Neural Comput & Applic (2018) 30:289–305
https://doi.org/10.1007/s00521-016-2781-y

ORIGINAL ARTICLE

Sensitivity versus accuracy in ensemble models of Artificial Neural Networks from Multi-objective Evolutionary Algorithms

Juan Carlos Fernández[1] · Manuel Cruz-Ramírez[1] · César Hervás-Martínez[1]

Received: 30 June 2015 / Accepted: 14 December 2016 / Published online: 28 December 2016

Abstract This paper proposes a framework to obtain ensembles of classifiers from a Multi-objective Evolutionary Algorithm (MOEA), improving the restrictions imposed by two non-cooperative performance measures for multiclass problems: (1) the Correct Classification Rate or Accuracy (*CCR*) and, (2) the Minimum Sensitivity (*MS*) of all classes, i.e., the lowest percentage of examples correctly predicted as belonging to each class with respect to the total number of examples in the corresponding class. The proposed framework is based on collecting Pareto fronts of Artificial Neural Networks models for multiclass problems by the Memetic Pareto Evolutionary NSGA2 (MPENSGA2) algorithm, and it builds a new Pareto front (ensemble) from stored fronts. The ensemble built significantly improves the closeness to the optimum solutions and the diversity of the Pareto front. For verifying it, the performance of the new front obtained has been measured with the habitual use of weighting methodologies, such as Majority Voting, Simple Averaging and Winner Takes All. In addition to *CCR* and *MS* measures, three trade-off measures have been used to obtain the goodness of a Pareto front as a whole: Hyperarea, Laumanns's Hyperarea (*LAUMANNS*) and Zitzler's Spread (*M*3). The proposed framework can be adapted for any MOEA that aims to improve the compaction and diversity of its Pareto front, and whose fitness functions impose severe restrictions for multiclass problems.

Keywords Ensemble · Multi-objective Evolutionary Algorithm · Multiclass classification · Artificial Neural Networks · Minimum Sensitivity · Pareto Performance measures

1 Introduction

Pattern classification is the scientific discipline with the purpose of labeling patterns into a set of categories. Commonly the classification is based on individual statistical models (classifiers or learners) that are induced from an exemplary set of preclassified patterns [1, 2]. Nevertheless, it has also been theoretically and empirically verified that the combination of the results obtained by different classifiers may improve the results that each classifier provides [3, 4]. This combination is named in the literature as an ensemble [5, 6]. In contrast to ordinary learning, which attempts to construct one learner or classifier from training data, ensemble methods try to train and build a set of learners that are then combined, using in most cases a single base learning algorithm to produce homogeneous base learners.

On the other hand, sometimes it is necessary to optimize several non-cooperative objectives for solving classification problems [7, 8]. For this purpose, Multi-objective Evolutionary Algorithms (MOEAs) have arisen based on the Pareto dominance concept [9]. MOEAs provide a set of solutions for a final decider, all it equally valid, instead of only one solution [10]. Other potential advantage of the Pareto-based evolutionary learning approach is that using

✉ Juan Carlos Fernández
jfcaballero@uco.es

Manuel Cruz-Ramírez
mcruz@uco.es

César Hervás-Martínez
chervas@uco.es

[1] Department of Computer Science and Numerical Analysis, University of Cordoba, Rabanales Campus, Albert Einstein Building, 3rd Floor, 14071 Córdoba, Spain

multi-objective techniques based on the concept of dominance in a search space may help the learning algorithm to get out from local optima [11–13], thus improving the performance of the learning model. This potential advantage together with the iterative changes over the generations of a MOEA can make good solutions be obtained even if there are many local optima in the multidimensional search space [14]. Note that these advantages are potential, depending on how the problem is "objectivized," the characteristics of the algorithm used to explore the multi-objective problem space, and the type of problem itself.

Once the ideas of ensemble and MOEAs as a potential methodology for multiclass classification optimization with more than one objective have been briefly discussed, it is also necessary to comment that, in a general, most machine learning (ML) methodologies in the literature try to improve overall generalization capability of a classifier designed, but they do not tend to pay attention to the classification level per class. In the case of binary problems, the sensitivity and specificity can be optimized, but regarding multiclass classification, the dimensionality of the problem makes the optimization of the classification level per class much more complex and computationally costly. Sometimes improving overall generalization capability means sacrificing accuracy in one or more classes (see Fig. 1 as an illustrative example, discussed below), so that using only the *CCR* measure for training and evaluating classifiers is not enough for imbalanced problems [15, 16] and for problems in which all classes are equally important (misclassifications are equally costly). Additionally, *CCR* cannot capture all the different behavioral aspects found in two different classifiers [17, 18]. Thus, in this work, two non-cooperative measures or fitness functions with a MOEA are used for building ensembles [19, 20] for multiclass classification tasks: (1) *CCR* and, (2) *MS* as the rate of the worse classified class. This framework tries to improve the sensitivity of each of the classes for the problem considered, maximizing the minimum of all them (*MS*), while maintaining an acceptable overall generalization capability (*CCR*). This objective is not easy and it is necessary to emphasize the difficulty especially for problems with many classes or many unbalanced classes, where resampling is not as common as in the binary case [21].

Mono-objective algorithms have already shown promising results in convergence to the optimal front, but its high selection pressure, with a possible diversity loss, may lead the algorithm to prefer some specific areas of the Pareto front. Also note that in a mono-objective context, measures based on weighing other measures or those using an aggregation function of all the individual objectives, convert a multiclass problem in a binary one by the *One versus All* (OvO) and *One versus One* (OvA) methodologies [22, 23], but they do not address the problem as a whole. In this sense, a MOEA could provide better results using *MS* than a mono-objective algorithm if a good classification rate per class is needed.

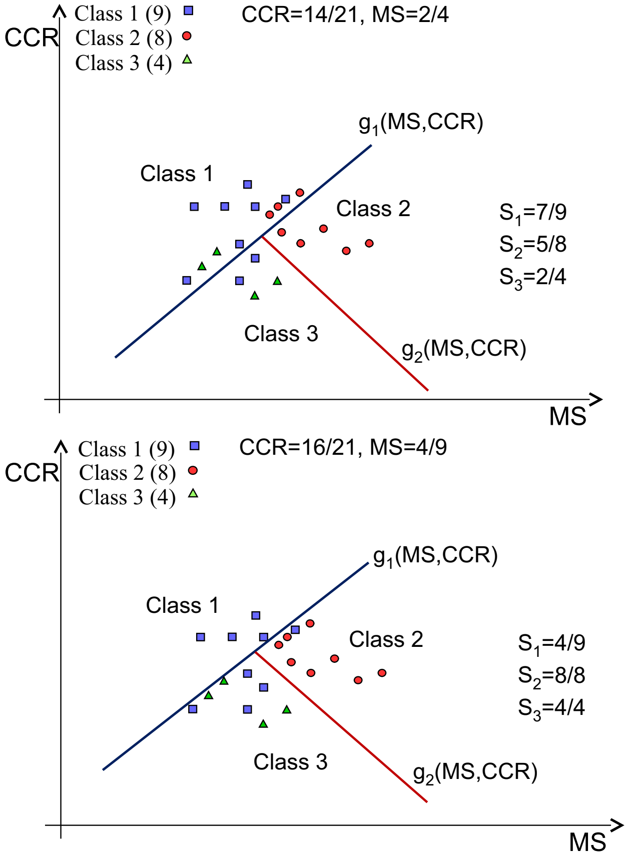

Fig. 1 Illustration of *CCR* and *MS* as conflicting objectives. $g_j(MS, CCR)$ denotes a classifier, and S_i denotes the sensitivity of the class i

Continuing with the *CCR* and *MS* measures for building multiclass classification models, these measures have a number of restrictive properties that are alleviated by building an ensemble of models. Both measures generate classifiers with a good classification rate per class, but, in a multi-objective context, they are very restrictive in the number of elements of the Pareto front generated by the MOEA, in the shape, in the closeness to the optimum solution, and in the diversity of the solution set, as shown in [24]. It is well-known [25]) that the diversity and the geometrical shapes of the Pareto fronts, among other characteristics of the multi-objective problem, can affect the performance of MOEAs. Therefore, one of the objectives of this work, besides obtaining good levels of classification accuracy for each of the classes, is to build an ensemble of multiclass classifiers using a framework to improve the performance of the Pareto fronts obtained with a MOEA, considering the *CCR* and *MS* measures.

The proposed framework builds a new and improved front of fronts (ensemble) using the MPENSGA2 algorithm [24], a MOEA that evolves a population of Artificial Neural Networks (ANNs) [26]. Then, a weighting methodology to obtain the prediction of the ensemble, such as Majority Voting (MV), Simple Averaging (SA) and Winner Takes All (WTA) [27], is applied. Also, three trade-off measures to obtain the quality of a Pareto front as a whole, Hyperarea (*HA*), Laumanns's Hyperarea (*LAUMANNS*) and Zitzler's Spread (*M*3), have been used. The new front minimizes the restrictions of the *CCR* and *MS* measures when they are used as fitness functions, obtaining a Pareto that is closer to the optimal solutions. Although the proposed framework has been used to improve the performance of the Pareto fronts obtained by the MPENSGA2 algorithm, it could be adapted to any MOEA based on the Pareto dominance concept, which uses non-cooperative fitness functions, especially those that are very restrictive in the search space.

In summary, the main contributions of the paper are the following:

- Based on a MOEA that evolves ANNs, the approach is able to produce a more diverse ensemble, increasing the number of models to solve multiclass problems. Some models are good at maximizing *CCR*, while others maximize *MS*, and some pay more attention to the cooperation of both measures. This extends the possibilities for the final decider.
- There is no need to weight the different objectives by optimizing the coefficient parameters. Being a framework based on a MOEA, an aggregation function considering the two single objectives is not needed. Also, the parameters of the individual networks can be effectively obtained in the MOEA.
- In general, the proposed framework obtains improved models with an acceptable overall generalization capability, while maximizing the sensitivity of each of the classes. Using conflicting metrics for this purpose is not very usual, which gives to this work an important novelty in the area.
- Due to the structure of the resulting Pareto front (ensemble), additional trade-off metrics to measure the quality of the new Pareto are employed to assess our proposal. The ensemble obtains better results in different trade-off metrics and better performance in *CCR* and *MS*, compared with a Pareto front methodology proposed in a previous work [24]. That is, the new framework improves the constraints imposed by both measures as fitness functions when a Pareto front is built.

