On the Use of Fuzzy ART and SOM Networks in Ensemble Classifiers: A Performance Comparison

César L. C. Mattos[1] **, Guilherme A. Barreto**[1]

[1]Federal University of Ceará, Department of Teleinformatics Engineering
Av. Mister Hull, S/N, Campus of Pici, Center of Technology
CP 6005 – CEP 60455-970 – Fortaleza – Ceará – Brazil

cesar@lesc.ufc.br, guilherme@deti.ufc.br

Abstract. *In this paper we introduce two novel ensemble models that are built using Fuzzy ART (FA) and SOM networks as base classifiers. For this purpose, we first describe three different strategies to convert these unsupervised competitive learning algorithms to supervised ones to allow them to be applied to pattern classification tasks. Then, a metaheuristic solution based on a hybrid PSO algorithm is devised for parameter optimization of the proposed ensemble classifiers. A comprehensive performance comparison using 10 benchmarking data sets indicates that the FA- and SOM-based ensemble classifiers consistently outperform ensembles built from standard supervised neural networks, such as the Fuzzy ARTMAP and the Extreme Learning Machine.*

1. Introduction

Neural classifiers are typically built from supervised artificial neural networks (ANNs), such as the MLP and RBF Networks [Haykin 2008], ARTMAP networks [Carpenter et al. 1991a] and, more recently, the Extreme Learning Machine (ELM) [Huang et al. 2006].

In contrast to learning stand-alone classifiers, Ensemble Learning (EL) has been studied as an alternative to improve the generalization power of single classifiers [Dietterich 2000]. The ensemble's output is obtained by combining the outputs of a set of single classifiers. The result is an ensemble learning method which can reduce both the bias and the variance of single learning algorithms and guarantee an error less than or equal to the average quadratic error of the individual classifiers [Krogh and Vedelsby 1995].

Unsupervised competitive learning ANNs, such as the SOM [Kohonen 1982], the ART2 [Carpenter and Grossberg 1987a] and Fuzzy ART [Carpenter et al. 1991b] networks, for not requiring labeled data for training, are commonly applied to data clustering, vector quantization for signal/image compression, and dimensionality reduction tasks. Available algorithms in both SOM and ART families are considered to be unstable algorithms and *weak* learners (indeed, these are typical features of ANNs in general), which are two base preconditions necessary for building efficient ensembles [Hansen and Salamon 2002]. However, the development of SOM- and ART-based ensemble models is still in its first infancy.

SOM-based classification ensembles frequently proceed with the following steps [Corchado et al. 2007, Petrikieva and Fyfe 2002]: after training several SOMs in the

usual unsupervised way, the training patterns are presented once more to the individual maps in order to tag each neuron with the more frequent class label among those mapped to that neuron. Then, each individual SOM in the ensemble outputs the class label associated with the corresponding best-matching unit for a given input pattern. Finally, a majority voting rule is adopted in order to choose the final decision of the ensemble.

ART-based ensemble models are less common. This can be explained partly due to the high number of parameters, such as the vigilance and the choice parameters, that need to be set up in advance for this type of ANN architecture. Anyway, the works found in the literature survey carried out for this research used ARTMAP networks as base classifiers [Santos and Canuto 2008, Loo et al. 2006]. It was not found previous works using ART networks. It is worth remembering that learning in ART networks is unsupervised, while for ARTMAP networks learning is supervised.

From the exposed so far, the main contributions of the present paper is threefold. The first one involves a comprehensive evaluation of the influence of three different supervised variants of the SOM network in the performances of SOM-based ensemble classifiers. This approach gives rise to the MUSCLE (Multiple SOM Classifiers in Ensembles) models. The second contribution consists in developing new techniques, inspired by the supervised variants of the SOM, for building ART-based ensemble models. This approach gives rise to the ARTIE (ART networks in Ensembles) models. The third contribution is related with the proposal of a hybrid Particle Swarm Optimization (PSO) technique for tuning the parameters of the ARTIE and MUSCLE models.

The proposed models are compared with ensembles of two popular supervised algorithms (Fuzzy ARTMAP and ELM) using 10 benchmarking data sets. The obtained results strongly suggest that the performances of ARTIE and MUSCLE architectures are comparable to those achieved by ensembles of standard supervised neural networks.

The remaining part of this paper is organized as follows. In Section 2 we present a brief overview on the basic operations of the Fuzzy ART and SOM networks. In Section 3 we introduce the ARTIE and MUSCLE models, as well as the I-HPSO algorithm. In Section 4 we present and discuss the obtained results for 10 benchmarking classification datasets. The paper is concluded in Section 5.

