalance is not the solely cause for this abnormal behavior. Specifically, one of the main drawbacks in classification is related to overlapping between classes.[29,36] Rules with a low confidence and/or coverage can be discarded in favor of more general ones, because they are associated with overlapped boundary areas.

The issue of overlapping is strongly related with the attributes that represent the problem. It is well known that a large number of features can degrade the discovery of the borderline areas of the problem, either because some of these variables might be redundant or because they do not show a good synergy among them. Therefore, the use of feature selection can ease to diminish the effect of overlapping.[4,13] However, the imbalance class problem cannot be addressed by itself just by carrying out a feature selection. For this reason, it is also mandatory to perform a preprocessing of instances by resampling the training data distribution,[6,51] avoiding a bias of the learning algorithm towards the majority classes. Additionally, the former approaches can be integrated into an ensemble-type classifier, both for instance selection[23,24] and feature selection.[2,66]

Obtaining the optimal set of features and instances for a given problem is not a trivial task. For this reason, an optimization procedure is often required, as they are known to improve the quality of Data Mining systems.[35,47] Among different approaches, recent works have shown the goodness of Multi-Objective Evolutionary Optimization (MOEA) procedures[67] due to their ability to perform a good exploration and exploitation of the solution space.[37,55] In particular, for imbalanced classification, several bioinspired approaches have shown to be especially efficient and valuable.[43,44]

In this research, we propose EFIS-MOEA, which stands for "Ensemble classifier from a Feature and Instance Selection by means of Multi-Objective Evolutionary Algorithm." This novel approach addresses learning on difficult classes focusing on the uneven class distribution and the overlapping simultaneously, as an extension of our previous work on the topic.[18] To do so, we will embed the C4.5 decision tree[52] in a wrapper procedure, applying the well-known NSGA-II multi-objective optimization algorithm.[12] The basis for this methodology involves several components. First, feature selection is devoted to simplify the overlapping areas easing the generation of rules to distinguish between the classes. Second, selection of instances from all classes will address the imbalance itself by finding the most appropriate class distribution for the learning task, as well as possibly removing noise and difficult borderline examples. Finally, the non-dominated solutions of the Pareto front from the MOEA can be directly combined into an ensemble of classifiers.[53] Accordingly, it allows reinforcing the recognition capabilities of the individual classifiers.[62]

It is known that the C4.5 classifier carries out an inner feature selection process by itself based on the information gain. Our approach is intended to help the learning process of C4.5 by carrying out a pre-selection of the variables based on the intrinsic characteristics of the problem. In particular, and as stated previously, we focus on the possible overlapping among the classes. In addition, the capabilities of C4.5 makes it a good choice to develop an ensemble system.[54]

For a fair validation of our novel EFIS-MOEA proposal, we have set up two different experimental frameworks for both binary and multi-class case studies:

(1) The first framework will serve us as an initial case study in order to analyze the behavior of EFIS-MOEA with respect to the overlapping between classes. In this scenario, we have selected a number of 66 different problems commonly used

in this area of research,[40] where half of them show a high degree of overlapping. We will contrast the performance of our methodology versus the SMOTE+ENN preprocessing technique.[6,10]

(2) In the second case, we use 24 different imbalanced datasets. We have set up a framework of difficult problems as the overlapping can be increased among the different set of classes. In order to provide a strong support to the goodness of EFIS-MOEA in this particular scenario, we will contrast the results versus the best algorithms from the state-of-the-art on multi-class imbalanced classification,[19] namely the AdaBoost.NC ensemble,[58] a global cost-sensitive learning approach,[68] and SMOTE+ENN with One-vs-One (OVO) methodology.[22] We will also make use of Random Forest[9] as a very robust approach for general classification.

All lessons learned extracted from these experimental results will be supported by means of the statistical analysis of the results.[28]

In order to carry out the research, this manuscript is arranged as follows. Section 2 introduces the problem of classification with imbalanced datasets, including its definition and characteristics, and the solutions developed to address this issue. Section 3 describes our novel EFIS-MOEA approach for addressing the problem of binary and multi-class imbalanced problems. Next, the details about the experimental framework regarding datasets, parameters, and statistical tests are provided in Section 4. Section 5 contains the experimental results and the analysis that has been carried out. Finally, Section 6 concludes the paper, and provides some topics for future work.

2. Imbalanced Datasets in Classification

The characteristics that define each class in a classification problem are usually different: the number of instances (distribution of examples), dependency among classes (including overlapping), or even relations between the examples of the own class.[8,40,51] Taking all of these into account, we may observe that in some problems there can be several classes that are harder to distinguish than others.[26]

Among all data intrinsic characteristics, the one that possibly hinders the performance in a higher degree is the overlapping between classes.[29,36] It is shown when a region of the data space contains a similar quantity of training data from each class, imposing a hard restriction to finding discrimination functions.[13,29]

To compute the overlapping degree for a given problem, the *maximum Fisher's discriminant ratio* ($F1$ metric)[32] is used. It is defined for one feature dimension as: $f = \frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}$ being μ_1, μ_2, σ_1^2, σ_2^2 the means and variances of the two classes in that feature dimension. Therefore, $F1 = \max_{i=1\ldots n} f_i$, so that smaller values imply a harder class separability.

In the context of imbalanced datasets, less represented classes are usually more affected by this issue, due to the generalization bias of the learning algorithms.[40] A dataset is said to be imbalanced when a class or set of classes are represented in a smaller percentage than the others. A common threshold to determine this scenario is when the ratio between the largest class and the smaller is about 1.5.[40]

In order to address the uneven class distribution, a large number of approaches have been developed throughout the years. They are based on methodologies that act at the data level,[6] algorithmic level,[5] or that apply a cost-sensitive learning.[15] These solutions can be applied directly over a single classifier, or they can be combined into an ensemble learning procedure,[23] aiming at boosting the performance by providing more diversity to the global system.

Among all these methodologies, those based on resampling are the most popular due to their versatility and robustness. The most significant approach in this area is the SMOTE algorithm.[10] It was designed to balance the training set distribution by creating new synthetic examples of the minority class through the interpolation among instances from a given neighborhood. Since the addition of these novel examples may lead to overgeneralization, SMOTE is sometimes used in synergy with cleaning techniques, such as SMOTE+ENN.[6]

The hitch of preprocessing techniques is that they are not directly applicable to a multi-class scenario, as they are many minority classes. To overcome this gap, in[19] authors developed a methodology that applies a binarization scheme. Specifically, the original multi-class problem is divided into simpler binary subproblems by means of the OVO scheme,[31] i.e. a new problem is derived for each possible pair of classes. Then, for each subproblem obtained, the

SMOTE+ENN preprocessing technique may be applied prior to the learning stage. Finally, given a new query instance, all binary models are fired and their response is combined using the *Weighted Voting strategy* (WV),[34] which is computed as:

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij} \tag{1}$$

being $r_{ij} \in [0, 1]$ the confidence of the classifier discriminating classes i and j in favor of the former; whereas the confidence for the latter is computed by $r_{ji} = 1 - r_{ij}$ (if the classifier does not provide it).

Another possibility to balance the significance of the examples for the different classes on an imbalanced framework is to weight positively instances depending on their representation and to apply a cost-sensitive learning. In order to do so, we may consider a factor of N_i/N_{max}, being N_i the number of examples of the i-th class and N_{max} the number of examples for the majority class of the problem.[68]

A more sophisticated approach may be found in.[58] It is based on AdaBoost algorithm,[21] so that instances are iteratively weighted according to an ad-hoc formula based on the negative correlation learning.[39] To cope with the dataset imbalance, initial weights are assigned in inverse proportion to the number of instances in the corresponding class.

Finally, we must state that although the main core of the former solutions is devoted to cope with the skewed class distribution, they can also implicitly act over the existing overlapping among classes. For example, SMOTE preprocessing can strengthen the borderline of the minority clusters, whereas in conjunction with ENN it removes some instances in overlapped areas. The OVO procedure simplifies the borderlines in the multi-class scenario via a divide-and-conquer strategy. Finally, the boosting procedure focus on the hardest examples, i.e. those that are more likely to be overlapped.

3. EFIS-MOEA: A Novel Approach to Address Overlapping and Imbalance in Classification Tasks

In this section, we will first describe the core of the procedure (Section 3.1). Then, we will present the MOEA approach to search for the best parameters of the model, i.e. instances and features (Section 3.2). Finally, we will propose the use of an ensemble classifier by means of the solutions extracted from the MOEA, resulting on our final approach: EFIS-MOEA (Section 3.3).

3.1. *Core of the procedure*

The easiest way to address uneven class distributions is by balancing the training set. In this way, standard classifiers are no longer biased towards the majority class examples. To do so, a mechanism of instance selection is well suited to compensate the class ratio by removing majority instances. Furthermore, this scheme comprises additional advantages. First, when applied to all classes disregard their representation we seek to remove noisy and borderline instances that can degrade the individual recognition from these concepts. Obviously, this implies a kind of informed search to focus on those “low-quality” instances, such as in Training Set Selection.[27] Second, if we are addressing a large problem, this procedure allows the training process to be more efficient, and the output model can also be simpler.