Once explained the motivation of this paper the following sections are organized as follows: Sect. 2 is devoted to an analysis of some works for building of ensembles from MOEAs with ANNs and from other methodologies; Sect. 3 shows the MOEA used for obtaining ensemble models in multiclass problems, the fitness functions used to obtain classifier models, and the proposed framework for building ensembles. Section 4 describes the datasets used in the experimentation, the experimental setup, and the obtained results in the comparison procedure. Finally, Sect. 5 summarizes the conclusions and future improvements.

2 Artificial Neural Network ensembles from Muti-objective Evolutionary Algorithms

Generally when speaking of ensembles it is necessary to distinguish between the following: (1) how to generate classifiers, (2) how to choose the classifiers among those available to form the ensemble (linked to the concept of diversity), and (3) the rules or methods with which to use the ensemble to classify a given pattern or instance. Depending on what is taken into account, authors can create different ensemble taxonomies. For example, there are taxonomies that are based on how to combine the obtained classification by several base classifiers, and taxonomies based on what techniques are used for training classifiers that compose an ensemble [3, 27, 28]. Others are based on the starting point of the search space, on how the inducer or the training set are manipulated, or even on how the number of elements that form the ensemble is determined. The reader can find an extensive and updated state of the art with respect to references and taxonomies for building ensembles in [5, 29–31].

Another aspect in forming ensembles is the manner in which classifiers are trained, through a dependent or independent framework [27]. For a dependent framework the result of a classifier affects to the creation of the next, so there is interaction in the learning process. In this way, the previous knowledge can be used to guide future learning iterations. For an independent framework the original dataset is partitioned into several subsets (disjointed or overlapping) or even from the same dataset several inducers can be used, from which several classifiers are obtained. For problems in which each of the classifiers performs the same task and has comparable success, the classification techniques most widely used for measuring the performance of an ensemble are the weighting methods [27, 32], being the most popular MV, SA and WTA, which have been successfully proven in previous ensemble learning approaches [33–35]. In this paper an independent framework for building ANN ensembles from a MOEA is proposed, and these three trade-off techniques to classify patterns from the built ensemble are used.

2.1 Ensembles based on negative correlation learning

Neural network ensembles [36] are a learning paradigm where the ML models are collection of ANNs trained for the same task. Negative correlation learning (NCL) [37] is a successful ANN ensemble learning algorithm. It is different from previous works such as Bagging [38] or Boosting [39], since NCL emphasizes interaction and cooperation among individual learners in the ensemble by using an unsupervised penalty term in the error function to produce biased individuals whose errors tend to be negatively correlated. From here starts a new methodology associated with obtaining diversity in an ensemble by NCL, and the authors and other contributors made improvements over the years.

Thus, in [40], Liu and Yao describe an approach to designing ANN ensembles for both regression and classification problems with noise called the cooperative ensemble learning system (CELS). This approach can be regarded as one way of decomposing a large problem into smaller and specialized ones, so that each subproblem can be addressed by an individual ANN. A correlation penalty term in the error function was proposed to encourage the formation of specialists in the ensemble. CELS produces biased ANNs whose errors tend to be negatively correlated, showing very competitive results. A year later, the same authors showed in [41] an evolutionary ensemble with NCL (EENCL) to address the issues of automatic determination of the number of ANNs in an ensemble and the exploitation of the interaction between the design and combination of ANN. The idea of EENCL is to encourage different individual ANNs in the ensemble to learn different parts or aspects of the training data so that the ensemble can better learn the entire training data. The cooperation and specialization among different individual ANNs are considered during the individual ANN design. Experiments on two real-world problems demonstrated that EENCL could produce ANN ensembles with good generalization capability.

In [42], Chen and Yao propose an algorithm with Multi-objective Regularised Negative Correlation Learning (MRNCL) by formulating the RNCL algorithm [43] within an evolutionary framework and using a MOEA, which adds an additional regularization term to the fitness function. The additional regularization term to penalize large network weights to improve generalization are used to evolve a radial basis function (RBF) network ensemble. The new approach is shown to outperform a two-objective version using only accuracy and NCL, particularly on noisy problems.

2.2 Ensembles based on Differential Evolution

In addition to methodologies based on NCL, there are others in the literature, e.g., those using Differential Evolution (DE) concepts and other penalty terms. In [35, 44] two multi-objective formulations based on DE are proposed to evolve neuron ensembles by Abbaas et al. The first approach splits the training set into two subsets and uses the error on the subsets as the learning objectives, while the second proposal adds Gaussian noise to the training set as the second objective. The first formulation shows better results than the second one, and these methods are competitive compared to NCL for two (binary) benchmark classification tasks.

In [34], Chandra and Yao propose an algorithm called Diverse and Accurate Ensemble Learning Algorithm (DIVACE), which makes use of ideas found in NCL and Memetic Pareto algorithm for ANNs (MPANN) based on DE algorithm developed previously by Abbaas et al. [35]. DIVACE formulates the ensemble learning problem as an explicit multi-objective problem within an evolutionary setup aimed at finding a good trade-off between diversity and accuracy. Later, Chandra and Yao [45] improve DIVACE using a new diversity measure that they call Pairwise Failure Crediting (PFC) in place of the NCL penalty function term. PFC credits individuals in the ensemble with differences in the failure patterns, taking each pair of individuals and accruing credits in a manner similar to implicit fitness sharing.

In [33], Jin compares three Pareto-based multi-objective approaches to ensemble generation: DIVACE, MPANN and an algorithm with a hybrid binary and real-valued coding for optimizing the structure and weights of ANNs [46]. The three works evolve the structure of the ANN models for maintaining diversity, using as objectives the error and the complexity of the ANNs. The authors conclude that a deeper insight into the learning problem can be gained by analyzing the Pareto front composed of multiple Pareto-optimal solutions.

2.3 Ensembles based on co-evolution and cooperation

On the other hand arise works based on the idea of co-evolution and cooperation, e.g, in [47] a framework to generate ensembles of ANNs by cooperative co-evolution is proposed. This framework uses a MOEA where several subsets of ANNs are evolved. In [48], a methodology for creating ensembles with clustering based on co-evolution is proposed, using for it a multi-objective co-evolutionary strategy. This methodology is called CONE (Clustering and Co-evolution to Construct Neural Network

Ensembles). A clustering method is used to divide the input space of the training set into several subspaces without intersecting each other, so that they are used to train different species of ANNs with a MOEA. In the year 2003, Islam et al. [49] present a constructive algorithm for training cooperative ANN ensembles (CNNEs). This algorithm combines ensemble architecture design with cooperative training for ANNs in ensembles, obtaining diversity using NCL and different training epochs showing good results in an extensive number of benchmark problems in ML.

2.4 Ensembles based on other techniques

There are also jobs about ANN ensembles [19, 50–53] that are not based on the ideas of NCL, DE, Cooperation and Co-evolution, e.g., in [54], Dong et al. propose an ensemble neural network-based hybrid data-driven model for short-term load forecasting for high-efficiency electricity production, obtaining accurate predictions. The parameters and structures for the model are calibrated by using the NSGAII multi-objective optimization algorithm and the early stopping Levenberg–Marquardt algorithm.

Although this paper is focused on ensembles for ANNs and MOEAs, there are other ensemble methods studied in the literature and not based on it, some very popular as Bagging [38] and Boosting [39]. Both methodologies use a resampling technique to create different datasets for training different ML models in an ensemble. The training collection of each individual model is decided by the behavior in previous models. If the sample has been mistaken in previous models, then it will be present to the training collection of new models with higher probability. The new model can effectively deal with the samples which were difficult for the previous one. The main difference of Bagging and Boosting is that the training collection choose of Bagging is stochastic and that of Boosting is in sequence.

There are also evolutionary methodologies that not necessarily comprise ANNs or MOEAS and they can be used for classification task with ensembles. For example in [55] an ensemble particle swarm model selection (EPSMS) is used in the context of type/subtype of acute leukemia classification, achieving the best performance in this type of problem with respect to the methods used so far. This methodology is an improvement in the particle swarm procedure proposed in [56] for obtaining full models, where a PSMS searches for the best combination of methods for preprocessing, feature selection and classification from a predefined set of methods that are available in a ML toolbox. EPSMS automatically selects ensembles instead of single PSMS models, so it is more robust to noisy data and it provides more stable predictions. For this purpose EPMS uses an one-vs-all (OvA) method where a set of independent binary classifiers (PSMS partial solutions) is built given a multiclass classification problem. Each classifier is able to discriminate examples of one class of the problem and a pattern is assigned to the class corresponding to the classifier with the highest confidence in the correct labeling. In [57] an approach which builds heterogeneous ensembles based on genetic programming (GP) is also used for classification tasks, outperforming alternative ensemble methodologies. The objective of the methodology is to determine a fusion function that maximizes the classification performance on unseen data. The genetic algorithm uses mutation and crossover operators for combining multiple models outputs and returns a function which produces an output matrix with the actual prediction, taking into account the confidence of each model for each instance.

Wang et al. [58] make an experimental study for building ensembles with evolutionary algorithms (EAs) presenting three novel evolutionary approaches in supervised data mining scenarios. The first approach is based on encoding rule sets with bit string genomes which evolve via crossovers, while the second one utilizes Genetic Programming (GP) to create decision trees with arbitrary expressions attached to the nodes, both using solutions based on the Pareto concept for building an ensemble. The third approach uses GP but with an advanced fitness measure and some novel genetic operators so far. The performance comparison of the three methods over other existing in the literature shows that evolutionary methodologies are an important tool to consider for building ensembles of classifiers. In [59] an ensemble of classifiers for face detection is built, each of which is trained based on a particular weighting over the training examples by a genetic algorithm (GA). The difference from a traditional GA is that all the weightings (chromosomes) throughout evolution are exploited to form the final ensemble with a voting procedure, not just the best weighting. Comparative results with the AdaBoost and Bagging algorithms show that proposed method is very competitive in performance and computational cost. A complete general review about evolutionary ensemble methodologies in classification tasks can be consulted by the reader in [4, 27].