2. Basics of Fuzzy ART and SOM Networks

This section summarizes the operations of Fuzzy ART and SOM networks as unsupervised learning algorithms. Both networks are comprised of one layer of neurons and their corresponding weight (prototype) vectors and the training process follows the principles of competitive learning. According to this learning paradigm, neurons "compete" in order to find groups of similar unlabeled input patterns (clustering) or, equivalently, to achieve a compact representation of the input patterns (vector quantization).

2.1. The Fuzzy ART Network (FA)

The main idea behind any ART architecture is that, if an input pattern is different enough from the patterns already stored in the long-term memory (i.e. weights) of the network, then create a new category (cluster) and associate the "different input pattern" to it [Keskin and Özkan 2009]. This novelty detection mechanism is highly effective in identifying abnormal (outliers) patterns in the data [Barreto and Aguayo 2009].

The FA network extends ART1, which was originally designed to process only binary data [Carpenter and Grossberg 1987b], being able to also learn analog inputs. The initial network contains only one neuron with its weights all set to 1. Let $\mathbf{a}(k) = [a_1(k) \quad a_2(k) \quad \cdots \quad a_p(k)]^T$, a p-dimensional input pattern at the k-th learning iteration. Then, the FA network algorithm follows the steps below.

Step 1 Complement code the p-dimensional input pattern $\mathbf{a}(k)$ into a $2p$-dimensional vector $\mathbf{x}(k) \in \mathbb{R}^{2p}$: $\mathbf{x}(k) = [\mathbf{a}(k) \quad \mathbf{a}^c(k)]^T$, where $a_i^c(k) = 1 - a_i(k), \quad \forall i \in \{1, 2, \ldots, p\}$.

Step 2 Present $\mathbf{x}(k)$ to the first layer of the network, L_1, and compute the activations $T_j(k), \forall j \in \{1, \ldots, n\}$, where n is the number of neurons of the network. For each neuron j, associated with a weight vector $\mathbf{w}_j(k) \in \mathbb{R}^{2p}$, compute the corresponding activation $T_j(k)$ as follows:

$$T_j(k) = \frac{|\mathbf{x}(k) \wedge \mathbf{w}_j(k)|}{\beta + |\mathbf{w}_j(k)|}, \quad \forall j \in \{1, 2, \ldots, n\}, \tag{1}$$

where $|\mathbf{x}(k)| = \sum_{i=1}^{2p} |x_i(k)|$ is the sum of the absolute value of the components of $\mathbf{x}(k)$, the symbol $\wedge$ denotes the fuzzy minimum operator and β is a small positive real number known as the choice parameter.

Step 3 Find the index of the winning neuron j^*, i.e. the one with maximum activation:

$$j^*(k) = \arg\max_{\forall j} \{T_j(k)\}. \tag{2}$$

Step 4 Check if the winning neuron j^* satisfies the vigilance criterion, i.e. if $|\mathbf{x}(k) \wedge \mathbf{w}_{j^*}(k)| \,/\, |\mathbf{x}(k)| \geq \rho$, where $0 < \rho < 1$ is the vigilance parameter. If the vigilance test is satisfied, go to Step 5; otherwise, reset the activation of neuron j^* to zero (i.e. $T_{j^*}(k) = 0$) and go to Step 3. The search is repeated until a neuron passes the vigilance test or all neurons have been tested. If all neurons fail to pass the vigilance test, create a new category using the current input pattern as its prototype vector and go to Step 1.

Step 5 Update the weight vector of the winning neuron:

$$\mathbf{w}_{j^*}(k+1) = \eta \left(\mathbf{w}_{j^*}(k) \wedge \mathbf{x}(k)\right) + (1 - \eta)\mathbf{w}_{j^*}(k), \tag{3}$$

where $0 < \eta \leq 1$ is the learning rate. Go to Step 1.

Steps 1 to 5 are repeated for each training pattern. In this paper just one training epoch (one cycle through the training set) is used, simulating an on-line operation. It is important to note that after the training phase each neuron defines a cluster by a hyperbox [Carpenter et al. 1991b]. The testing process is done through a simple winner-take-all competition based on Eq. (2).