On the other hand, we have stressed those data intrinsic characteristics that, in conjunction with the IR, can hinder the learning ability of the classifier. Specifically, the overlapping among classes is probably the most relevant issue for measuring the complexity of the problem to be solved.

Our hypothesis is that the use of feature selection will allow at simplifying the boundaries of the problem by limiting the influence of those features that may create difficulties for the discrimination process.

We consider that the synergy between both methodologies should result into a very successful methodology for addressing classification tasks in an imbalanced scenario. The ultimate goal of our proposal is to provide a rule-based model that *maximizes the recognition of all individual classes.* This must be achieved by focusing on the minority class clusters that are hard to identify. To do so, we focus on boosting the confidence of those rules associated with the former areas by means of the cleaning procedure, i.e. instance selection. In this way, a good criteria is to minimize the number of “bad” examples or, in other words, to *maximize the reduction of instances.* Additionally, and taking into account the findings made in,[42] the coverage of the rules may imply capturing some of the non-related classes. Specifically, in[1] authors made use of a neural network, and considered a combination between both a global and local scheme. The local scheme is based on the radius of coverage from a given instance, so that it follows similar idea than the one we stated previously.

We must stress that the estimation of the best-suited subset of instances and features is not trivial. Therefore, an optimization search procedure must be carried out in order to determine the former values. As stated at the beginning of this section, an MOEA methodology will be used. For the chromosome representation two genes will be considered, one (FS) for the feature selection and another one (IS) for the instance selection. Both are represented with a binary codification, in such a way that a 0 means that the variable (or instance) will not take part for generating the classification model, whereas a 1 value stands for the opposite case:

$$\begin{array}{l} FS = (a_1, a_2, \ldots, a_L) \\ IS = (x_1, x_2, \ldots, x_N), \end{array} \tag{2}$$

where L is the number of features, and N the number of instances in the training set (which can be preprocessed as stated previously).

Chromosomes will be evaluated jointly with aims at obtaining the best synergy between both characteristics, instead of optimizing them separately. This issue is based on the fact that it is not clearly defined which the best order for carrying our both processes is.

In the end, we must obtain a classifier with a high performance, being aware that all classes must be regarded with the same importance, but also a low degree of confidence related to misclassifications. Among all possibilities, the mean area under the curve (MAUC[30]) is the best suited metric to optimize the ability of the final model to separate pairs of classes in both binary and multi-class imbalanced classification.

In the binary case, let C_i and C_j be the two classes of a problem. The value $AUC(C_i, C_j)$ represents the probability that a randomly selected element from the first class also has a higher probability of being assigned to that class by the classifier compared to a randomly selected element of the other class ($A(C_i, C_j)$) and vice versa ($A(C_j, C_i)$). It is obtained as shown in Equation (3).

$$\mathrm{AUC}(C_i, C_j) = \frac{A(C_i, C_j) + A(C_j, C_i)}{2} \tag{3}$$

In our experiments, we follow[17] and calculate the AUC by approximating the continuous ROC-curve by a finite number of points. The coordinates of these points in ROC-space are taken as false positive and true positive rates obtained by varying the threshold of the probability above which an instance is classified as positive. The curve itself is approximated by linear interpolation between the calculated points. The AUC can therefore be determined as the sum of the areas of the successive trapezoids. This method is referred to as the trapezoid rule and is also described in e.g.[46]

Finally, MAUC is computed as the macro-average of the pairwise AUC values of all pairs of classes (see Equation (4)).

$$\mathrm{MAUC} = \frac{2}{m(m-1)} \sum_{i<j} AUC(C_i, C_j) \tag{4}$$

As baseline classifier we will make use of the C4.5 decision tree[52] for several reasons. The first one is its wide use in classification with imbalanced data, so that we may carry out a fair comparative versus the state-of-the-art. The second one is its efficiency, since we need to perform a large number of evaluations throughout the search process. Then, it is important the base model to be particularly quick for not biasing the global complexity of the methodology. It can be also applied to both the binary-class and multi-class scenarios without modifying its working procedure. Finally, its properties make it a common baseline classifier to be embedded into ensemble learning approaches.[14,54]

3.2. *MOEA approach*

In this research we aim at maximizing the performance while minimizing the number of instances used to generate the model, as stated in Eq. (5). For this reason, we propose to use as basis of the optimization an MOEA. In addition to the former, the goodness of this decision is two-fold: (1) we take advantage of the wider exploration capabilities of this type of technique, and (2) we allow the selection among a set of different solutions, depending on the user's requirements.

$$\begin{array}{l} OBJ_1 : M{-}AUC \\ OBJ_2 : RED = N - \sum_{i=0}^{N-1} IS_i; \end{array} \tag{5}$$

Specifically, we will make use of the NSGA-II algorithm[12] to implement our model. The fitness evaluation of this approach is based on both the Pareto ranking and a crowding measure. Ranking is used to organize solutions of the population according to their dominance degrees, i.e. rank 1 for non-dominated solutions (Pareto front), rank 2 for solutions dominated by those in rank 1, but that are still "better" than remaining solutions, and so forth. Crowding distance is used to create a total ordering among chromosomes, giving a higher fitness value to those solutions that are spread along the Pareto line. Another interesting feature of this methodology is the elitist generation update procedure. Specifically, the steps for NSGA-II are shown next:

procedure NSGA-II
  P_0 = initial population
  $Q_0 = 0$ ▷ offspring population
  repeat
    $R_t = P_t + Q_t$
    Evaluate(R_t) ▷ for all objective functions
    Generate all non-dominated fronts $F = (F_1, F_2, ...)$ of R_t.
    Initialize $P_{t+1} = 0$ and $i = 1$.
    repeat
      Calculate crowding-distance in F_i.
      Include i-th non-dominated front in the parent population.
      Check the next front for inclusion.
      Sort in descending order using crowded-comparison operator.
      Choose the first $(M - |P_{t+1}|)$ elements of F_i. ▷ M = Size front
      i = i + 1
    until parent population is filled.
    Use selection, crossover and mutation to create a new population Q_{t+1}.
    t = t + 1
  until t == Maximum generations
end procedure

Next, we describe in detail the different components selected for our current approach:

(1) *Initial population:* The initial population is formed of random chromosomes except for one that is taken to have all its genes set to 1 in order to represent the full training set.
(2) *Evaluation Mechanism:* First, Rank 1 is assigned to all non-dominated solutions in the current population, which are then tentatively removed. The former procedure is iterated until ranks are assigned to all solutions. Among solutions with the same rank, an additional criterion called a crowding measure is taken into account. Specifically, it computes the distance between its adjacent solutions with the same rank in the objective space, so that less crowed solutions are preferred.
(3) *Selection Procedure:* Binary tournament is used based on the fitness values, until the set of offspring solutions is full.
(4) *Crossover Operator:* The Heterogeneous Uniform Crossover (HUX) is used, since we are considering binary chromosomes. This operator interchanges exactly half of the different genes between both selected individuals.
(5) *Mutation Operator:* We use the "Bit flip" mutation in which each gene is changed from 0 to 1 and vice versa, with a certain probability.
(6) *Elitism:* Current and offspring populations are merged and the best solutions are maintained for the next population.

3.3. *EFIS-MOEA algorithm*

Ensemble-based classifiers, also known as multiple classifier systems,[50] are composed by a set of classifiers, with aims at solving a particular learning task. They have their basis on gathering several opinions to reinforce the support of the decision making process. It has been shown that the global combination of classifiers in ensemble learning improves the predictive performance of a single model, i.e. to obtain a better generalization.[63]

The advantage from the use of an MOEA approach is that it allows us to build an ensemble model by combining the C4.5 different decision tree models learned from all the training sets obtained after the optimization procedure. This design allows us to reinforce the capabilities from the classifiers extracted from each of the non-dominated solutions obtained in the Pareto into a single "Decision Forest".[54]

It is important to point out that for the success of this methodology, two main principles must be accomplished: (1) predictive performance; and (2) diversity.

The first issue implies the synergy of individual trees with a high predictive performance. Regarding this fact, we must consider that we are addressing different vectors obtained in the optimization, i.e. from the most accurate approach (best solution for M-AUC), to the "simplest" model (best solution for the number of instances). However, we must stress that this last case does not necessarily represents a trivial solution (M-AUC = 0.5), since it depends on both the characteristics of the problem and the focus of the search.

Additionally, we have pointed out that in order to make the ensemble to be accurate, individual trees should be sufficiently different from each other.[48,64] In order to accomplish this goal, training samples are usually manipulated. This is exactly the procedure followed by EFIS-MOEA, in which we reduce the original training data into smaller sets by horizontal (instance selection) and vertical (feature selection) partitions. For C4.5, these variations on the training set may result in a major change in the model. Furthermore, due to the use of the crowding measure of NSGA-II, the diversity of the components in the Pareto is guaranteed.