3 A methodology for building ensembles

This section shows the proposed framework for building an ensemble of classifiers from Pareto fronts provided by a MOEA, specifically the MPENSGA2 (Memetic Pareto Evolutionary NSGA2) algorithm [24]. Firstly, the non-cooperatives measures to optimize during the evolutionary process are presented, and then the procedure for building an ensemble is shown. Note that the main objective is to improve the structure of the Pareto fronts building a final ensemble, so that the number of models to solve multiclass problems may be increased, expanding the possibilities for the final decider and improving the performance and quality of the ensemble.

3.1 Non-cooperative objective functions

Two measures as fitness functions for obtaining classification models are used in this work: (1) the Correct Classification Rate or Accuracy (*CCR*) and, (2) the Minimum Sensitivity (*MS*), defined as the minimum of the sensitivities of all classes (for more details see [24]).

The *CCR* measure is given by the following expression:

$$CCR = \frac{1}{N}\sum_{i=1}^{Q} n_{ii}, \tag{1}$$

where Q is the number of classes, N is the number of patterns in the training or testing set, and n_{ii} is the number of patterns from the *i-th* class that are correctly classified in that class.

Let $S_i = n_{ii}/f_i$ be the number of patterns correctly predicted to be in class i with respect to the total number of patterns in class i, f_i. Then, the *MS* measure is given by the following expression:

$$MS = \min(S_i; i = 1...Q), \tag{2}$$

The *MS* measure is necessary to check the quality of a multiclass classifier. This is because most techniques improve the overall generalization capability of the classifier designed, but they do not maintain this capability in all classes. Therefore, the *MS* versus *CCR* pair, (*MS*, *CCR*), expresses two features associated with a classifier where each misclassification has the same importance in each class: the global performance and the rate of the worse classified class (the value of both measures must be maximized).

The selection of *MS* as a measure that is complementary to *CCR* can be justified by considering that *CCR* is the weighted average of the sensitivities of each class (the weights depend on the data set):

$$CCR = \sum_{i=1}^{Q} \frac{f_i}{N} S_i, \tag{3}$$

providing both intuitive and computational support of the sensitivities of each one of the Q classes. *CCR* is estimated based on the fixed weight f_i, and it is such that minimizes the variance of S_i. *MS* will improve *CCR* as a weighted average of the correct classification rates of each of the classes. This perspective involves two objectives that have not been used previously in a multi-objective context for multiclass classification, although they are equivalent subsets to those points on the $Q(Q-1)$ surface that trade-off the smallest total error (with respect to the $Q \times Q$ contingency or confusion matrix), sum of the $Q(Q-1)$ elements, against the worst misclassification rate for any class (i.e., the largest sums in the rows of the off-diagonal elements in the confusion matrix) [60]. Then in a Pareto sense, a 2-D line is traced out on the $Q(Q-1)$ surface.

The (*MS*, *CCR*) point of view allows to represent the classifiers in a two-dimensional space to visualize their performance, regardless of the number of classes in the problem. *MS* is represented in the abscissa and *CCR* in the ordinate (Fig. 2). In this way, the (*MS*, *CCR*) pair tries to find the point between the scalar *CCR* measure and the multidimensional ones based on misclassification rates.

Let *CCR* and *MS* be associated with the classifier g, then these measures verify that

$$MS \leq CCR \leq 1 - (1 - MS)p^*, \tag{4}$$

where p^* is the minimum of the estimated prior probabilities. Therefore, each classifier will be represented as a point in the white region in Fig. 2; hence, the area outside of the triangle is marked as unfeasible. It is important to note that the area inside the triangle may be feasible or not (attainable), depending upon the classifier and the difficulty of the problem. Also, it should be noted that high values of p^* make more difficult to obtain Pareto fronts near to the optimal, diverse and with an acceptable number of solutions, specially when the number of classes is high or the dataset is not balanced. Because of this restrictions, the number of individuals or classifiers in a Pareto front should usually be small, by the difficulty of approaching the (1, 1) point in (*MS*, *CCR*) space.

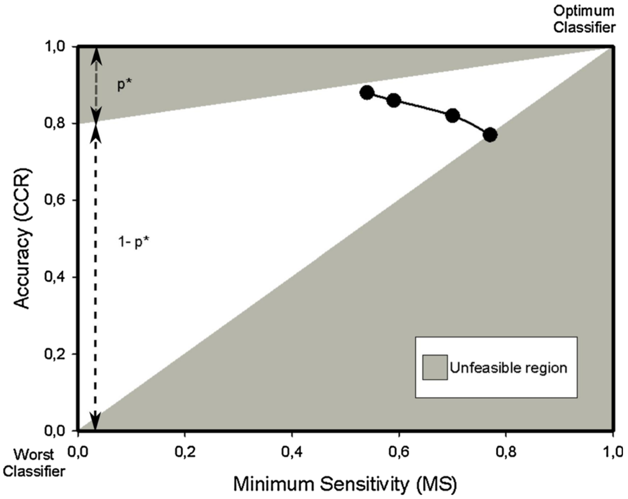

Fig. 2 Pareto front in the feasible region in the two-dimensional non-cooperative space (*MS*, *CCR*) for a multiclass classification problem

Additionally, a priori it could seem that *MS* and *CCR* objectives could be positively correlated; while this may be true for small values of *MS* and *CCR*, it is not so for values close to 1 in either *MS* or *CCR*, being highly competitive objectives (this fact justifies the use of a MOEA) at the top right corner of the white region, as it is demonstrated in more detail in [24]. First, observe that an increase in *CCR* does not imply an increase in *MS*. Reciprocally, an increase in *MS* does not mean an increase in *CCR*, since this means that the sensitivity of the worse classified class increases, but it may be at the expense of decrease in the sensitivities of other classes. Given the context of MOEAs, at the beginning of the evolutionary process, *CCR* and *MS* could be cooperative, but after a certain level, when their values are approaching the optimum value of 1, the objectives become competitive and an improvement in one objective tends to involve a decrease in the other one. Figure 1 shows an simple multiclass problem where in the first graph, a value of $CCR = \frac{14}{21}$ and $MS = \frac{2}{4}$ is obtained, being S_i the sensitivities of each class *i*. If the *CCR* value wants to be increased, the decision boundary should be moved to separate the class 2 and class 3 from the class 1 (second graph), but it would be necessary to reduce the sensitivity value in the class 1. Therefore, is needed to improve the Pareto fronts within the restrictive space (*MS*, *CCR*), because it is difficult to obtain Pareto fronts with diversity, with large numbers of individuals and close to the optimal point.

3.2 Proposed framework using the MPENSGA2 algorithm

The MPENSGA2 algorithm is a Memetic MOEA used for training a population of ANNs for multiclass problems, evolving architectures and connection weights simultaneously, and it is based on the NSGA2 algorithm proposed by Deb et al. [61]. The use of MOEAs along with local optimizers have been successfully considered in the literature [54, 62, 63], and, in this sense, MPENSGA2 includes a local search (LS) procedure called *iRprop*$^+$ (improved backpropagation) [64]. The *iRprop*$^+$ lifetime learning fine-tunes the weights, enhancing some individuals in the population. Additionally, in the evolutionary process, five mutations are used to obtain new individuals in the population: four structural mutations (for introducing diversity), add/delete neurons and add/delete links, and one parametric mutation based on the addition of Gaussian noise to the weights of the ANNs. The crossover operator is not considered due to its potential disadvantages in evolving ANNs [65]. Figure 3 shows the steps of MPENSGA2.

Regarding the ANN models used in the population, we consider a standard feedforward Multilayer Perceptron (MLP), with one input layer with *K* independent variables or features, one hidden layer with *M* sigmoidal hidden nodes and $Q-1$ outputs nodes in the output layer. We adopt the common technique of representing the class levels using a $1-of-Q$ encoding vector $y = (y^1, y^2, \ldots, y^Q)$, such as $y^l = 1$ if *x* corresponds to an example belonging to class *l*, and $y^l = 0$ otherwise. The output layer is interpreted from a probability point of view that considers the softmax activation function. Taking this consideration into account, it can be observed that the class predicted by the classifier corresponds to the neuron on the output layer whose a posteriori probability is greater. Therefore, one of the classes does not need to be estimated due to the properties of the probability function.

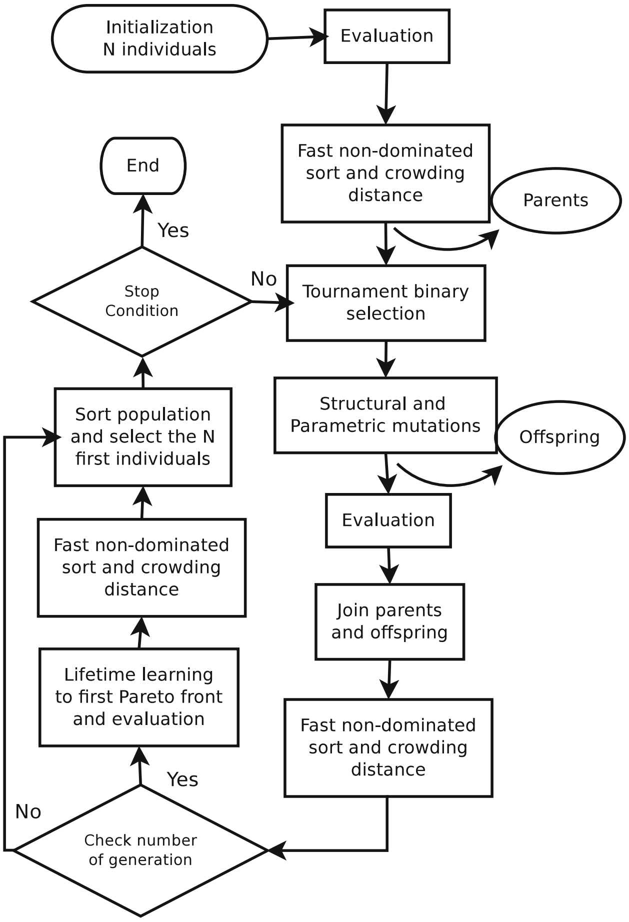
Initialization
N individuals
Evaluation
Fast non-dominated sort and crowding distance
Parents
End
Yes
Stop Condition
No
Tournament binary selection
Structural and Parametric mutations
Offspring
Sort population and select the N first individuals
Evaluation
Fast non-dominated sort and crowding distance
Join parents and offspring
Lifetime learning to first Pareto front and evaluation
Fast non-dominated sort and crowding distance
Yes
No
Check number of generation

Fig. 3 Steps in MPENSGA2

The *iRprop*$^+$ algorithm, it has been adapted to the softmax activation function, modifying the gradient function for the weights in the hidden and output layers. This LS procedures is used after the mutation phase, the fitness of each individual is updated with respect to the approximation error, and the modified weights are encoded back to the chromosome. *iRprop*$^+$ is applied only to the *CCR* measure to not adding computational cost to the algorithm, and it occurs in only three generations of the evolutionary process, at the beginning, middle and end (this reduces the computational cost, and a greater number of lifetime learning does not improve performance). For a complete description see [24].