2.2. The Self-Organizing Map (SOM)

Introduced by [Kohonen 1982], the SOM learns from examples a mapping (projection) from a high-dimensional continuous input space $\mathcal{X}$ onto a low-dimensional discrete space (lattice) $\mathcal{Z}$ of n neurons which are arranged in fixed topological forms, e.g. as a rectangular 2-dimensional array. The map $j^*(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Z}$, defined by the weight matrix

$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q), \mathbf{w}_j \in \mathbb{R}^p \subset \mathcal{X}$, assigns to each input vector $\mathbf{a}(k) \in \mathbb{R}^p \subset \mathcal{X}$ a winning neuron $j^*(k) \in \mathcal{Z}$, determined by

$$j^*(k) = \arg\min_{\forall j} \|\mathbf{a}(k) - \mathbf{w}_j(k)\|, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean distance and k is the current iteration of the algorithm.

The weight vector of the current winning neuron and the ones of its neighboring neurons are simultaneously adjusted according to the following learning rule:

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \alpha(k)h(j^*, j; k)[\mathbf{a}(k) - \mathbf{w}_j(k)] \tag{5}$$

where $0 < \alpha(k) < 1$ is the learning rate and $h(j^*, j; k)$ is a weighting function which limits the neighborhood of the winning neuron. A usual choice for $h(j^*, j; k)$ is given by the Gaussian function:

$$h(j^*, j; k) = \exp\left(-\frac{\|\mathbf{r}_j - \mathbf{r}_{j^*}\|^2}{2\sigma^2(k)}\right), \tag{6}$$

where $\mathbf{r}_j$ and $\mathbf{r}_{j^*}$ are respectively, the coordinates of the neurons j and j^* in the output array, and $\sigma(k) > 0$ defines the radius of the neighborhood function at iteration k. The variables $\alpha(k)$ and $\sigma(k)$ should both decay with time to guarantee stable convergence of the weight vectors. In this paper, we adopt an exponential decay for both variables:

$$\alpha(k) = \alpha_0 \left(\frac{\alpha_T}{\alpha_0}\right)^{(k/T)} \quad \text{and} \quad \sigma(k) = \sigma_0 \left(\frac{\sigma_T}{\sigma_0}\right)^{(k/T)} \tag{7}$$

where α_0 (σ_0) and α_T (σ_T) are the initial and final values of $\alpha(k)$ ($\sigma(k)$), respectively.

Weight adjustment is performed until a steady state of global ordering of the weight vectors has been achieved. In this case, we say that the map has converged. The resulting map also preserves the topology of the input samples in the sense that adjacent patterns are mapped into adjacent regions on the map. Due to this topology-preserving property, the SOM is able to cluster input information and spatial relationships of the data on the map. In this paper, the SOM is trained for one epoch only. The testing phase is similar to FA networks, with a winner-take-all competition based on Eq. (4).

3. The Proposed Approaches

The MUSCLE model extends previous SOM-based EL models [Corchado et al. 2007, Petrikieva and Fyfe 2002] by developing two other ways of designing SOM-based EL classifiers. Previous ART-based EL models [Santos and Canuto 2008, Loo et al. 2006] are built using ARTMAP networks, which are supervised classifiers. The ARTIE model, inspired by the strategies used by the MUSCLE models, is a novel approach for building ART-based EL classifiers from the FA networks. Finally, a novel hybrid PSO method is proposed for parameter optimization of the base classifiers used by the MUSCLE and ARTIE models. More details are given in the following sections.

3.1. Adapting FA and SOM for Classification Problems

In order to use FA and SOM networks for supervised classification, some modifications are necessary in their original learning algorithms. The present paper will focus on three strategies for designing FA- and SOM-based classifiers. These strategies, identified by a suffix Ci, i $\in \{1, 2, 3\}$, have been used to design SOM-based classifiers. However, to the best of our knowledge, it is the first time they are used to design FA-based classifiers.

3.1.1. Strategy C1: Post-Training Neuron Labeling

In this approach the SOM and FA networks are trained in the usual unsupervised way. Afterwards, a neuron labeling process is carried out by presenting the training dataset once again to the corresponding network and determining the winning neuron for each pattern vector, according to Eq. (4) for SOM networks or Eq. (2) for FA networks. The label of the winning neuron is assigned on a majority voting basis. Ties can be broken by random selection of the competing labels.

During the testing phase, the winning neuron for an unknown pattern is searched through Eq. (4) for SOM networks, or through Eq. (2) for FA networks. The predicted class for the current input pattern is then the class associated with the winning neuron.

Regarding the SOM, this strategy has been used for the design of single classifiers [Wyns et al. 2004, Christodoulou et al. 2003]; however, it seems that it was never tested before with the FA network.