When a query instance arrives this system, each classifier will output its confidence degrees for each possible class. Finally, the label of the instance will be given as the class with the highest sum of confidences:

$$\begin{aligned} Class &= arg \max_{i=1,\ldots,m} S_i \\ S_i &= \textstyle\sum_{j=1 \leq K} Conf_{ji} \end{aligned} \tag{6}$$

where m is the number of classes, K the number of elements of the ensemble, and $Conf_{ji}$ the confidence degree of the j-th classifier for label i.

In order to determine the goodness of EFIS-MOEA, we will also consider the behavior shown by the classifier with the highest precision, i.e. the one that achieves the best results with respect to OBJ_1 (M-AUC). This particular case of our proposed approach will be named as 1-FIS-MOEA.

Additionally, before the use of the NSGA-II procedure, three different approaches to address imbalance are considered in synergy with EFIS-MOEA:

(1) **None**: Acting directly over the original training set. This is the simplest and most straightforward approach that leaves the MOEA approach to both balancing the data for the learning stage and cleaning noisy instances from all classes to enhance the problem description.

(2) **Weighting**: Update the training set by applying different weights to the instances in accordance to their distribution. Values are computed as N_i/N_{max}, with N_i being the number of examples of the i-th class and N_{max} the number of examples for the majority class of the problem. The idea is to take into account the a priori class distribution for boosting the recognition of the minority class instances. Therefore, the instance selection carried out in the MOEA will be mainly designed to remove noisy or redundant instances for improving the performance.

(3) **SMOTE**: Using SMOTE as oversampling preprocessing prior to the learning stage. This approach follows the same scheme as the previous case, i.e. to compensate for the uneven class distribution and to remove both original and synthetic instances that can hinder the classification ability of the algorithm. We must state that due to the nature of this approach its use is limited to the binary class case study.

Finally, we have depicted in Figure 1 the whole work-flow of the EFIS-MOEA proposal, for the sake of summarizing all steps.

4. Experimental Framework

This section includes the complete set up for the experimental analysis. Firstly, we present the datasets selected for both the binary and multi-class case studies (Section 4.1). Then, we will include the parameters selected for our proposal and the algorithms used for comparison (Section 4.2). Finally, a description about the statistical tests for adding support to the extracted conclusions is presented (Section 4.3).

4.1. *Binary and multi-class datasets*

The binary-class benchmark problems selected for our study, in which the name, number of examples (#Ex.), number of attributes (#Atts), IR and F1 metrics are shown in Table 1. A number of 66 datasets have been selected, as they comprise the standard experimental framework used in our studies on the topic.[4,25,40] Datasets in this table are presented in increasing order with respect to their imbalance ratio (IR) values.[49]

Regarding values of the F1 metric, these problems can be divided into two folds: (1) a number of 30 problems with a low degree of overlapping, considering $F1 > 1.5$; (2) 36 problems with a high degree of overlapping, when $F1 < 1.5$. In accordance with the former properties, this last set contains the hardest problems to be addressed.

Next, Table 2 shows the 24 multi-class imbalanced datasets, where the IR is computed as the ra-

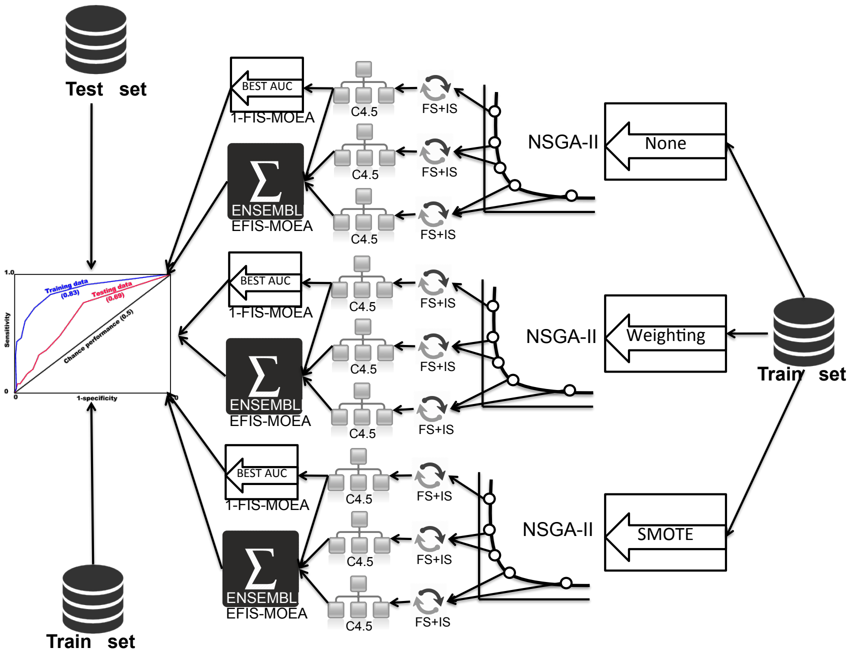

Figure 1. Complete work-flow of the EFIS-MOEA algorithm. Procedure starts at the rightmost side with the training set as input. Depending on the preprocessing, 3 schemes (None, Weighting and SMOTE) are considered. NSGA-II is then used to optimize the number of features and instances using C4.5 as baseline classifier. Finally, the output model is either the decision tree with the best AUC or an ensemble among all solutions, which are used to act on the test set

tio between the class with the highest representation and the one with the lowest one. The F1 metric has been obtained as the average of the values computed by pairs of classes. We must point out that for this task we have considered only half of the total pairs. This choice has been made for the sake of stressing the most difficult classes of the problem, as well as for avoiding those ones that are linearly separable. In addition, it is shown the distribution of examples among classes for the sake of considering not only the IR, but also those problems with multi minority/majority classes. We must stress that this set of problems implies the same experimental conditions used in one of our latest research on the topic.[19]

These problems have been downloaded from KEEL dataset repository.[3] We must stress that all datasets comprise real case studies originally from UCI repository,[38] varying in complexity and inner characteristics, as it has been highlighted by the IR and F1 metrics. A wider description on their characteristics can be found in its associated Website at `http://www.keel.es/datasets.php`.

The estimates for the AUC metric will be obtained by means of a Distribution Optimally Balanced Stratified Cross-Validation (DOB-SCV), as suggested for working in imbalanced classification.[41] DOB-SCV avoids dataset shift,[40,41] which hinders the results obtained in the experimental analysis. This procedure is carried out using 5 folds, aiming to include enough minority class instances in the different folds. In this way, we avoid additional problems in the data distribution, especially for highly imbalanced datasets. In accordance with the stochastic nature of the learning methods, each one of the 5-fold cross-validation is run 3 times. Therefore, experimental results for each method and dataset are computed with the average of 15 runs.

Finally, experiments have been carried out under a computer with an Intel(R) Core(TM) i7 CPU 930 microprocessor (4 cores/8 threads, 2.8 GHz, 8 MB Cache) with 24 GB of DDR2 RAM memory and using CentOS 6.4. The maximum Java heap space reserved for each execution was only 1GB.

4.2. *Algorithms and Parameters*

As stated in Section 3.1, in order to analyze the behavior of our proposed EFIS-MOEA methodology, we have selected the C4.5 decision tree[52] to induce the classification rules. The construction of the tree