Once the algorithm and its fitness functions have been defined, it is explained how to improve the performance of the Pareto fronts obtained and how to apply ensemble methods. Using weighting methodologies such as MV, SA and WTA, the habitual procedure for measuring the performance of a Pareto front is described in **Procedure 1** and it can be observed in Fig. 4. Additionally, Fig. 5 has been introduced here to show an example about how Pareto fronts can be obtained by this procedure for making statistical studies, applying it to a fold which is run N times for a dataset. This figure serves the reader as help to clarify the experimental section of this paper and shows the common way to obtain results from the MOEA.

Procedure 1 Ensemble methodologies applied to the Pareto Front provided by one MOEA.

```
STEP 1. Run the MOEA using the training set.
STEP 2.
while (last generation not reaches) do
    Train and evaluate the individuals (models for multidass
    classification) of the population by the non-cooperative
    fitness functions.
end while
STEP 3. Store the individuals of the Pareto front obtained
in the last generation.
STEP 4. Apply MV, SA or WTA to obtain with the training
and testing sets the performance of the Pareto front
($CCR$ and $MS$).
STEP 5. Apply Pareto measures to obtain the performance
of the trade-off surface.
```

Similarly, the procedure for obtaining an ensemble with the proposed framework is detailed in **Procedure 2** and it can be observed in Fig. 6. The evolutionary process is repeated for a number of runs, storing each Pareto front. At the end of the

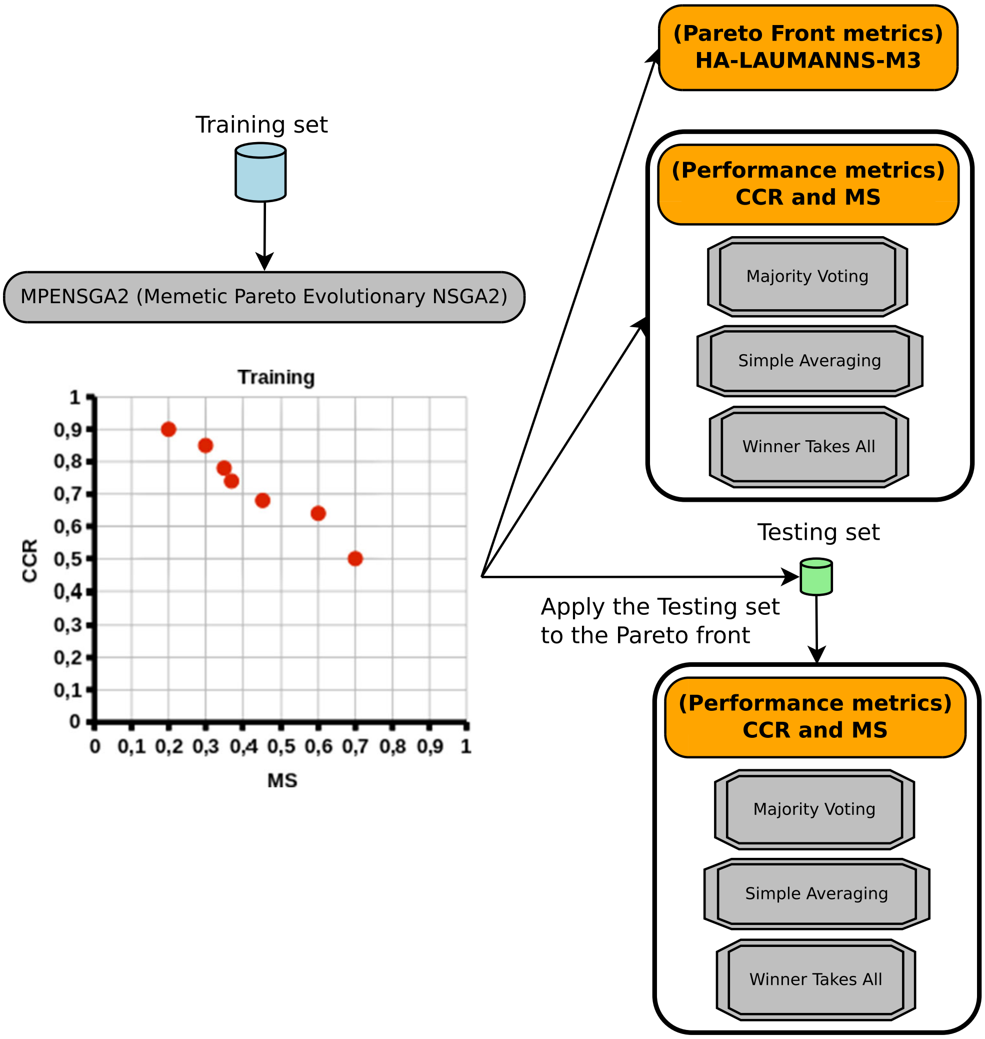

Fig. 4 **Procedure 1** for obtaining a Pareto front and how to apply weighting methodologies and trade-off measures

procedure, a new Pareto front (from all stored fronts) closer to the optimal solutions more diverse and with more performance is derived.

Procedure 2 Ensemble methodologies applied to the Pareto front provided by the proposed framework.

```
STEP 1. Run the MOEA using the training set with a
different seed in each run.
STEP 2.
while (last generation not reaches) do
    Train and evaluate the individuals of the population by
    the non-cooperative fitness functions.
end while
STEP 3. Store the individuals of the Pareto front obtained
in the last generation.
STEP 4.
if (all runs have not been run) then
    go to STEP 1.
end if
STEP 5. Recovery the Pareto fronts stored.
STEP 6. Build a new Pareto front (set of Pareto fronts)$_1$
checking the new Pareto dominance of the individuals.
STEP 7. Apply MV, SA or WTA to obtain with the training
and testing sets the performance of the new Pareto front
($CCR$ and $MS$).
STEP 8. Apply Pareto measures to obtain the performance
of the trade-off surface.
```

In the proposed framework, the precision of the ensemble is covered considering *CCR* and *MS* as objective functions to optimize, and the diversity of the individuals in the population is based on the following: (1) the NSGAII algorithm, which uses the concept of order and crowding distance, (2) the variation of the training set for each dataset (see Sect. 4.2) through *k*-folds (randomness of the method itself), and (3) using various initial random weights and varying the network architecture by the mutation operator of the evolutionary algorithm.

4 Experiments

4.1 Datasets

The proposed methodology is applied to 11 classification datasets taken from the UCI machine learning repository [66] and 3 interesting problems. Table 1 shows the features for each dataset, including number of patterns, number of input variables, number of classes, number of patterns per class and the p^* value. p^* is defined as the minimum of the estimated prior probabilities, value that has an important role in the relationship between the two measures.

A brief description of the three datasets outside the UCI machine learning repository is the following:

Agrarian This dataset corresponds to a complete socio-economic structure of 1620 agrarian enterprises in the south of Spain based on both the Gross Value Added (GVA) of the main productive activity and the size of the farm (very small, small, medium sized, big and very big). The dataset contains information about farmer characteristics, mechanization, the size of the farm and the costs and revenues of all productive activities. For more information see [67].

RichesRanking The Center for Global Development created the Commitment to Development Index (CDI) in 2003 [68], which ranks countries according to their contribution to the reduction in poverty in developing countries. The dataset contains ranked rich countries in terms of 22 OECD countries (after the incorporation of South Korea in 2008). The CDI assesses the commitment of these rich countries in terms of seven different policy areas. The countries are intended to classify according to the commitment to development: Highly committed countries, improving commitment countries, enough committed countries, not enough committed countries and not committed countries. For more information about this dataset, the reader can see [69].

Bankrupt This dataset corresponds to Financial Crisis study of 79 countries in the period 1981–1999 (annual data) for the detection and prediction of banking crises. The independent variables of the dataset are based on monetary policy strategies, being classified each case on crisis and non-crisis. For more details see [70].

4.2 Experimental setup

For the experimentation, three measures frequently used in the literature that consider the closeness and diversity of the trade-off surface of a Pareto front [71] have been selected in this work:

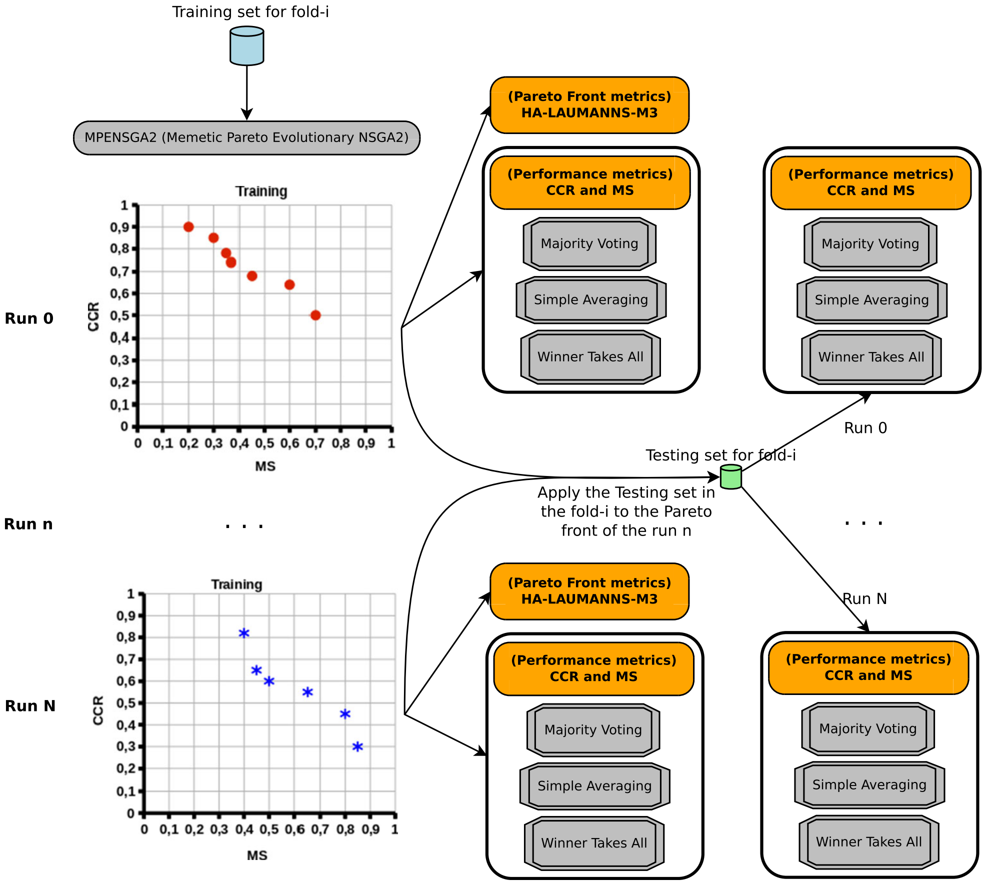

Fig. 5 **Procedure 1** for obtaining a Pareto front and how to apply weighting methodologies and trade-off measures in a fold run *N* times for a dataset

Hypervolume or Hyperarea (*HA*) The Hyperarea measure [72] calculates the volume or area in the objective space covered by the members of a Pareto front. *HA* is widely used in the literature as a closeness (convergence) and diversity measure.