3.1.2. Strategy C2: One Network per Class

The second strategy uses one SOM (or FA) network for each available class; for instance, if three classes of data are available, three SOMs (or three FAs) will be trained, one for each class. The several SOMs (or FAs), however, are trained independently, using only the data vectors of the class it represents. There is no need for the several SOMs (or FAs) to have the same number of neurons, unless for the sake of simplicity. During testing, the winning neuron is searched among the neurons of all available SOM (or FA) networks, so that its class label is assigned to the current input vector.

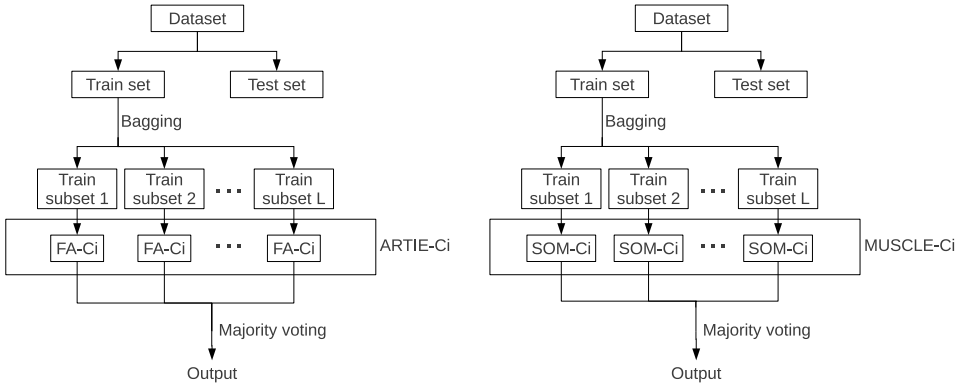
As occurs for the Strategy C1, the Strategy C2 has been used before for the design of SOM-based classifiers [Souza Júnior et al. 2011, Biebelmann et al. 1996]; however, it seems that it was never tested before with the FA network.

3.1.3. Strategy C3: Augmenting Input Space Dimension with Class Labels

In the third strategy, the SOM (or FA) network is made supervised by adding class information to each input data vector. Specifically, the input vectors $\mathbf{x}(k)$ are now formed of two parts, $\mathbf{x}_p(k)$ and $\mathbf{x}_l(k)$, where $\mathbf{x}_p(k)$ is the pattern vector itself and $\mathbf{x}_l(k)$ is the corresponding class label of $\mathbf{x}_p(k)$. During training, these vectors are concatenated to build augmented vectors $\mathbf{x}(k) = [\mathbf{x}_p(k)\ \mathbf{x}_l(k)]^T$ which are used as inputs to the SOM (or FA) network. The corresponding augmented weight vectors, $\mathbf{w}_j(k) = [\mathbf{w}_j^p(k)\ \ \mathbf{w}_j^l(k)]^T$, are adjusted as in the usual SOM/FA training procedure.

During recognition of an unknown vector $\mathbf{x}(k)$, only its $\mathbf{x}_p$ part is compared with the corresponding part of the weight vectors. Then, the class label of the unknown pattern vector is decided on the basis of the $\mathbf{w}_{i^*}^l(k)$ part of the winning weight vector $\mathbf{w}_{i^*}(k)$. The index of the component of $\mathbf{w}_{i^*}^l(k)$ with highest value defines the class label of $\mathbf{x}(k)$.

As occurs for the Strategies C1 and C2, the Strategy C3 has been used before for the design of SOM-based classifiers [Xiao et al. 2005, del-Hoyo et al. 2003, Kohonen 1988]; however, it seems that it was never tested before with the FA network.



(a) Block diagram of the ARTIE model. (b) Block diagram of the MUSCLE model.

Figure 1. ARTIE and MUSCLE models.

3.2. ARTIE: *ART* networks *I*n *E*nsembles

The Fuzzy ARTMAP (FAM) network is a supervised learning method that utilizes FA as building block. For being a supervised classifier, FAM is a natural choice for the base classifier in building EL models [Santos and Canuto 2008, Loo et al. 2006]. However, as described previously, the Strategies C1, C2 and C3 can be used to turn the FA network into a supervised classifier. This way, FA networks can also serve as base learners for building EL models for classification. This approach gives rise to the ARTIE model.