Table 1. Summary description of binary-class imbalanced datasets used

Name	**#Ex.**	**#Atts.**	**IR**	**F1**	**Name**	**#Ex.**	**#Atts.**	**IR**	**F1**
glass1	214	9	1.82	0.1897	glass04vs5	92	9	9.22	1.5420
ecoli0vs1	220	7	1.86	9.7520	ecoli0346vs5	205	7	9.25	1.5950
wisconsin	683	9	1.86	3.5680	ecoli0347vs56	257	7	9.28	1.1300
pima	768	8	1.90	0.5760	yeast05679vs4	528	8	9.35	1.0510
iris0	150	4	2.00	16.8200	ecoli067vs5	220	6	10.00	1.6920
glass0	214	9	2.06	0.6492	vowel0	988	13	10.10	2.4580
yeast1	1484	8	2.46	0.2422	glass016vs2	192	9	10.29	0.2692
vehicle2	846	18	2.52	0.1691	glass2	214	9	10.39	0.3952
vehicle1	846	18	2.52	0.3805	ecoli0147vs2356	336	7	10.59	0.5275
vehicle3	846	18	2.52	0.1855	led7digit02456789vs1	443	7	10.97	1.9570
haberman	306	3	2.68	0.1850	ecoli01vs5	240	6	11.00	1.0490
glass0123vs456	214	9	3.19	3.3240	glass06vs5	108	9	11.00	1.3900
vehicle0	846	18	3.23	1.1240	glass0146vs2	205	9	11.06	0.3487
ecoli1	336	7	3.36	2.6500	ecoli0147vs56	332	6	12.28	0.9124
newthyroid2	215	5	4.92	3.5790	cleveland0vs4	177	13	12.62	1.3500
newthyroid1	215	5	5.14	3.5790	ecoli0146vs5	280	6	13.00	1.3400
ecoli2	336	7	5.46	1.8260	ecoli4	336	7	13.84	3.2470
segment0	2308	19	6.01	1.7980	shuttle0vs4	1829	9	13.87	0.3534
glass6	214	9	6.38	2.3910	yeast1vs7	459	8	13.87	12.9700
yeast3	1484	8	8.11	2.7510	glass4	214	9	15.47	1.4690
ecoli3	336	7	8.19	1.5790	pageblocks13vs4	472	10	15.85	1.5470
pageblocks0	5472	10	8.77	0.5087	abalone918	731	8	16.68	0.6320
ecoli034vs5	200	7	9.00	1.6320	glass016vs5	184	9	19.44	1.8510
yeast2vs4	514	8	9.08	1.5790	shuttle2vs4	129	9	20.50	12.1300
ecoli067vs35	222	7	9.09	0.9205	yeast1458vs7	693	8	22.10	0.1757
ecoli0234vs5	202	7	9.10	1.6180	glass5	214	9	22.81	1.0190
glass015vs2	506	8	9.12	0.1375	yeast2vs8	482	8	23.10	1.1420
yeast0359vs78	172	9	9.12	0.3113	yeast4	1484	8	28.41	0.7412
yeast0256vs3789	1004	8	9.14	1.6350	yeast1289vs7	947	8	30.56	0.3660
yeast02579vs368	1004	8	9.14	0.6939	yeast5	1484	8	32.78	4.1980
ecoli046vs5	203	6	9.15	1.6030	yeast6	1484	8	39.15	2.3020
ecoli01vs235	244	7	9.17	1.1030	ecoli0137vs26	281	7	39.15	1.9670
ecoli0267vs35	244	7	9.18	0.9129	abalone19	4174	8	128.87	0.5295

is carried out in a top-down manner. The normalized information gain (difference in entropy) is used to select the attribute that better splits the data in each node.

As introduced in Section 2, several approaches from the state-of-the-art have been chosen in order to contrast the results. Particularly, the SMOTE+ENN preprocessing approach[6] for binary class problems and multi-class problems (using the binarization scheme[19]), and both Global-CS[68] and AdaBoost.NC[58] for the multi-class case study. Additionally, we have selected Random Forest[9] as a robust algorithm for standard classification tasks. Finally, we must recall that the behavior of EFIS-MOEA will be also contrasted versus 1-FIS-MOEA, i.e. the classifier obtained by selecting the best solution of the Pareto in terms of M-AUC.

The parameters used for each algorithm are shown in Table 3. These values are common for all problems. They were selected according to the recommendation of the corresponding authors and it is also the default setting of the parameters included in the KEEL[a] software suite,[3] which we have used to develop our experiments, except for Random Forest which is based on the Weka implementation.[61] In the case of the MOEA, we have make use of the *jmetal* library.[16]

[a]`http://www.keel.es`

Table 2. Summary description of multi-class imbalanced datasets used

id	Name	#Ex.	#Atts.	#Cl.	IR	F1	Class distribution
Aut	Autos	150	25	6	16.00	1.2486	3/20/48/46/29/13
Bal	Balance	625	4	3	5.88	0.1352	288/49/288
Cle	Cleveland	297	13	5	12.62	0.2350	164/55/36/35/13
Con	Contraceptive	1473	9	3	1.89	0.0769	629/333/511
Der	Dermatology	358	33	6	5.55	9.2647	111/60/71/48/48/20
Eco	Ecoli	336	7	8	71.50	0.8518	143/77/2/2/35/20/5/52
Fla	Flare	1066	11	6	7.70	0.8614	331/239/211/147/95/43
Gla	Glass	214	9	6	8.44	1.3186	70/76/17/13/9/29
Hay	Hayes-Roth	160	4	3	2.10	0.0980	160/65/64/31
Led	Led7digit	500	7	10	1.54	4.2275	45/37/51/57/52/52/47/57/53/49
Lym	Lymphography	148	18	4	40.5	7.4840	2/81/61/4
New	New-thyroid	215	5	3	5.00	3.4007	150/35/30
Nur	Nursery	12690	8	5	2160.0	0.3957	2/4320/4266/328/4044
Pag	Page-blocks	5472	10	5	175.46	1.5015	4913/329/28/87/115
Pos	Post-operative	87	8	3	62	0.0000	62/24/1
Sat	Satimage	6435	36	7	2.45	2.7252	1533/703/1358/626/707/1508
Shu	Shuttle	58000	9	5	4558.6	3.1322	45586/49/171/8903/3267/10/13
Spl	Splice	3190	60	3	2.16	1.2621	767/768/1655
Thy	Thyroid	7200	21	3	40.16	0.8106	166/368/6666
Win	Wine	178	13	3	1.48	3.8438	59/71/48
Wqr	Wine-Q.-Red	1599	11	6	68.10	0.3680	10/53/681/638/199/18
Wqw	Wine-Q.-White	4898	11	7	439.60	0.2462	20/163/1457/2198/880/175/5
Yea	Yeast	1484	8	10	92.60	1.1171	244/429/463/44/51/163/35/30/20/5
Zoo	Zoo	101	16	7	10.25	1.9311	41/13/10/20/8/5/4

Table 3. Parameter specification for the algorithms employed in the experimentation

Algorithm	*Parameters*
MOEA	Pop. size = 60 individuals, Evaluations = 6000 Crossover Prob. = 0.8, Mutation Prob. = 0.025
C4.5	Prune = True, Confidence level = 0.25 Minimum number of item-sets per leaf = 2
SMOTE+ENN	Balance Ratio = 1, Neighbors for SMOTE = 5 Neighbors for ENN = 3, Distance = HVDM[60]
AdaBoost.NC	$\lambda = 2$ (penalty strength), #classifiers = 51
RandomForest	#classifiers = 51, depth = #vars

4.3. *Statistical tests for performance comparison*

In this paper the hypothesis testing techniques will be used to provide statistical support for the analysis of the results.[28] Specifically we will use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with these types of tests.[28] Any interested reader can find additional information on the Website `http://sci2s.ugr.es/sicidm/`.

First of all, we will use the Friedman Aligned test[28] to show at a first glance how good a method is with respect to its partners. In addition, this test also provides information to check whether there are significant differences among the results. When the null hypothesis of equality is rejected, the Holm post-hoc test[33] finds which algorithms are statistically different to a selected control method in a $1 \cdot n$ comparison.

The Friedman Aligned test[28] will be used to check whether there are significant differences among the results, and the Holm post-hoc test[33] in order to find which algorithms reject the hypothesis of equality with respect to a selected control method in a $1 \cdot n$ comparison. We will compute the adjusted p-value (APV) associated with each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. This value differs from the standard p-value in the sense that it determines unequivocally whether the null hypothesis of equality is rejected at a significance level α.

Regarding pairwise comparisons, we will make use of Wilcoxon signed-rank test[59] to find out whether significant differences exist between a pair of algorithms. This procedure computes the differences between the performance scores of the two classifiers on each one of the available datasets (N_{ds}). The differences are ranked according to their absolute values, from smallest to largest, and average ranks are assigned in case of ties. We call R^+ the sum of ranks for the datasets on which the second algorithm outperformed the first, and R^- the sum of ranks for the opposite. Let T be the smallest of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for N_{ds} degrees of freedom (Table B.12 in[65]), the null hypothesis of equality of means is rejected.

5. Experimental Results

We divide this section into two different studies for binary (Section 5.1) and multi-class problems (Section 5.2). As stated in the introduction of this paper, the first case will serve us as an initial case study in order to analyze the behavior of EFIS-MOEA with respect to the overlapping between classes. Then, we will determine the suitability of EFIS-MOEA in a more significant framework, i.e. for multiple classes, in which the recognition of the boundaries becomes harder because of the wider amount of overlapping among classes.

It is important to remark that all the findings extracted throughout this experimental analysis are based in the output of statistical tests, i.e., average ranking and p-values. However, we have also included the average performance results to provide a reference of the global quality of the different methodologies selected for this study. In this way, any interested researcher can be aware of the performance shown in this work in contrast with their own methods.

5.1. *Analysis of the behavior of EFIS-MOEA in binary classification*

Our first part of the experimental study is focused on addressing imbalanced datasets with two classes. To do so, we will proceed as follows:

(1) We will start by contrasting the different versions designed for the feature and instance selection. These different approaches were suggested in order to address the imbalanced class problem in synergy with EFIS-MOEA.
(2) Once the best method has been chosen, we will contrast the performance of EFIS-MOEA versus C4.5 and C4.5-SMOTE-ENN into three different scenarios: all datasets, datasets with high overlapping, and datasets with both a high overlapping and imbalance.

5.1.1. *Analyzing the preprocesing approach for EFIS-MOEA*

We aim at analyzing the best approach among the three versions suggested in Section 3.1 for the modifying the training set prior to the evolutionary optimization of the features and instances. Specifically, the options were using the standard set (None), applying weights (Weighting), or to use SMOTE to balance the class distribution.