Laumanns's Hyperarea (*LAUMANNS*) Another measure of closeness and diversity for checking the size of the dominated space of the objective functions is the Laummans's measure [73]. Laumanns's measure is an elitist measure based on the Lebesgue measurement, which also allows to measure the size of Pareto Front even if the set is composed of an infinite number of elements.

Zitzler's Spread (*M*3) *M*3 measures the spread of the trade-off surface. This measure was proposed by Zitzler in [9, 72]. It is a measure that also obtains the hypervolume that contains the trade-off surface using the sum of the greatest distance for each component *i* of the Pareto front. For two objectives, this measure refers to the Euclidean distance between the two extreme solutions in the objective space.

On the other hand, to analyze the performance of the Pareto fronts obtained from the proposed framework, three generic and well-known weighting methodologies [4, 32] from the literature have been utilized: MV, SA and WTA. In the comparison, a tenfold cross-validation was used for each dataset. The two procedures described are not deterministic, therefore, MPENSGA2 is run with several seeds

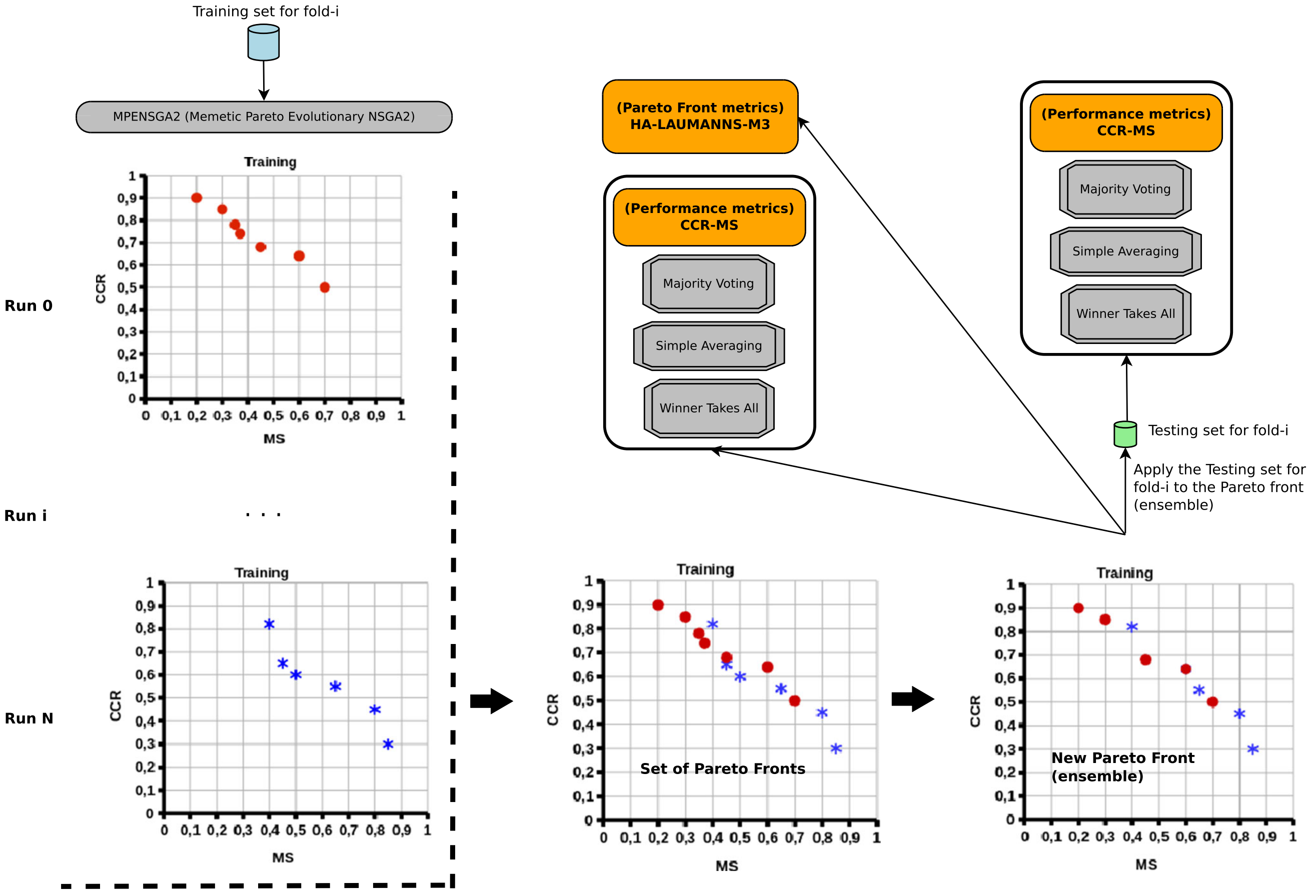
Training set for fold-i
MPENSGA2 (Memetic Pareto Evolutionary NSGA2)
Run 0
Training
CCR
MS
Run i
. . .
Run N
Set of Pareto Fronts
New Pareto Front (ensemble)
(Pareto Front metrics) HA-LAUMANNS-M3
(Performance metrics) CCR-MS
Majority Voting
Simple Averaging
Winner Takes All
Testing set for fold-i
Apply the Testing set for fold-i to the Pareto front (ensemble)
0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1

Fig. 6 **Procedure 2** for obtaining an ensemble from Pareto fronts and how to apply weighting methodologies and trade-off measures

for each fold. For **Procedure 1**, in order to take into account the randomness of the method, 30 runs were performed per fold, resulting in a total of 300 runs. Then, there are 300 Pareto fronts, so that the weighting methodologies are used for each front provided by each run of the $i-th$ fold. Figure 5 has been introduced for further clarification.

Regarding **Procedure 2**, 30 runs were also carried out for each fold, but, in this case, each fold provides an ensemble of 30 Pareto fronts. Hence, a new Pareto front is built from the set of the 30 Pareto fronts belonging to the concrete fold (see Fig. 6), and therefore, there are 10 Pareto fronts from the experimentation. The reader should keep in

Table 1 Characteristics of 11 UCI Benchmarks and 3 interesting problems

Dataset	#Patterns	#Inputs var.	#Classes	#Patterns per class	$p*$
Agrarian	1620	17	3	252-1047-321	0.155
RichesRanking	193	33	5	24-37-60-57-15	0.077
Autos	205	72	6	67-3-22-54-32-27	0.015
Balance	625	4	3	288-49-288	0.078
Bankrupt	521	22	2	357-164	0.315
German	1000	61	2	700-300	0.300
Heart	270	13	2	150-120	0.444
Horse	367	58	3	225-89-53	0.144
Liver	345	6	2	145-200	0.420
PageBlocks	5478	10	5	4914-330-29-89-116	0.005
Pima	768	8	2	500-268	0.349
Satimage	6435	36	6	1533-703-1358-626-707-1508	0.097
Segment	2310	19	7	330-330-330-330-330-330-330	0.142
Vehicle	846	18	4	204-218-217-207	0.241

mind these clarifications together with Figs. 6 and 5 to properly analyze the experimental results.

In all experiments, the population size for MPENSGA2 is established at $N_p = 100$. The mutation probability for each operator is equal to 1/5. The number of neurons that can be added or deleted has been established at a minimum of one neuron and a maximum of two (random value every time a mutation is used). With respect to the number of links, randomly are added or deleted 30% of the total number of links in the input-hidden layers, and 5% of the total number of links in the hidden-output layers. These values have been obtained experimentally by cross-validation over the training set. For the *iRprop* $^+$ algorithm, the number of epochs established for each LS is 10 (a greater number of epochs does not improve the results), $\eta^+ = 1.2$, $\eta^- = 0.5$, $\Delta_0 = 0.0125$ (the initial value of the Δ_{ij}), $\Delta_{\min} = 0$ and $\Delta_{\max} = 50$ (see [64] for *iRprop* $^+$ parameter descriptions).

4.3 Comparison procedure

A comparison of the new framework against the previous methodology presented in [24] and described in Fig. 1 is carried out. Table 2 compares **Procedure 1** (**P1**) versus **Procedure 2** (**P2**) in terms of *HA*, *LAUMANNS* and *M*3 trade-off measures applied to the Pareto fronts obtained by each procedure, showing the mean for each measure ($\overline{Metric}$). It is observed, and only from a descriptive point of view, that **P2** obtains the best mean result ($\overline{Metric}$) with respect to the use of **P1**.

Using the trade-off measures the intention is to prove whether the new proposed framework builds better Pareto fronts (as a whole) within the restrictive space (*MS*, *CCR*). It be must clarified that the comparison using trade-off measures can be only made between **P1** and **P2** without specifying the MV, SA or WTA weighting methodologies. This is because the three philosophies use the same Pareto front to make their predictions, each using a different calculation of the performance from the same front. That is, each weighting methodology has the same number of elements in the same position in the Euclidean plane with respect to the same dataset and for a given run. The main difference between the weighting methodologies when are used together with **P1** or **P2** is in how the Pareto front is obtained, which is built in **P2** from a set of fronts (ensemble). Mean and standard deviation values are extracted from the tenfold used in the experimentation in the way each procedure proposes and explained above.