The immediate advantage in using FA-based classifiers instead FAM-based as base learners is related with the computational costs. Training an ARTIE model requires half the number of FA networks required by an ensemble of ARTMAP-like networks and, hence, half the number of parameters. Since FA is simpler than FAM, building ARTIE classifiers is expected to be much easier.

Each one of ARTIE's L base classifiers are trained independently using different subsets of the original training set. Theses subsets are created using the Bagging[1] (Bootstrap Aggregating) method with the objective of generating diversity [Breiman 1996]. Figure 1(a) shows the block diagram of the ARTIE model. The output of ARTIE is decided with a simple majority voting process within the results of the single classifiers.

It is necessary to determine some parameters of FA networks for their use in ARTIE models, namely: the vigilance parameter ρ, the choice parameter β and the learning step η. In order to find almost optimal values for these problem-specific parameters, a hybrid PSO strategy is introduced in Section 3.4.

3.3. MUSCLE: *MU*ltiple *S*OM *CL*assifiers in *E*nsembles

As mentioned in Section 1, the SOM network has been been used in EL classifiers [Corchado et al. 2007, Petrikieva and Fyfe 2002]. For this purpose, the Strategy C1 has been the chosen one. However, Strategies C2 and C3 can also be used for the same purpose. Thus, in doing so, we aimed at extending the range of alternative ways of building SOM-based EL classifiers. This approach gives rise to the MUSCLE model.

[1]Bagging is carried out by sampling (with replacement) training examples, forming new training sets, usually with the same size of the original one. For a training set of N samples and N being large enough, this procedure causes each sample to have a probability of $\left(\frac{N-1}{N}\right)^N \approx 0.368$ of not being chosen.

MUSCLE's base classifiers have the same dimensions and parameters. Diversity is obtained through random weight initialization and using the Bagging for data set selection. The final result is obtained by a simple majority voting among the base classifiers, as with the ARTIE model. Figure 1(b) shows the block diagram of the MUSCLE model.

As the ARTIE model, the MUSCLE model also needs some parameters to be tuned. The main parameters are the dimensions of the maps (P_1 and P_2), the initial and final learning steps (η_0 and η_f) and the initial and final spread parameters (σ_0 and σ_f). In order to find almost optimal values for all these problem-specific parameters, a hybrid PSO strategy is introduced in the next section.

3.4. An Improved Hybrid Particle Swarm Optimization

The Particle Swarm Optimization (PSO) method [Kennedy and Eberhart 1995] is inspired by the social behavior and auto-organization of bird flocking and fish schooling. The exchange of information among the population generates exploration for better solutions, while the individual learning corresponds to the exploitation component, providing a combination of global and local search to the algorithm [Pedersen and Chipperfield 2010].

In [He and Wang 2007] it is proposed a variant of the original PSO algorithm, called Hybrid Particle Swarm Optimization (HPSO), by adding feasibility-based rules for handling constraints and a local search step based on Simulated Annealing (SA). In this paper, for the sake of parameter optimization of the base classifiers of the ARTIE and MUSCLE models, we incorporate the features of the HPSO algorithm into the PSO Standard 2007 [Bratton and Kennedy 2007], resulting in an improved version of the HPSO algorithm, called from now onwards, the Improved HPSO (I-HPSO) algorithm.

Let $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{v}_i \in \mathbb{R}^d$ be, respectively, the position and velocity vectors of the i-th element in a swarm of particles, where d is the number of variables of the problem. Let also $\mathbf{p}_i \in \mathbb{R}^d$ and $\mathbf{pl}_k \in \mathbb{R}^d$ be, respectively, the vectors of best historical individual position of the i-th particle and the best historical position of the neighborhood k.

For a given objective function $f(\cdot)$, the I-HPSO algorithm is implemented according the following steps for a total of L_{PSO} generations:

Step 1 Let $m = 1$ and do $\mathbf{x}_i(0) = \mathbf{x}_{min} + (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{U}$, $\mathbf{v}_i(0) = (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{U} - \mathbf{x}_i(0)$, $\mathbf{p}_i(0) = \mathbf{0}$ and $\mathbf{pl}_k(0) = \mathbf{0}$, where $\mathbf{U}$ is a d-dimensional random vector whose components are uniformly distributed in the interval $[0, 1]$, $\mathbf{0}$ is a d-dimensional null vector and $\mathbf{x}_{max}$ and $\mathbf{x}_{min}$ are, respectively, the maximum and the minimum limits for the components of $\mathbf{x}_i$.