Average experimental results in training and test using all datasets and considering AUC metric are shown in Table 4. Results for the three aforementioned versions are given in different rows, according to the “Preprocessing” column. This table also includes the statistical comparison, showing the average ranks computed by the Friedman aligned test, and the APVs obtained by means of a Holm test. We explicitly stress whether there are statistical differences with a degree of confidence higher than 95% (symbol $*$) or 90% (symbol $+$). We also show the number of wins/ties/loses (W/T/L) for each approach in comparison with the control method, i.e., that with the highest rank. This will serve as a complementary measure to the p-value for pointing out the degree of improvement achieved by EFIS-MOEA.

We must highlight the strong synergy between the instance generation step (made by SMOTE) and the instance selection of EFIS-MOEA. First, the resampling procedure allows balancing of the class distribution so that initial models (at the beginning of the evolutionary search) are more robust. In addition, it acts on the minority class clusters by spreading the borderline to facilitate their recognition in overlapped areas. Finally, EFIS-MOEA implies a data-cleaning step for both these novel synthetic instances and those examples that can degrade the learning ability of the classifier. This combination of methodologies has been already stressed in the specialized literature, especially regarding the high number of approaches that follow this scheme.[40]

These conclusions are supported by the statistical analysis, from which 1-FIS-MOEA and EFIS-MOEA plus SMOTE obtain significant differences

Table 4. Average training and test results (AUC), ranks (Friedman aligned) and APVs (Holm test) for **the three versions for EFIS-MOEA**.

GA-Approach	Preprocesing	AUC Train	AUC Test	Ranking	APV (Holm test)	W/T/L
s	None	.9610 ±.0097	.8219 ±.0745	113.947 (2)	.00000*	23/0/43
1-FIS-MOEA	Weighting	**.9909 ±.0033**	.8205 ±.8205	116.7652 (3)	.00000*	23/0/43
	SMOTE	.9844 ±.0058	**.8407 ± .0647**	67.7879 (1)	******	-/-/-
	None	.9568±.0151	.8687 ±.0570	117.12 (3)	.00000*	26/0/40
EFIS-MOEA	Weighting	**.9873 ±.0084**	.8694 ±.0589	110.44 (2)	.00007*	27/0/39
	SMOTE	.9803±.0087	**.8803 ± .0513**	70.94 (1)	*****	-/-/-

Table 5. Average training-test results (AUC), ranks (Friedman aligned) and APVs (Holm test) for SMOTE+ENN, EFIS-MOEA and 1-FIS-MOEA (both with SMOTE preprocessing) for binary imbalanced datasets.

Scenario	Method	AUC Train	AUC Test	Ranking	APV (Holm test)	W/T/L
	C4.5	.9510 ± .0253	.8892 ± .0661	94.00 (4)	.00000*	4/0/26
Low overlap	C4.5-SMOTE+ENN	.9797 ± .0090	.9263 ± .0472	53.30 (2)	.00737*	8/0/22
(F1 > 1.5) [30])	1-FIS-MOEA	**.9943 ± .0031**	.9195 ± .0514	65.47 (3)	.00005*	3/0/27
	EFIS-MOEA	.9906 ± .0072	**.9439 ± .0414**	29.23 (1)	*****	-/-/-
	C4.5	.8437 ±.0454	.7352 ±.0726	113.78 (4)	.00000*	2/0/34
High overlap	C4.5-SMOTE+ENN	.9338 ±.0182	.7817 ±.0740	71.61 (2)	.00000*	3/0/33
(F1 < 1.5) [36])	1-FIS-MOEA	**.9761 ±.0081**	.7749 ±.0757	79.22 (3)	.00000*	0/0/36
	EFIS-MOEA	.9717± .0100	**.8273± .0596**	25.39 (1)	*****	-/-/-

versus the remaining versions. We must also stress the high confidence degree associated with each comparison (p-values are close to zero in all cases).

5.1.2. *Comparison versus the state-of-the-art*

In this part of the study, we will contrast the performance of EFIS-MOEA versus C4.5 and C4.5-SMOTE+ENN under two different scenarios: (1) for two-class imbalanced datasets with low overlapping (the easiest problems); and (2) for binary imbalanced datasets with high overlapping (the hardest problems).

Table 5 includes the average performance values for training and test partitions together with their standard deviation. We also show the ranking (computed by Friedman aligned method), p-values (with post-hoc Holm test) and wins/ties/loses (W/T/L) for each method with respect to the best one. This table is divided into three parts as stated above, where the number of datasets for each case study is given between brackets. Additionally, Table A.1 in the appendix of the manuscript includes the complete table of results for all 66 problems. We must recall that in accordance to the results obtained in the previous part of this study, we apply SMOTE to the training set prior to EFIS-MOEA.

From these experimental results, our proposed EFIS-MOEA is the approach that presents the best behavior overall. This is supported by both the high average results in AUC for the test partitions, and the top ranking achieved in both case studies. We also observe an overfitting problem for 1-FIS-MOEA. This is due to the fact that the best solution for AUC is always selected. Indeed, when we apply our EFIS-MOEA extension, the collaboration among all solutions allows mitigating this negative effect.

Finally, the synergy between feature selection and instance selection boosts the performance of our approach versus the oversampling and cleaning carried out by SMOTE+ENN, especially for highly overlapped problems in which the absolute differences are almost 4 points on average.

We conclude that this interesting behavior is due to the fact that true hits are associated with high confidence values (around 1.0), whereas misclassifi-

Table 6. Average training and test results ($M\text{-}AUC$), ranks (Friedman aligned) and APVs (Holm test) for EFIS-MOEA and the state-of-the-art for multi-class imbalanced datasets.

Method	AUC Train	AUC Test	Ranking	APV (Holm test)	W/T/L
C4.5	.9006 ± .0141	.8157 ±.0297	102.54 (6)	.00000*	2/0/22
OVO-SMOTE+ENN	.9369 ± .0136	.8292 ± .0352	74.58 (5)	.00725*	6/0/19
Global-CS	**.9726 ± .0060**	.8324 ± .0346	72.48 (3)	.01206*	4/0/20
AdaBoost.NC	.9530 ± .0147	.8233 ± .0319	69.06 (2)	.02597*	8/0/16
1-FIS-MOEA	.9715 ±.0041	.8299 ±.0355	74.08 (4)	.00820*	5/0/19
EFIS-MOEA	.9691 ± .0058	**.8441 ± .0322**	42.25 (1)	*****	-/-/-

Table 7. Average training and test results ($M\text{-}AUC$), ranks and p-values (Wilcoxon Test) for EFIS-MOEA versus Random Forest in multi-class imbalanced datasets.

Method	AUC Train	AUC Test	Ranks	p-value (Wilcoxon)	W/T/L
RandomForest	.9648 ± .0042	.8382 ±.0330	128.0	.52032	11/0/13
EFIS-MOEA	**.9691 ± .0058**	**.8441 ± .0322**	172.0	*****	-/-/-

cations are associated with low confidences (around 0.5). This way, the final Area Under the ROC is positively weighted for all case studies.

5.2. *Analysis of the behavior of EFIS-MOEA in multi-class datasets*

Regarding the different preprocessing approaches to be applied prior to the MOEA procedure, i.e. None, Weighting and SMOTE, in our previous research on the topic[19] we stressed that using SMOTE in datasets with multiple classes is not the better choice. On the contrary, we suggested the use of an instance weighting approach for addressing multi-minority and multi-majority classes. In this way, the significance of all classes are balanced and the final system obtained will be able to correctly classify them disregard their initial representation in the problem. Even in the case of using the oversampling approach, the size of the multi-class problems will be significantly increased. In this sense, the search space for EFIS-MOEA will become too large in order to obtain accurate solutions. The former analysis supports the use of the "*Weighting*" version for the preprocessing of the training set in the context of multi-class imbalanced problems.

We have compiled the average training and test performance values together with the statistical validation of the former into a unique table of results (Table 6). The different algorithms are shown by rows, whereas by columns we include the M-AUC values both in training and test (with the standard deviation), the ranking value and position (computed by Friedman aligned procedure), the APVs (obtained by a Holm test), and the number of wins/ties/loses (W/T/L) in comparison to EFIS-MOEA.

The findings extracted from the results obtained in this case study are similar to those given for binary-class problems. The goodness shown by our EFIS-MOEA approach is clear, as it is able to outperform all algorithms selected for comparison. The statistical results provide a strong support to the excellent capabilities for our approach. By taking advantage from all the solutions discovered in the optimization stage into an ensemble, results are significantly boosted with respect to the best classifier found in the MOEA search, i.e. 1-FIS-MOEA, which suffers from the curse of overfitting.

The full results among all datasets are shown in Table A.2 in the appendix of this work. We must stress the quality shown by EFIS-MOEA for the hardest problems, i.e. those with a high overlapping ($F1 < 1.5$). In this subset, our proposal achieves the highest performance in contrast with the state-of-the-art in multi-class imbalanced classification in almost half of the datasets (7 out of 16). Therefore, the significance of our methodology for addressing the overlapping among classes has been clearly established.