Note that a Pareto front obtained in training in the (*MS*, *CCR*) space does not have to be a Pareto front by using the testing dataset on the same individuals. Therefore, trade-off measures are always used in training and then accuracy measures are used to obtain a performance value in testing. For this reason, in Figs. 4, 5 and 6, the trade-off measures are joined by an arrow from the Pareto front obtained in training.

To compare **P1** versus **P2** in terms of *HA*, *LAUMANNS* and *M*3 trade-off measures on the basis of both statistical and practical considerations, a signed-rank Wilcoxon's test is performed with the mean values obtained for each dataset (see Table 3), since there are only two methodologies with which to compare the mean rankings in 14 datasets,for each of the three trade-off measures for the Pareto fronts. The results show significant differences in mean for *HA*, p value $= 0.001$ (for a signification level $\alpha = 0.05$), while no significant differences for the measures *LAUMANNS* and *M*3, p value $= 0.149$ and p value $= 0.572$, respectively. This indicates that it is difficult in some datasets to improve the structure of the Pareto front obtained with **P1** methodology when in certain levels the front is approached the (1, 1) point in (*MS*, *CCR*) space and p^* has a high value.

On the other hand, Table 4 shows the MV, SA and WTA weighting methods comparing **P1** versus **P2** in terms of *CCR* and *MS* in the testing set (G); therefore, this table contains six methodologies: P1-MV, P1-SA, P1-WTA, P2-MV, P2-SA and P2-WTA. Mean Accuracy, Minimum Sensitivity ($\overline{CCR}(G)$, $\overline{MS}(G)$) and Mean Ranking ($\overline{R}(G)$) are also shown. From a descriptive point of view, first it should be noted the improvement obtained in the mean values for the testing sets for the *CCR* and *MS* measures with the new proposed procedure regarding the habitual one. The best Mean Accuracy ($\overline{CCR}(G)$) value is obtained by the WTA ensemble method with **P2**, followed by the SA ensemble method, also with **P2**.

For the mean Minimum Sensitivity ($\overline{MS}(G)$), the best result is obtained by the MV ensemble method with **the P2**, and the second best value is obtained by the SA ensemble method, also with the same procedure. It can be said that while using **P2** the best methodology for the *CCR* measure is the WTA method, and the best methodology for the *MS* measure is the MV method. The SA method is an intermediate point between the three weighting philosophies, and the use of **P1** always obtains worse results. If the mean rankings are observed, ($\overline{R}(G) = 1$ for the best performing method and $\overline{R}(G) = 6$ for the worst one), the descriptive conclusions are the same, that is, the best mean ranking in *CCR* is obtained by the WTA weighting methodology using **P2**, followed by the SA weighting methodology. In the case of *MS*, the best result is obtained by the MV weighting methodology, followed by the SA weighting methodology. The SA philosophy is again an intermediate point between the three methodologies, and the use of **P1** always obtains worse results. Therefore, it can be said that

Table 2 Comparison of **P1** versus **P2** in terms of *HA*, *LAUMANNS* and *M*3 trade-off measures

Dataset	HA		LAUMANNS		M3	
	(Zitzler's Hyperarea)		(Laumanns's Hyperarea)		(Zitzler's Spread)	
	Procedure 1 Mean ± SD	Procedure 2 Mean ± SD	Procedure 1 Mean ± SD	Procedure 2 Mean ± SD	Procedure 1 Mean ± SD	Procedure 2 Mean ± SD
Agrarian	0.343 ± 0.009	0.364 ± 0.006	0.555 ± 0.097	0.635 ± 0.036	0.152 ± 0.031	0.181 ± 0.038
RichesRanking	0.373 ± 0.036	0.432 ± 0.007	0.203 ± 0.220	0.243 ± 0.263	0.049 ± 0.046	0.023 ± 0.015
Autos	0.292 ± 0.032	0.337 ± 0.017	0.257 ± 0.219	0.307 ± 0.233	0.119 ± 0.069	0.181 ± 0.100
Balance	0.535 ± 0.029	0.575 ± 0.004	0.349 ± 0.248	0.309 ± 0.237	0.159 ± 0.104	0.121 ± 0.045
Bankrupt	0.549 ± 0.009	0.567 ± 0.005	0.494 ± 0.126	0.526 ± 0.158	0.077 ± 0.025	0.091 ± 0.029
German	0.536 ± 0.012	0.561 ± 0.003	0.503 ± 0.121	0.500 ± 0.181	0.082 ± 0.028	0.072 ± 0.019
Heart	0.602 ± 0.011	0.624 ± 0.008	0.218 ± 0.208	0.237 ± 0.197	0.028 ± 0.012	0.030 ± 0.012
Horse	0.420 ± 0.021	0.469 ± 0.014	0.401 ± 0.222	0.521 ± 0.226	0.085 ± 0.047	0.091 ± 0.027
Liver	0.507 ± 0.015	0.531 ± 0.008	0.375 ± 0.195	0.288 ± 0.263	0.049 ± 0.023	0.047 ± 0.020
Pageblk.	0.369 ± 0.035	0.428 ± 0.013	0.649 ± 0.170	0.751 ± 0.079	0.144 ± 0.063	0.202 ± 0.055
Pima	0.535 ± 0.007	0.549 ± 0.005	0.471 ± 0.145	0.487 ± 0.147	0.064 ± 0.025	0.059 ± 0.023
Satimage	0.345 ± 0.011	0.359 ± 0.002	0.730 ± 0.070	0.750 ± 0.052	0.089 ± 0.032	0.098 ± 0.029
Segment	0.362 ± 0.028	0.398 ± 0.006	0.501 ± 0.233	0.428 ± 0.219	0.030 ± 0.025	0.014 ± 0.007
Vehicle	0.380 ± 0.011	0.400 ± 0.004	0.654 ± 0.136	0.699 ± 0.089	0.040 ± 0.015	0.044 ± 0.012
$\overline{Metric}$	0.438	**0.47114**	0.45429	**0.47721**	0.08336	**0.08957**

Mean and standard deviation are extracted from the tenfold used in the experimentation. Mean for each measure, ($\overline{Metric}$), is also shown. The results are expressed between [0,1]. The best result is in boldface

Table 3 Comparison of **P1** versus **P2** in terms of *HA*, *LAUMANNS* and *M*3 measures using a signed-rank Wilcoxon's test with $\alpha = 0.05$

Z	P2HA-P1HA −3.297(a)	P2LAUMANNS-P1LAUMANNS −1.444(a)	P2M3-P1M3 −0.565(a)
Asymp. Sig. (2-tailed)	0.001	0.149	0.572

(a) Based on negative ranks

in the case of *MS* metric, it is reasonable that the *MV* methodology obtains the best results, as each model can have different results because of its greater variability, being reasonable to take decisions by majority. In the case of *CCR* metric, its values in each model are more homogeneous and then the *WTA* or *SA* methodologies are more appropriate.

To determine if there are statistical significant differences between the 6 weighting methodologies in terms of *CCR* and *MS*, and not only from a descriptive point of view, a procedure for comparing multiple classifiers over multiple datasets is employed [74] (the average ranking of each method in each dataset shown in Table 4 is used for it). The study begins with the non-parametric Friedman's test for the *CCR* and *MS* measures (now there are 6 methodologies and the distributions of the results are not normal), establishing the significance level at $\alpha = 0.05$ and rejecting the null-hypothesis by the test (all methods perform equally in mean ranking).

On the basis of this rejection a Holm's test [74] is conducted with respect to the six weighting methodologies. The Holm's test is a multiple comparison procedure that works with a control algorithm and compares it to the remaining methods, taking into account all datasets used in the experimentation for a concrete measure. The test statistics for comparing the *i*th and *j*th method using this procedure as follow:

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}} \tag{5}$$

where k is the number of algorithms, N is the number of data sets, and R_i is the mean ranking of the *i*th method. The z value is used to find the corresponding probability from the table of normal distribution, which is then compared

Table 4 MV, SA and WTA weighting methodologies comparing **P1** versus **P2** in terms of *CCR* and *MS* measures for the testing sets (*G*)