Step 2 Evaluate all the particles of the swarm. The vector $\mathbf{p}_i(m)$ of each particle receives the current position, as well as its associated value for the objective function. The vector $\mathbf{pl}_k(m)$ and its objective value receive the best position and the best objective function among the particles of the neighborhood k.

Step 3 Considering $\mathbf{x}_i(m)$ and $\mathbf{v}_i(m)$ respectively as the position and velocity of the particle i during the current iteration m and k the neighborhood of the i-th particle, calculate the following update equations:

$$\mathbf{v}_i(m+1) = \chi\{\mathbf{v}_i(m) + c_1 r_1[\mathbf{p}_i(m) - \mathbf{x}_i(m)] + c_2 r_2[\mathbf{pl}_k(m) - \mathbf{x}_i(m)]\}, \quad (8)$$

$$\mathbf{x}_i(m+1) = \mathbf{x}_i(m) + \mathbf{v}_i(m+1). \quad (9)$$

where χ is the constriction factor, c_1 and c_2 are positive constants called acceleration coefficients, while r_1 and r_2 are independent random variables uniformly distributed in the interval $[0, 1]$.

Step 4 Evaluate all the particles of the swarm.

Step 5 For each particle i, if $f(\mathbf{x}_i(m)) > f(\mathbf{p}_i(m))$, $\mathbf{p}_i(m)$ is updated with $\mathbf{x}_i(m)$.

Step 6 For each neighborhood k, let $\mathbf{p}_{k_{max}}(m) = \arg\max(f(\mathbf{p}_i(m)))$ for all i referred to particles within the considered neighborhood. If $f(\mathbf{p}_{k_{max}}(m)) > f(\mathbf{pl}_k(m))$, $\mathbf{pl}_k(m)$ is updated with the solution $\mathbf{p}_{k_{max}}(m)$.

Step 7 Randomly select a fraction of the neighborhoods and perform the local search step based on SA in the solutions $\mathbf{pl}_k$ for each selected value of k.

Step 8 Let $m = m + 1$. If $m > L_{PSO}$, stop and output the best $\mathbf{pl}_k(m)$ of all neighborhoods as the best solution found. Otherwise, go to Step 3.

The local search step is performed only over a fraction of the neighborhoods (in this paper only 10%), randomly chosen, to speedup the search process. Let $\mathbf{pl}_k(m) \in \mathbb{R}^d$ be the best solution found in the k-th neighborhood until the generation m and $p_a \in [0, 1]$ be the acceptance probability of a new solution. Let also $t(m)$ be a temperature parameter. The local search is done according the following steps during the generation m of I-HPSO algorithm for a total of L_{SA} iterations:

Step 1 Do $n = 1$ and $\mathbf{pl}'_k = \mathbf{pl}_k(m)$.

Step 2 Generate a new solution using the following equation:

$$\mathbf{x}' = \mathbf{pl}'_k + \eta_{SA}(\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

where η_{SA} is an incremental step, $\mathbf{x}_{max}$ and $\mathbf{x}_{min}$ are, respectively, the maximum and the minimum allowed values for the variables of a solution and $\mathbf{N}(\mathbf{0}, \mathbf{I})$ is a d-dimensional Gaussian random vector with zero mean vector and the identity matrix as the covariance matrix.

Step 3 Calculate the acceptance probability $p_a = \min\left\{1, \exp\left[\frac{f(\mathbf{pl}'_k) - f(\mathbf{x}')}{t(m)}\right]\right\}$.

Step 4 If $p_a \geq U(0, 1)$, where $U(0, 1)$ is a random number uniformly distributed in the interval $[0, 1]$, do $\mathbf{pl}'_k = \mathbf{x}'$.

Step 5 Do $n = n + 1$. If $n \geq L_{SA}$, stop and do $\mathbf{pl}_k(m + 1) = \mathbf{pl}'_k$, or else, go to Step 2.

The parameter t is initiated with an empirical value [He and Wang 2007]: $t(0) = -\frac{f_{max} - f_{min}}{\ln(0.1)}$, where f_{max} and f_{min} are respectively the maximum and minimum values for the objective function in the initial swarm. Along the generations, t is reduced exponentially, i.e. $t(m + 1) = \lambda t(m)$, where the annealing rate λ satisfies $0 < \lambda < 1$.

In order to apply I-HPSO to parameter optimization, each particle of the swarm is formed by a vector of values for the tunable parameters and the objective function $f(\cdot)$ is chosen to be the accuracy obtained by the classifier with a possible set of parameters.