A final comparison versus one state-of-the-art in standard classification was carried out in Table 7. Specifically, we have applied Random Forest[9] to contrast the quality of our novel proposal versus probably one of the highest performing rule-based classifiers currently. First, we are able to stress the capabilities of EFIS-MOEA based on the average results in M-AUC. Additionally, the sum of ranks achieved in a Wilcoxon pairwise comparison, and the p-value associated to the statistical test, implies that our approach is competitive in terms of overall performance.

In order to complement our study, we show in Table 8 some interesting information from the EFIS-MOEA model for every dataset. Specifically, this table includes by columns the total number of classifiers of the ensemble ("#Classif."), measured as the number of solutions from the Pareto, the average number of selected features ("#Feats.") and the percentage of reduction from the total ("RedFS"), the number of selected instances ("#Inst.") as well as the percentage of reduction from the initial size ("RedIS"), and the elapsed training time.

Table 8. Information about the number of classifiers, variables and instances selected, and elapsed training time for EFIS-MOEA for multi-class imbalanced datasets.

Data	#Class.	#Vars.	RedFS	#Inst.	RedIS	Tr. Time
aut	35.0	14.4	42.40	50.6	60.26	0:00:28.2
bal	41.4	4.0	0.00	262.2	47.56	0:00:44.8
cle	47.0	9.0	30.77	133.2	43.94	0:00:46.7
con	47.8	7.8	13.33	731.4	37.93	0:07:17.1
der	11.8	19.4	42.94	62.2	78.27	0:00:48.3
eco	36.0	5.4	22.86	130.4	51.49	0:00:34.2
fla	40.0	9.6	12.73	458.6	46.26	0:01:59.9
gla	42.0	6.8	24.44	82.0	52.10	0:00:24.8
hay	32.8	3.0	25.00	28.2	77.95	0:00:07.0
led	30.6	6.8	2.86	162.2	59.40	0:00:48.5
lym	27.4	11.4	36.67	43.6	63.19	0:00:16.0
new	15.2	2.4	52.00	32.6	81.05	0:00:06.9
nur	50.6	7.2	10.00	6523.4	37.08	0:17:58.5
pag	29.2	6.0	40.00	2297.8	47.51	0:39:38.1
pos	52.0	5.8	27.50	25.6	63.15	0:00:06.7
sat	59.8	27.4	23.89	3574.6	30.56	3:14:44.0
shu	18.4	5.6	37.78	22418.2	51.68	54:43:43.5
spl	41.4	40.0	33.33	1504.2	41.06	0:24:13.9
thy	10.8	11.8	43.81	2522.4	56.21	0:50:26.7
win	14.8	6.8	47.69	25.0	82.46	0:00:06.8
wqr	58.4	8.6	21.82	816.0	36.21	0:07:54.6
wqw	59.6	9.2	16.36	2616.4	33.23	1:26:08.1
yea	50.2	7.0	12.50	734.4	38.11	0:09:25.8
zoo	58.0	9.6	40.00	6.2	92.31	0:00:08.9
Avg.	37.9	10.2	27.53	1885.1	54.54	2:35:22.4

From this information, we can conclude the following:

- The number of classifiers that compose the ensemble is quite low on average, between 10 and 40 classifiers, which is the standard in this framework.[23] In comparison with AdaBoost.NC, which uses 51 classifiers in total, our approach only comprises an average of 36 classifiers.
- Regarding the dimensionality reduction, about 25% of the initial variables are considered for the learning stage, thus implying the necessity of carrying out the feature selection process for simplifying the borderline areas of the problem.
- Only half of the initial instances are finally used. Considering the boost in performance achieved, we may conclude that our methodology carried out the removal of "low-quality" instances that were hindering the classification ability of the learning algorithm.
- Regarding the elapsed training time, we observe that for most of the problems the time consumption is minimal (less than a minute). For larger problems (those with more than 1,000 instances) the computation time obviously increases; but there are only 3 cases out of 24 in which more than an hour is needed to generate the final model. In any case, a distributed mechanism to compute the evaluation function can enhance response times for those problems with a high number of examples.

6. Concluding Remarks

In this paper we have proposed EFIS-MOEA, a novel methodology to improve the classification ability of algorithms in two-class and multi-class imbalanced datasets. This approach has been designed under a double perspective: (1) removing instances that may hinder the classification ability; and (2) removing features to act on the overlapping areas. One of the main advantages of our novel methodology is its versatility, as it follows the same structure for both binary and multi-class problems, as well as to be embedded with any classifier.

The results obtained by EFIS-MOEA were very competitive, especially for highly overlapped problems. The selection of instances allowed rebalancing the training set as well as to clean the low quality data, i.e. noisy and redundant examples. In addition, feature selection simplified the boundaries of the problem to manage the aforementioned overlapping issue. The behavior of EFIS-MOEA is excelled as it was shown to outperform the state-of-the-art algorithms, especially the AdaBoost.NC approach, which has been stressed as the most competitive approach in this context. Additionally, when contrasted with more general classifiers such as Random Forest, it also reaches a superior performance in terms of AUC.

As future work, we propose focusing on the final ensemble generated by the MOEA, carrying out an optimal selection of classifiers.[11,25] Another topic of high interest is to analyze the scalability of our approach to address Big Data problems in terms of number of instances, features and also classes. This may imply to act directly on the evolutionary scheme,[45] or to redesign the whole methodology to embed it in a distributed MapReduce methodology.[20]

Acknowledgments

This work was supported by the Spanish Ministry of Science and Technology under projects TIN2014-57251-P and TIN2015-68454-R; the Andalusian Research Plan P11-TIC-7765; and both the University of Jaén and Caja Rural Provincial de Jaén under project UJA2014/06/15.

Appendix A Full Tables of Results

This final section shows Tables A.1 and A.2, which include the full results in training and test for the probabilistic AUC and M-AUC metrics in both in two-class and multi-class imbalanced problems.

Bibliography

1. M. Ahmadlou and H. Adeli. Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering*, 17(3):197–210, 2010.
2. M. A. H. Akhand and K. Murase. Ensembles of neural networks based on the alteration of input feature values. *International Journal of Neural Systems*, 22(1):77–87, 2012.
3. J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera. KEEL data–mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multi–Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.
4. S. Alshomrani, A. Bawakid, S.-O. Shim, A. Fernandez, and F. Herrera. A proposal for evolutionary fuzzy systems using feature weighting: Dealing with overlapping in imbalanced datasets. *Knowledge-Based Systems*, 73:1–17, 2015.
5. R. Barandela, J. S. Sanchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.
6. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behaviour of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.
7. M. L. Borrajo, B. Baruque, E. Corchado, J. Bajo, and J. M. Corchado. Hybrid neural intelligent system to predict business failure in small-to-medium-size enterprises. *International Journal of Neural Systems*, 21(4):277–296, 2011.
8. P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modelling under imbalanced distributions. *ACM Computing Surveys*, 49(2):31:1–31:50, 2016.
9. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
10. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over–sampling technique. *Journal of Artificial Intelligent Research*, 16:321–357, 2002.
11. L. F. S. Coletta, E. R. Hruschka, A. Acharya, and J. Ghosh. Using metaheuristics to optimize the combination of classifier and cluster ensembles. *Integrated Computer-Aided Engineering*, 22(3):229–242, 2015.
12. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
13. M. Denil and T. Trappenberg. Overlap versus imbalance. In *Proceedings of the 23rd Canadian conference on Advances in artificial intelligence (CCAI'10)*, volume 6085 of *Lecture Notes on Artificial Intelligence*, pages 220–231, 2010.
14. J.-F. Diez-Pastor, C. Garcia-Osorio, and J. J. R. Diez. Tree ensemble construction using a grasp-based heuristic and annealed randomness. *Information Fusion*, 20:189–202, 2014.
15. P. Domingos. Metacost: A general method for making classifiers cost–sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 155–164, 1999.
16. J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.
17. T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