Dataset	P1-MV Mean ± SD	P1-SA Mean ± SD	P1-WTA Mean ± SD	P2-MV Mean ± SD	P2-SA Mean ± SD	P2-WTA Mean ± SD
CCR (Correctly Classified Rate)						
Agrarian	0.598 ± 0.044	0.611 ± 0.041	0.630 ± 0.039	0.625 ± 0.029	*0.633 ± 0.034*	**0.661±0.027**
RichesRanking	0.651 ± 0.141	0.655 ± 0.136	0.651 ± 0.137	*0.690 ± 0.062*	**0.697 ± 0.057**	0.652 ± 0.070
Autos	0.552 ± 0.112	0.545 ± 0.109	0.539 ± 0.112	*0.668 ± 0.063*	0.663 ± 0.052	**0.669 ± 0.047**
Balance	0.886 ± 0.042	0.890 ± 0.039	0.890 ± 0.040	*0.917 ± 0.032*	**0.920 ± 0.037**	0.915 ± 0.036
Bankrupt	0.737 ± 0.048	0.735 ± 0.046	0.732 ± 0.048	0.752 ± 0.041	*0.756 ± 0.042*	**0.764 ± 0.049**
German	0.705 ± 0.043	0.708 ± 0.043	0.709 ± 0.040	*0.718 ± 0.030*	0.717 ± 0.039	**0.743 ± 0.036**
Heart	0.819 ± 0.055	0.820 ± 0.052	0.821 ± 0.052	0.822 ± 0.060	**0.826 ± 0.055**	*0.822 ± 0.049*
Horse	0.604 ± 0.056	0.606 ± 0.054	0.604 ± 0.053	**0.641 ± 0.06**1	*0.636 ± 0.058*	0.634 ± 0.046
Liver	0.679 ± 0.083	0.681 ± 0.084	0.684 ± 0.083	0.689 ± 0.078	*0.693 ± 0.081*	**0.699 ± 0.089**
Pageblk.	0.932 ± 0.026	0.934 ± 0.019	*0.938 ± 0.013*	0.935 ± 0.025	0.938 ± 0.015	**0.942 ± 0.009**
Pima	0.755 ± 0.049	0.754 ± 0.047	0.758 ± 0.048	**0.760 ± 0.050**	0.755 ± 0.052	*0.760 ± 0.053*
Satimage	0.832 ± 0.037	0.834 ± 0.036	0.837 ± 0.036	0.854 ± 0.011	*0.857 ± 0.011*	**0.860 ± 0.012**
Segment	0.883 ± 0.046	0.885 ± 0.046	0.884 ± 0.050	0.933 ± 0.018	*0.937 ± 0.015*	**0.939 ± 0.014**
Vehicle	0.715 ± 0.055	0.718 ± 0.053	0.722 ± 0.052	0.761 ± 0.047	**0.765 ± 0.048**	*0.762 ± 0.046*
$\overline{CCR}(G)$	0.738	0.73993	0.74164	0.76921	*0.77107*	**0.77379**
$\overline{R}(G)$	5.53571	4.82143	4.35714	2.57143	*2.07143*	**1.64286**
MS (Minimum Sensitivity)						
Agrarian	*0.192 ± 0.122*	0.090 ± 0.121	0.041 ± 0.087	**0.288 ± 0.126**	0.113 ± 0.160	0.069 ± 0.099
RichesRanking	**0.352 ± 0.244**	0.351 ± 0.243	*0.352 ± 0.246*	0.350 ± 0.238	0.350 ± 0.238	0.290 ± 0.198
Autos	0.082 ± 0.163	0.079 ± 0.155	0.071 ± 0.144	**0.230 ± 0.226**	*0.193 ± 0.234*	0.174 ± 0.212
Balance	0.537 ± 0.276	0.527 ± 0.287	0.562 ± 0.269	**0.767 ± 0.094**	*0.741 ± 0.101*	0.665 ± 0.145
Bankrupt	0.594 ± 0.104	0.594 ± 0.102	0.599 ± 0.100	**0.616 ± 0.114**	*0.609 ± 0.110*	0.567 ± 0.106
German	0.559 ± 0.095	0.560 ± 0.093	0.557 ± 0.094	**0.602 ± 0.092**	*0.577 ± 0.088*	0.523 ± 0.085
Heart	0.740 ± 0.087	0.741 ± 0.082	*0.742 ± 0.081*	0.737 ± 0.079	**0.747 ± 0.080**	0.742 ± 0.082
Horse	0.226 ± 0.118	0.219 ± 0.117	0.212 ± 0.116	**0.266 ± 0.151**	*0.245 ± 0.146*	0.227 ± 0.124
Liver	0.585 ± 0.102	0.592 ± 0.103	*0.597 ± 0.102*	0.596 ± 0.100	0.595 ± 0.093	**0.613 ± 0.116**
Pageblk.	0.337 ± 0.179	0.311 ± 0.183	0.251 ± 0.180	**0.556 ± 0.094**	*0.538 ± 0.114*	0.459 ± 0.130
Pima	0.654 ± 0.105	0.654 ± 0.104	0.660 ± 0.106	**0.705 ± 0.091**	*0.671 ± 0.100*	0.664 ± 0.099
Satimage	0.629 ± 0.128	0.626 ± 0.131	0.600 ± 0.146	*0.692 ± 0.065*	**0.705 ± 0.056**	0.655 ± 0.067
Segment	0.691 ± 0.126	0.696 ± 0.128	0.692 ± 0.136	0.785 ± 0.069	*0.795 ± 0.056*	**0.811 ± 0.042**
Vehicle	0.493 ± 0.093	0.498 ± 0.094	0.494 ± 0.092	*0.564 ± 0.076*	**0.565 ± 0.063**	0.544 ± 0.074
$\overline{MS}(G)$	0.47307	0.46357	0.45586	**0.55721**	*0.53507*	0.50436
$\overline{R}(G)$	4.60714	4.71429	4.64286	**1.82143**	*1.96429*	3.25

Mean and standard deviation are extracted from the tenfold applied to each Procedure. Mean Accuracy, $\overline{CCR}(G)$, Minimum Sensitivity, $\overline{MS}(G)$, and Mean Ranking, $\overline{R}(G)$, are also shown. The results are expressed between [0,1]. The best result is in boldface and the second best result in italics

with an appropriate level of significance α. The Holm's test adjusts the value for α to compensate for multiple comparisons. This is done in a step-up procedure that sequentially tests the hypotheses ordered by their significance. We denote the ordered p values by $p1 \leq p2 \leq \cdots \leq p_{k-1}$. The Holm's test compares each p_i with $\alpha'_{Holm} = \alpha/(k-i)$, starting from the most significant p value. If $p1$ is below $\alpha/(k-1)$, the corresponding hypothesis is rejected and we allow it to compare $p2$ with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null-hypothesis cannot be rejected, all the remaining hypotheses are retained as well.

The Holm's test significance level is established to be $\alpha = 0.05$. As a control method each of the weighting methodologies that use **P2** (P2-MV, P2-SA and P2-WTA) are used for the *CCR* and *MS* measures. The results of the test are shown in Table 5. For the *CCR* measure, it is shown that all weighting methodologies that use **P2**

Table 5 MV, SA and WTA weighting methodologies comparing **P1** versus **P2**, using the Holm's test ($\alpha = 0.05$) in terms of *CCR* and *MS* measures

Algorithm	Control method	z	p	$\alpha'_{Holm0.05}$
CCR				
P1-MV	P2-MV	4.19212	0.00003	0.0100^{+}
P1-SA		3.18198	0.00146	0.0125^{+}
P1-WTA		2.52538	0.01156	0.0166^{+}
P2-SA		0.70711	0.47950	0.0500
P2-WTA		1.31320	0.18912	0.0250
P1-MV	P2-SA	4.89923	0.00000	0.0100^{+}
P1-SA		3.88909	0.00010	0.0125^{+}
P1-WTA		3.23248	0.00123	0.0166^{+}
P2-MV		0.70711	0.47950	0.0250
P2-WTA		0.60609	0.54446	0.0500
P1-MV	P2-TA	5.50532	0.00000	0.0100^{+}
P1-SA		4.49518	0.00001	0.0125^{+}
P1-WTA		3.83857	0.00012	0.0166^{+}
P2-MV		1.31320	0.18912	0.0250
P2-SA		0.60609	0.54446	0.0500
MS				
P1-MV	P2-MV	4.09112	0.00004	0.0100^{+}
P1-SA		3.99010	0.00007	0.0125^{+}
P1-WTA		3.93959	0.00008	0.0166^{+}
P2-SA		0.20203	0.83989	0.0500
P2-WTA		2.02030	0.04335	0.0250
P1-MV	P2-SA	3.88909	0.00010	0.0100^{+}
P1-SA		3.78807	0.00015	0.0125^{+}
P1-WTA		3.73755	0.00019	0.0166^{+}
P2-MV		0.20203	0.83989	0.0500
P2-WTA		1.81827	0.06902	0.0250
P1-MV	P2-WTA	1.91929	0.05495	0.0250
P1-SA		2.07082	0.03838	0.0100
P1-WTA		1.96980	0.04886	0.0166
P2-MV		2.02030	0.04335	0.0125
P2-SA		1.81827	0.06902	0.0500

$^{+}$ The difference is in favor of the control method

(regardless of the control method used) are significantly better on mean compared to the same weighting methodologies using **P1**. Also, it can be observed that the variance for **P2** is smaller for all methodologies compared with **P1**, indicating that the results, besides being better in mean, are more homogeneous.

For the *MS* measure, it can also be seen that the methodologies that use **P2** are significantly better on mean with respect to the methodologies that use **P1** in all methodologies, except in the WTA philosophy, that has no significant differences when it is used as a control algorithm. In contrast, for *MS* the variance of the results for **P2** are not smaller than for **P1**, and although there are significant differences in mean except for WTA, as it is expected, the results are not homogeneous due to the high variability of the results that can be obtained with the *MS* measure.

5 Conclusions

In this paper a framework to obtain an ensemble of classifiers (ANN models) from an MOEA is proposed, with the goal of improving the strong restrictions imposed by two measures associated with the training and performance of a classifier, the *CCR* and *MS*. Using these measures together with a MOEA produce restrictions, obtaining fronts with low diversity, low performance, and with a low number of individuals. The proposed framework is based on the collection of Pareto fronts obtained from several runs, building a new Pareto front (ensemble) that improves the closeness to the optimum solutions and the diversity of the set. For verifying this idea, the performance of the Pareto fronts obtained with habitual weighting methods, such as MV, SA and WTA have been compared with those obtained without using the ensemble methodology. The obtained results show that there are statistically significant differences between the proposed procedure and the usual procedure, and that those differences point favorably to the proposed method, with improvements in *CCR* and *MS*. In addition to these measures, the *HA*, *LAUMANNS* and *M*3 trade-off measures have been used to measure performance of the new Pareto fronts obtained, improving the area under the individuals within the Pareto front and, therefore, its performance. The proposed framework has been tested with 11 UCI datasets and 3 interesting problems. Currently, we are working to apply this methodology in future works addressing ordinal classification.

Acknowledgements This work has been partially subsidized by the TIN2014-54583-C2-1-R project of the Spanish Ministry of Economy and Competitiveness (MINECO), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain).