4. Simulations and Discussion

Tests with 10 real-world benchmarking datasets were carried out. We used 9 UCI datasets (Balance, Breast-w, Dermatology, Glass, Heart, Sonar, Vehicle, Wall-Following, Zoo) [Frank and Asuncion 2010] and the vertebral column pathologies dataset described in [Rocha Neto and Barreto 2009], named henceforth VCP dataset, which is available upon request. A summary of the evaluated datasets are presented in Table 1.

Table 1. Summary of the benchmarking datasets.

	Instances	Features	Classes
Balance	625	4	3
Breast-w	683	9	2
Dermatology	358	34	6
Glass	214	9	6
Heart	270	13	2
Sonar	208	60	2
VCP	310	6	3
Vehicle	846	18	4
Wall-Following	5456	2	4
Zoo	101	16	7

Table 2. Results from classification problems. The accuracy and standard deviation are percentages.

	Balance	Breast-w	Dermat.	Glass	Heart	Sonar	VCP	Vehicle	Wall-F.	Zoo
ARTIE-C1	86.08 ±5.22	97.23 ±2.20	96.05 ±3.64	65.30 ±10.82	82.22 ±6.49	82.64 ±9.72	77.42 ±6.27	68.60 ±6.24	97.21 ±0.38	**97.09** ±4.69
ARTIE-C2	86.09 ±3.26	96.35 ±2.57	**98.11** ±2.18	73.71 ±8.39	80.0 ±8.76	88.0 ±7.53	80.65 ±6.27	**73.78** ±5.90	**99.95** ±0.09	96.09 ±6.91
ARTIE-C3	84.66 ±4.12	97.09 ±2.72	97.20 ±1.91	73.87 ±5.12	**83.70** ±5.84	84.5 ±11.17	83.87 ±5.89	71.44 ±6.16	99.87 ±0.19	94.09 ±6.94
MUSCLE-C1	85.45 ±3.42	97.22 ±2.23	95.53 ±4.07	72.29 ±9.11	79.63 ±9.44	82.29 ±10.61	**85.81** ±9.40	66.49 ±7.29	95.47 ±0.52	96.09 ±5.05
MUSCLE-C2	86.58 ±3.98	97.08 ±1.92	97.14 ±3.01	70.06 ±3.95	82.59 ±7.82	85.64 ±7.83	84.19 ±6.71	71.10 ±5.35	98.02 ±0.59	94.09 ±6.94
MUSCLE-C3	**90.59** ±3.36	**97.67** ±2.28	98.05 ±1.91	**74.42** ±7.02	78.52 ±7.96	**89.0** ±10.75	85.16 ±6.66	69.45 ±5.91	96.63 ±0.61	94.09 ±5.09
ELM ensemb.	89.96 ±3.36	96.80 ±2.71	96.96 ±2.75	66.72 ±8.44	82.59 ±8.56	77.71 ±7.28	84.84 ±9.26	72.27 ±7.81	94.74 ±1.11	92.09 ±6.30
FAM ensemb.	86.75 ±3.15	96.65 ±2.64	98.0 ±2.35	72.76 ±8.90	79.63 ±8.95	83.36 ±6.36	81.61 ±8.61	72.11 ±6.13	99.73 ±0.28	93.09 ±6.71

For each problem and base classifier a set of parameters was determined using I-HPSO. The optimization process was realized with swarms of 20 particles, a total of 25 generations and 10 iterations of local search. The other parameters of I-HPSO were $c_1 = c_2 = 2.05$, $\chi = 0.72984$, $\lambda = 0.94$ and $\eta_{SA} = 0.001$.

For all simulations, ARTIE-Ci and MUSCLE-Ci, i =1, 2, and 3, are comprised of $L = 10$ base classifiers. The 10 base classifiers are constrained to adopt a single supervision strategy (C1, C2 or C3). This was done in order to have a better idea of the influence of the type of supervision strategy on the EL performance.

For the sake of completeness, performance comparisons between the proposed models and ensembles of ELM and FAM networks are carried out. The ELM/FAM ensembles also used $L = 10$ classifiers and were trained through the Bagging method. Decisions were also made through the majority voting rule.

The performance evaluation procedure follows the approach recommended by Salzberg [Salzberg 1997], which applies 10-fold cross-validation and McNemar's test [Everitt 1977] for comparing the resulting models. The results for each evaluated dataset are shown in the columns of Table 2, where the best values are highlighted in bold.