18. A. Fernandez, M. J. del Jesus, and F. Herrera. Addressing overlapping in classification with imbalanced datasets: A first multi-objective approach for feature and instance selection. In K. Jackowski, R. Burduk, K. Walkowiak, M. Wozniak, and H. Yin, editors, *IDEAL*, volume 9375 of *Lecture Notes in Computer Science*, pages 36–44. Springer, 2015.
19. A. Fernandez, V. Lopez, M. Galar, M. J. del Jesus, and F. Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97–110, 2013.
20. A. Fernandez, S. Rio, V. Lopez, A. Bawakid, M. J. del Jesus, J. M. Benitez, and F. Herrera. Big data with cloud computing: An insight on the computing environment, mapreduce and programming framework. *WIREs Data Mining and Knowledge Discovery*, 4(5):380–409, 2014.
21. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *13th. Int. Conf. Mach. Learn. (ICML'96)*, pages 148–156, 1996.
22. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.
23. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for class imbalance problem: Bagging, boosting and hybrid based approaches. *IEEE Transactions on System, Man and Cybernetics Part C: Applications and Reviews*, 42(4):463–484, 2012.
24. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, 46(12):3460–3471, 2013.
25. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets. *Information Sciences*, 354:178–196, 2016.
26. M. Galar, A. Fernandez, E. Barrenechea, and F. Herrera. Empowering difficult classes with a similarity-based aggregation in multi-class classification problems. *Information Sciences*, 264:135–157, 2014.
27. S. Garcia, A. Fernandez, and F. Herrera. Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing*, 9:1304–1314, 2009.
28. S. García, A. Fernandez, J. Luengo, and 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(10):2044–2064, 2010.
29. V. Garcia, R. Mollineda, and J. S. Sanchez. On the k–NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis Applications*, 11(3–4):269–280, 2008.
30. D. J. Hand and R. J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.
31. T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.
32. T. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.
33. S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.
34. E. Hüllermeier and S. Vanderlooy. Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. *Pattern Recognition*, 43(1):128–142, 2010.
35. G. Iacca, F. Caraffini, and F. Neri. Multi-strategy coevolving aging particle optimization. *International Journal of Neural Systems*, 24(1), 2014.
36. N. Japkowicz and S. Stephen. The class imbalance problem: a systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450, 2002.
37. S. Jiang and S. Yang. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts. *IEEE Transactions on Cybernetics*, 46(2):421–437, 2016.
38. M. Lichman. UCI machine learning repository; university of california, irvine, school of information and computer sciences. `http://archive.ics.uci.edu/ml`, 2013.
39. B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *Proceedings of the 5th International Conference on Knownledge Discovery and Data Mining (KDD'99)*, pages 337–341, 1999.
40. V. Lopez, A. Fernandez, S. Garcia, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250(20):113–141, 2013.
41. V. Lopez, A. Fernandez, and F. Herrera. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Information Sciences*, 257:1–13, 2014.
42. A. Manukyan and E. Ceyhan. Classification of imbalanced data with a geometric digraph family. *Journal of Machine Learning Research*, 17:1–40, 2016.
43. A. Marcano-Cedeño, A. M. de la Barcena, J. Jiménez-Trillo, J. A. Piñuela, and D. Andina. Artificial metaplasticity neural network applied to credit scoring. *International Journal of Neural Sys-*

tems, 21(4):311–317, 2011.

44. A. Marcano-Cedeño, J. Quintanilla-Domínguez, and D. Andina. Breast cancer classification applying artificial metaplasticity algorithm. *Neurocomputing*, 74(8):1243–1250, 2011.
45. M. Martnez-Ballesteros, J. Bacardit, A. T. Lora, and J. C. Riquelme. Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets. *Integrated Computer-Aided Engineering*, 22(1):21–39, 2015.
46. S. J. Mason and N. E. Graham. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128(584):2145–2166, 2002.
47. H. D. Menéndez, D. F. Barrero, and D. Camacho. A genetic graph-based approach for partitional clustering. *International Journal of Neural Systems*, 24(3), 2014.
48. A. Omari and A. R. Figueiras-Vidal. Post-aggregation of classifier ensembles. *Information Fusion*, 26:96–102, 2015.
49. A. Orriols-Puig and E. Bernado-Mansilla. Evolutionary rule–based systems for imbalanced datasets. *Soft Computing*, 13(3):213–225, 2009.
50. R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.
51. R. C. Prati, G. E. A. P. A. Batista, and D. F. Silva. Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270, 2015.
52. J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kauffman, 1993.
53. Y. Ren, L. Zhang, and P. N. Suganthan. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, 11(1):41–53, 2016.
54. L. Rokach. Decision forest: Twenty years of research. *Information Fusion*, 27:111–125, 2016.
55. S. Rostami and F. Neri. Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm. *Integrated Computer-Aided Engineering*, 23(4):313–329, 2016.
56. J. Stefanowski. Dealing with data difficulty factors while learning from imbalanced data. In S. Matwin and J. Mielniczuk, editors, *Challenges in Computational Statistics and Data Mining*, volume 605 of *Studies in Computational Intelligence*, pages 333–363. Springer, 2016.
57. J. Villar, S. González, J. Sedano, C. Chira, and J. Trejo-Gabriel-Galan. Improving human activity recognition and its application in early stroke diagnosis. *International Journal of Neural Systems*, 25(4), 2015.
58. S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on System Man and Cybernetics: Part B*, 42(4):1119–1130, 2012.
59. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
60. D. Wilson and T. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
61. I. H. Witten, M. A. H. E. Frank, and C. J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Mateo, CA, 4th edition, 2017.
62. M. Wozniak, M. Graña, and E. Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.
63. X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge Information Systems*, 14(1):1–37, 2007.
64. X.-C. Yin, K. Huang, C. Yang, and H.-W. Hao. Convex ensemble learning with sparsity and diversity. *Information Fusion*, 20:49–59, 2014.
65. J. H. Zar. *Biostatistical Analysis*. Prentice Hall, Upper Saddle River, New Jersey, 1999.
66. Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, and A. Cichocki. Aggregation of sparse linear discriminant analyses for event-related potential classification in brain-computer interface. *International Journal of Neural Systems*, 24(1), 2014.
67. A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhangd. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.
68. Z.-H. Zhou and X.-Y. Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010.

Table A.1. Experimental results for C4.5, C4.5 with SMOTE+ENN (C4.5+S_ENN), 1-FIS-MOEA and EFIS–MOEA in training and test with AUC metric. Datasets are ordered according to the F1 metric in ascending order (from highly overlapped to linearly separable problems)