References

1. Duda RO, Hart PE, Stork DG (2000) Pattern classification, 2nd edn. Wiley, New York
2. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
3. Kuncheva LI (2004) Combining patterns classifiers, methods and algorithms. Wiley, London
4. Rokach L (2010) Pattern classification using Ensemble Methods. Series in machine perception artificial intelligence. World Scientific Publishing Company, Singapore
5. Zhou ZH (2012) Ensemble methods, foundations and algorithms. CRC Press, Boca Raton
6. Mohapatra S, Patra D, Satpathy S (2014) An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in

blood microscopic images. Neural Comput Appl 24(7–8):1887–1904

7. Vieira DAG, Vasconcelos JA, Caminhas WM (2007) Controlling the parallel layer perceptron complexity using a multiobjective learning algorithm. Neural Comput Appl 16(4–5):317–325
8. Ahmad F, Mat-Isa NA, Hussain Z, Sulaiman SN (2013) A genetic algorithm-based multi-objective optimization of an artificial neural network classifier for breast cancer diagnosis. Neural Comput Appl 23(5):1427–1435
9. Deb K (2004) Multi-objective optimization using evolutionary algorithms. Wiley-interscience series in systems and optimization. John Wiley & Sons, London
10. Loghmanian SMR, Jamaluddin H, Ahmad R, Yusof R, Khalid M (2012) Structure optimization of neural network for dynamic system modeling using multi-objective genetic algorithm. Neural Comput Appl 21(6):1281–1295
11. Albuquerque R, Pádua A, Takahashi RHC, Saldanha RR (2000) Improving generalization of MLPs with multi-objective optimization. Neurocomputing 35:189–194
12. Knowles J, Watson R, Corne D (2001) Reducing local optima in single-objective problems by multi-objectivization. In: Proceedings of the 1st international conference on evolutionary multi-criterion optimization, vol 1993. Springer, pp 269–283
13. Abbass HA (2003) Speeding up backpropagation using multiobjective evolutionary algorithms. Neural Comput 15:2705–2726
14. Tan KC, Khor EF, Lee TH (2006) Multiobjective evolutionary algorithms and applications. Advanced information and knowledge processing. Springer, London
15. Ghanem AS, Venkatesh S, West G (2010) Multi-class pattern classification in imbalanced data. In: Proceedings of the 20th international conference on pattern recognition (ICPR 2010), IEEE Press, pp 2881–2884
16. Wang S, Yao X (2012) Multiclass imbalance problems: analysis and potential solutions. IEEE Trans Syst Man Cybern B Cybern 42(4):1119–1130
17. Provost F, Fawcett T (1997) Analysis and visualization of the classifier performance: comparison under imprecise class and cost distribution. In: Proceedings of the third international conference on knowledge discovery (KDD97) and Data Mining, California, USA, August 1997, pp 43–48
18. Provost F, Fawcett T (1998) Robust classification system for imprecise environments. In: Proccedings of the fihteenth national conference on artificial intelligence, Chicago, USA, July 1998, pp 706–713
19. Yao X, Islam MM (2008) Evolving artificial neural network ensembles. IEEE Comput Intell Mag 3(1):31–42
20. Rahman MM, Islam MM, Murase K, Yao X (2016) Layered ensemble architecture for time series forecasting. IEEE Trans Cybern 46(1):270–283
21. Wang S, Minku LL, Yao X (2015) Resampling-based ensemble methods for online class imbalance learning. IEEE Trans Knownl Data Eng 27(5):1356–1368
22. Hsu CW, Lin CJ (2002) A comparison of methods for multi-class support vector machines. IEEE Trans Neural Netw 13(2):415–425
23. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45:427–437
24. Fernández JC, Martínez FJ, Hervás C, Gutiérrez PA (2010) Sensitivity versus accuracy in multi-class problems using memetic pareto evolutionary neural networks. IEEE Trans Neural Netw 21(5):750–770
25. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evolut Comput 10(5):477–506
26. Paliwal M, Kumar UA (2009) Neural networks and statistical techniques: a review of applications. Expert Syst Appl 36:2–17
27. Rokach L (2009) Taxonomy for characterizing ensemble methods in classification tasks: a review and annotated bibliography. Comput Stat Data Anal 53:4046–4072
28. Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Inf Fusion 6(1):5–20
29. Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach Learn 51:181–207
30. Zhenan H, Gary GY (2011) An ensemble method for performance metrics in multiobjective evolutionary algorithms . In: Proccedings of the 2011 IEEE congress on evolutionary computation (CEC), pp 1724–1729
31. Woźniak M, Graña M, Corchado E (2014) A survey of multiple classifier systems as hybrid systems. Inf Fusion 16:3–17
32. Theodoridis S, Koutroumbas K (2006) Pattern recognition, 3rd edn. Elsevier, Academic Press, San Diego, USA
33. Jin Y, Sendhoff B (2008) Pareto-based multiobjective machine learning: an overview and case studies. IEEE Trans Syst Man Cybern C Appl Rev 38(3):397–415
34. Chandra A, Yao X (2004) DIVACE: Diverse and accurate ensemble learning algorithm. In: Proceedings of the fifth international conference on intelligent data engineering and automated learning, vol 3177 of Lectures Notes and Computer Science. Springer, Berlin, pp 619–625
35. Abbass HA (2003) Pareto neuro-evolution: constructive ensemble of neural networks using multi-objective optimization. In: IEEE congress on evolutionary computation CEC2003, IEEE press, vol 3. Canberra, Australia, pp 2074–2080
36. Hansen L, Salamon P (1990) Neural network emsembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001
37. Liu Y, Yao X (1999) Ensemble learning via negative correlation. Neural Netw 12(10):1399–1404
38. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
39. Schapire RE (1999) A brief introduction to boosting. In: Proceedings of the 16th international joint conference on artificial intelligence, vol 2, pp 1401–1406
40. Liu Y, Yao X (1999) Simultaneous training of negatively correlated neural networks in an ensemble. IEEE Trans Syst Man Cybern B Cybern 29(6):716–725
41. Liu Y, Yao X (2000) Evolutionary ensembles with negative correlation learning. Trans Evolut Comput 4(4):380–387
42. Chen H, Yao X (2009) Multi-objective neural network ensembles based on regularized negative correlation learning. IEEE Trans Knowl Data Eng 22(12):1738–1751
43. Chen H, Yao X (2009) Regularized negative correlation learning for neural network ensembles. IEEE Trans Neural Netw 20(12):1962–1979
44. Abbass HA (2006) Pareto-optimal approaches to neuro-ensemble learning. In: Proceeding on studies in computational intelligence (SCI), vol 16. Springer, pp 407–427
45. Chandra A, Yao X (2006) Ensemble learning using multi-objective evolutionary algorithms. J Math Model Algorithms 5(4):417–445
46. Jin Y, Okabe T, Sendhoff B (2004) Evolutionary multi-objective approach to constructing neural network ensembles for regression, volume 1 of advances in natural computation. World Scientific, pp 653–672
47. Garcia-Pedrajas N, Hervas-Martinez C, Ortiz-Boyer D (2005) Cooperative coevolution of artificial neural network ensembles for pattern classification. IEEE Trans Evolut Comput 9(3):271–302
48. Minku FL, Ludemir TB (2008) Clustering and co-evolution to construct neural network ensembles: an experimental study. Neural Netw 21:1363–1379
49. Islam MM, Yao X (2003) A constructive algorithm for training cooperative neural networks ensembles. IEEE Trans Neural Netw 14(4):820–834

50. Jin Y, Okabe T, Sendhoff B (2004) Neural network regularization and ensembling using multi-objective evolutinary algorithms. In: Proceedings of the congress on evolutionary and ensembling using multi-objective evolutionary algorithms, vol 1. Portland, pp 1–8
51. Fieldsend JE, Singh S (2005) Pareto evolutionary neural networks. IEEE Trans Neural Netw 16(2):338–354
52. Jin Y, Sendholf B, Körner E (2006) Simultaneous generation of accurate and interpretable neural network classifiers. Stud Comput Intell 16:291–312
53. Yen GG (2006) Multiobjective evolutionary algorithm for radial basis function neural network design. Multi-objective machine learning. Studies in computational intelligence, vol 162. Springer, Berlin, Heidelberg, pp 221–239
54. Dong JR, Zheng CY, Kan GY, Zhao M, Wen J, Yu J (2015) Applying the ensemble artificial neural network-based hybrid data-driven model to daily total load forecasting. Neural Comput Appl 26(3):603–611
55. Escalante HJ, Montes y Gómez M, González JA, Gómez-Gil P, Altamirano L, Reta CA, Rosales A (2012) Acute leukemia classification by ensemble particle swarm model selection. Artif Intell Med 55:163–175
56. Escalante HJ, Montes M, Sucar E (2010) Ensemble particle swarm model selection. In: Proceedings of the international joint conference on neural networks (IJCNN 2010), pp 1814–1824, July 2010
57. Acosta-Mendoza N, Morales-Reyes A, Escalante HJ, Gago-Alonso A (2014) Learning to assemble classifiers via genetic programming. Int J Pattern Recognit Artif Intell 28(7):1460005
58. Wang P, Weise T, Chiong R (2011) Novel evolutionary algorithms for supervised classification problems: an experimental study. Evolut Intell 4:3–16
59. Wang X, Wang H (2006) Classification by evolutionary ensembles. Pattern Recognit 39(4):595–607
60. Everson RM, Fieldsend JE (2006) Multi-class ROC analysis from a multi-objetive optimisation perspective. Pattern Recognit Lett 27:918–927
61. Deb K, Pratab A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA2. IEEE Trans Evolut Comput 6(2):182–197
62. Chen MR, Lu YZ, Yang G (2008) Multiobjective optimization using population-based extremal optimization. Neural Comput Appl 17(2):101–109
63. Basseur M, Zeng RQ, Hao JK (2012) Hypervolume-based multiobjective local search. Neural Comput Appl 21(8):1917–1929
64. Igel C, Hüsken M (2003) Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing 50(6):105–123
65. Angeline PJ, Sauders GM, Pollack JB (1994) An evolutionary algorithm that constructs recurren neural networks. IEEE Trans Neural Netw 5:54–65
66. Lichman M (2013) UCI Machine learning repository. University of california, school of information and computer science, CA. Available online at http://archive.ics.uci.edu/ml.Irvine
67. Fernandez-Navarro F, Hervás-Martínez C, García-Alonso C, Torres-Jiménez M (2011) Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity. Expert Syst Appl 38(10):12483–12490
68. McGillivray M (2003) Commitment to development index: a critical appraisal. Technical report, AusAid
69. Sianes A, Dorado-Moreno M, Hervás-Martínez C (2014) Rating the rich: an ordinal classification to determine which rich countries are helping poorer ones the most. Soc Indic Res 116(1):47–65
70. Gutiérrez PA, Segovia-Vargas MJ, Salcedo-Sanz S, Hervás-Martínez C, Sanchis A, Portilla-Figueras JA, Fernandez-Navarro F (2010) Hybridizing logistic regression with product unit and RBF networks for accurate detection and prediction of banking crises. Omega 38(5):333–344
71. Yen GG, Zhenan H (2014) Performance metrics ensemble for multiobjective evolutionary algorithms. IEEE Trans Evolut Comput 18(1):131–144
72. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strenght Pareto approach. IEEE Trans Neural Netw 3(4):414–417
73. Laumanns M, Zitzler E, Thiele L (2000) A unified model for multiobjective evolutionary algorithms with Elitism. In: 2000 congress on evolutionary computation, vol 1, pp 46–53
74. Demsar J (2006) Statistical comparisons of clasiffiers over multiple data sets. J Mach Learn Res 7:1–30