Dataset	IR	F1	C4.5		C4.5+S_ENN		1-FIS-MOEA		EFIS-MOEA	
			Tr	Tst	Tr	Tst	Tr	Tst	Tr	Tst
glass015vs2	9.12	0.1375	.8001	.6048	.9678	.7925	.9920	.6405	.9792	.7085
vehicle1	2.52	0.1691	.8984	.7228	.8912	.7658	.9719	.7354	.9833	.8365
yeast1458vs7	22.10	0.1757	.5000	.5000	.8893	.5637	.9740	.5966	.9545	.6099
haberman	2.68	0.1850	.6143	.5671	.7283	.6355	.8485	.6279	.8438	.6548
vehicle3	2.52	0.1855	.9133	.7229	.8785	.7540	.9691	.7160	.9821	.8139
glass1	1.82	0.1897	.8652	.7400	.8702	.7531	.9786	.7997	.9680	.8343
yeast1	2.46	0.2422	.7946	.6970	.8041	.7166	.8771	.7057	.8993	.7660
glass016vs2	10.29	0.2692	.9495	.7194	.9384	.6938	.9921	.6825	.9863	.7731
yeast0359vs78	9.12	0.3113	.6973	.5803	.9210	.7056	.9806	.6557	.9802	.7291
glass0146vs2	11.06	0.3487	.8900	.6753	.9613	.6890	.9920	.6820	.9868	.7953
yeast1vs7	13.87	0.3534	.7859	.5759	.9343	.6215	.9691	.5772	.9718	.6592
yeast1289vs7	30.56	0.3660	.6353	.6176	.9324	.6131	.9559	.6458	.9219	.6955
vehicle2	2.52	0.3805	.9940	.9430	.9894	.9462	.9943	.9533	.9965	.9815
glass2	10.39	0.3952	.8959	.6802	.9518	.7217	.9944	.6875	.9904	.7949
page-blocks0	8.77	0.5087	.9703	.9396	.9759	.9486	.9916	.9445	.9930	.9725
ecoli0147vs2356	10.59	0.5275	.9363	.8286	.9791	.8488	.9938	.8613	.9912	.8909
abalone19	128.87	0.5295	.5000	.5000	.9057	.5523	.9245	.5900	.8549	.6214
pima	1.90	0.5760	.8279	.7328	.8261	.7468	.9142	.7058	.9245	.7802
abalone9-18	16.68	0.6320	.6780	.5985	.9575	.6752	.9711	.6982	.9713	.7304
glass0	2.06	0.6492	.9480	.7890	.8862	.7916	.9787	.8048	.9700	.8674
yeast0256vs3789	9.14	0.6939	.7872	.7445	.9182	.7836	.9756	.7686	.9709	.8266
yeast4	28.41	0.7412	.7973	.7050	.8923	.7843	.9560	.7417	.9645	.8199
ecoli0147vs56	12.28	0.9124	.9263	.8167	.9754	.8292	.9961	.9170	.9941	.9362
ecoli0267vs35	9.18	0.9129	.8821	.8344	.9730	.8525	.9951	.8348	.9927	.8987
ecoli067vs35	9.09	0.9205	.8828	.8450	.9708	.8703	.9956	.8918	.9913	.9084
glass5	22.81	1.0190	.9841	.9463	.9902	.8890	.9990	.8709	.9974	.9380
glass06vs5	11.00	1.0490	.9949	.9947	.9886	.9775	.9996	.9609	.9952	.9639
yeast05679vs4	9.35	1.0510	.8338	.7360	.9267	.7760	.9850	.7719	.9843	.8553
ecoli01vs235	9.17	1.1030	.9433	.8469	.9814	.9279	.9961	.8646	.9903	.9229
vehicle0	3.23	1.1240	.9884	.9443	.9768	.9194	.9934	.9290	.9933	.9713
ecoli0347vs56	9.28	1.1300	.9088	.8193	.9717	.8041	.9954	.8732	.9930	.9169
yeast2vs8	23.10	1.1420	.5528	.4891	.9543	.7996	.9973	.7862	.9846	.8171
ecoli0146vs5	13.00	1.3400	.9405	.7786	.9863	.8942	.9965	.8443	.9950	.8978
cleveland0vs4	12.62	1.3500	.9147	.5808	.9624	.7457	1.0000	.8039	.9914	.8299
ecoli01vs5	11.00	1.3900	.9668	.8355	.9764	.8770	.9972	.8632	.9955	.8875
glass4	15.47	1.4690	.9766	.8156	.9828	.8739	.9992	.8652	.9973	.8767
glass04vs5	9.22	1.5420	.9940	.9941	.9940	.9550	.9990	.9554	.9925	.9771
page-blocks13vs4	15.85	1.5470	.9989	.9978	.9930	.9708	.9996	.9647	.9983	.9891
ecoli3	8.19	1.5790	.9220	.8230	.9714	.7999	.9902	.8491	.9857	.8903
yeast2vs4	9.08	1.5790	.9631	.8757	.9821	.8841	.9938	.8703	.9922	.9298
ecoli0346vs5	9.25	1.5950	.9206	.8166	.9899	.9182	.9995	.8659	.9957	.9123
ecoli046vs5	9.15	1.6030	.9214	.7542	.9793	.8915	.9981	.9156	.9959	.9278
ecoli0234vs5	9.10	1.6180	.9208	.7898	.9858	.9036	.9987	.9259	.9951	.9222
ecoli034vs5	9.00	1.6320	.9147	.7632	.9878	.9056	.9978	.8933	.9931	.9012
yeast02579vs368	9.14	1.6350	.8783	.8382	.9726	.9013	.9938	.8954	.9942	.9313
ecoli067vs5	10.00	1.6920	.9172	.8800	.9711	.9144	.9965	.8754	.9932	.9471
segment0	6.01	1.7980	.9926	.9831	.9978	.9916	.9999	.9920	.9998	.9951
ecoli2	5.46	1.8260	.9372	.8821	.9754	.8812	.9949	.8978	.9922	.9191
glass016vs5	19.44	1.8510	.9832	.9414	.9850	.9571	.9989	.9748	.9962	.9860
led7digit02456789vs1	10.97	1.9570	.9184	.8225	.9232	.8846	.9451	.8606	.9357	.8485
yeast6	39.15	1.9670	.8878	.7943	.9669	.7961	.9932	.7955	.9921	.8635
ecoli0137vs26	39.15	2.3020	.8360	.7800	.9692	.8299	.9986	.8104	.9968	.8316
glass6	6.38	2.3910	.9527	.9113	.9743	.9369	.9968	.9039	.9927	.9306
vowel0	10.10	2.4580	.9970	.9644	.9983	.9860	.9999	.9839	.9995	.9942
ecoli1	3.36	2.6500	.9387	.8650	.9423	.8870	.9765	.8812	.9739	.9158
yeast3	8.11	2.7510	.9403	.8943	.9341	.9145	.9744	.9029	.9697	.9485
ecoli4	13.84	3.2470	.9115	.8078	.9942	.9329	.9974	.8763	.9949	.9323
glass0123vs456	3.19	3.3240	.9787	.9037	.9699	.9554	.9980	.9390	.9926	.9609
wisconsin	1.86	3.5680	.9856	.9406	.9810	.9530	.9932	.9550	.9921	.9736
new-thyroid2	4.92	3.5790	.9879	.8919	.9902	.9540	.9993	.9554	.9971	.9867
new-thyroid1	5.14	3.5790	.9926	.9522	.9954	.9534	.9993	.9596	.9967	.9847
yeast5	32.78	4.1980	.9721	.8858	.9797	.9575	.9955	.9177	.9949	.9743
ecoli0vs1	1.86	9.7520	.9870	.9841	.9870	.9841	.9994	.9762	.9977	.9817
shuttle2vs4	20.50	12.1300	.9802	.9500	1.0000	1.0000	1.0000	.9960	.9998	.9986
shuttle0vs4	13.87	12.9700	1.0000	.9997	.9998	.9997	1.0000	.9997	1.0000	.9998
iris0	2.00	16.8200	1.0000	.9900	1.0000	.9900	1.0000	.9967	.9667	.9633
Average			.8925	.8052	.9546	.8474	.9844	.8407	.9803	.8803

Table A.2. Experimental results for C4.5, C4.5 with OVO and SMOTE+ENN (OVO+S_ENN), C4.5 with the global cost-sensitive learning (Global-CS), AdaBoost.NC, Random Forest, 1-FIS-MOEA and EFIS-MOEA in training and test with M-AUC metric.)

	C4.5		OVO+S_ENN		Global-CS		AdaBoost.NC		RandomForest		1-FIS-MOEA		EFIS-MOEA	
Data	Tr	Tst	Tr	Tst	Tr	Tst	Tr	Tst	Tr	Tst	Tr	Tst	Tr	Tst
aut	.9561	.8490	.9735	.8608	.9886	.9294	.9958	.8919	1.0000	.9006	.9910	.9122	.9742	.9174
bal	.8428	.7057	.9213	.7020	.9613	.6838	.9948	.7814	.8147	.7875	.9542	.7059	.9557	.7498
cle	.8458	.5702	.8868	.5508	.9923	.6027	.8252	.5328	1.0000	.6084	.9757	.5740	.9772	.5666
con	.8026	.6422	.7934	.6502	.8755	.6307	.8204	.6459	.8660	.6791	.8511	.6178	.8821	.6651
der	.9892	.9725	.9875	.9684	.9978	.9723	.9996	.9806	1.0000	.9831	.9948	.9780	.9923	.9808
eco	.8570	.7922	.9634	.8441	.9956	.7929	.9742	.8406	.9926	.8438	.9855	.7921	.9813	.8037
fla	.8045	.7729	.8397	.7894	.9023	.7840	.8719	.7593	.8944	.7639	.9028	.7813	.8975	.7861
gla	.9499	.8136	.9635	.8171	.9837	.8060	.9640	.7981	.9974	.8445	.9909	.8147	.9799	.8378
hay	.9404	.9249	.9386	.9209	.9421	.9191	.9368	.9132	.9079	.9000	.9520	.9252	.9449	.9276
led	.8810	.8536	.8771	.8438	.8825	.8463	.5739	.5880	.8946	.8361	.8871	.8415	.8831	.8506
lym	.9364	.7684	.9097	.8485	.9805	.8325	1.0000	.7395	1.0000	.7994	.9850	.7566	.9761	.8361
new	.9814	.9391	.9929	.9713	.9966	.9436	.9997	.9696	1.0000	.9563	.9981	.9461	.9931	.9591
nur	.8766	.9245	.9792	.9408	.9943	.9524	.9998	.9760	1.0000	.9517	.9944	.9540	.9936	.9539
pag	.9590	.9127	.9836	.9431	.9974	.9362	.9894	.9548	.9892	.9271	.9921	.9469	.9911	.9669
pos	.5071	.4917	.8359	.4885	.9436	.5632	.9755	.4728	.9653	.4883	.9585	.5472	.9148	.5462
sat	.9844	.9043	.9825	.9190	.9948	.9073	.9914	.9395	.9999	.9432	.9934	.9121	.9979	.9396
shu	.9835	.9592	.9985	.9907	.9999	.9927	.9997	.9913	.9993	.9843	.9996	.9924	.9972	.9640
spl	.9788	.9571	.9741	.9574	.9884	.9515	.9999	.9308	.9999	.9659	.9888	.9487	.9874	.9667
thy	.9992	.9834	.9821	.9529	.9994	.9931	.9999	.9921	1.0000	.9965	.9994	.9937	.9993	.9977
win	.9918	.9558	.9917	.9520	.9940	.9676	1.0000	.9812	1.0000	.9928	.9971	.9584	.9941	.9770
wqr	.8639	.6066	.9371	.6295	.9855	.6127	.9908	.6567	.9926	.6440	.9816	.6238	.9896	.6333
wqw	.8313	.6201	.8987	.6309	.9887	.6808	.9957	.7132	.9457	.6356	.9783	.6807	.9928	.6927
yea	.8694	.7560	.9048	.7790	.9580	.7402	.9740	.7747	.8951	.7290	.9643	.7410	.9752	.7862
zoo	.9826	.9013	.9691	.9506	1.0000	.9356	1.0000	.9356	1.0000	.9560	1.0000	.9722	.9885	.9534
Avg.	.9006	.8157	.9369	.8292	.9726	.8324	.9530	.8233	.9648	.8382	.9715	.8299	.9691	.8441