It is worth noting that ARTIE and MUSCLE presented the best results for all datasets in comparison with ensembles of ELM/FAM classifiers. Indeed, the best among the proposed ARTIE/MUSCLE models performed always better the ensembles

Table 3. McNemar's test for $\alpha = 0.05$. Results are in units of datasets. Winners are in bold. Ties are broken evaluating the best results in Table 2.

Classifier A	Classifier B	A $\approx$ B	A $>$ B	A $<$ B
ELM ensemb.	ARTIE-C1	7	2	1
ELM ensemb.	**ARTIE-C2**	5	2	3
ELM ensemb.	**ARTIE-C3**	6	1	3
ELM ensemb.	**MUSCLE-C1**	7	1	2
ELM ensemb.	**MUSCLE-C2**	7	1	2
ELM ensemb.	**MUSCLE-C3**	8	-	2
FAM ensemb.	ARTIE-C1	7	3	-
FAM ensemb.	**ARTIE-C2**	9	-	1
FAM ensemb.	**ARTIE-C3**	9	-	1
FAM ensemb.	MUSCLE-C1	8	2	-
FAM ensemb.	MUSCLE-C2	9	1	-
FAM ensemb.	**MUSCLE-C3**	8	1	1

of ELM/FAM classifiers. The ARTIE-C2 and MUSCLE-C3 achieved the best overall performances, presenting best accuracy in 30% and 40% of the datasets, respectively.

As a final evaluation, we applied the McNemar's statistical test as described in [Dietterich 1998] for comparing the performances of the classifiers, taken in pairs. The objective is to estimate the probability p of two given classifiers being equivalent (null hypothesis) from a score γ calculated with the McNemar's test. For a significance value of $\alpha = 0.05$, if $p < \alpha$ it is unlikely that two classifiers are equivalent, i.e. the null hypothesis is rejected. In this case, the classifier with better mean accuracy is considered the best for that dataset. Results are summarized in Table 3.

ARTIE-C2, ARTIE-C3 and MUSCLE-C3 models performed better in general than both ELM- and FAM-based ensembles. ARTIE-C1 was the only variant that presented poor results. It is worth mentioning, however, that McNemar's test only claims that two algorithms are different when one of them almost outperforms the other [Dietterich 1998]. Thus, when it suggests that the two classifiers are similar, it is recommended to evaluate the performance metrics shown in Table 2 to choose the best one.

5. Conclusion and Further Work

Although both FA and SOM networks are originally unsupervised competitive learning algorithms, three different techniques were described with the goal of converting them into supervised learning methods. With the supervised variants of the FA and SOM networks at hand, one can build ensemble classifiers with them, giving rise to the ARTIE and MUSCLE models, respectively. In addition, the problem of choosing the appropriate parameters for training base classifiers, for each classification task, was tackled using a metaheuristic approach, which led to the development of the I-HPSO algorithm. A comprehensive performance evaluation was then carried out for 10 different real-world datasets in order to compare the ARTIE and MUSCLE variants with ensembles built from standard supervised classifiers, such as ELM and FAM networks. The obtained results indicates the superior performances of the proposed ARTIE and MUSCLE models.

In this paper we have developed ensembles built using a single supervised variant of the FA and SOM networks. Further work will evaluate the role of diversity of the base classifiers in building efficient ARTIE and MUSCLE models. The goal is to build ensembles whose base classifiers are different variants of the FA and SOM networks.

References

Barreto, G. A. and Aguayo, L. (2009). Time series clustering for anomaly detection using competitive neural networks. In Principe, J. C. and Miikkulainen, R., editors, *Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps (WSOM'09)*, volume LNCS-5629, pages 28–36. Springer.

Biebelmann, E., Köppen, M., and Nickolay, B. (1996). Pratical aplications of neural networks in texture analysis. *Neurocomputing*, 13(2–4):261–279.

Bratton, D. and Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 120–127, Honolulu, Hawaii.

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

Carpenter, G. and Grossberg, S. (1987a). Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, 26:4919–4930.

Carpenter, G., Grossberg, S., and Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4(5):565–588.

Carpenter, G. A. and Grossberg, S. (1987b). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37(1):54–115.

Carpenter, G. A., Grossberg, S., and Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning, categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6):759–771.

Christodoulou, C. I., Michaelides, S. C., and Pattichis, C. S. (2003). Multifeature texture analysis for the classification of clouds in satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 41(11):2662–2668.

Corchado, E., Baruque, B., and Yin, H. (2007). Boosting unsupervised competitive learning ensembles. In de Sá, J. M., Alexandre, L. A., Duch, W., and Mandic, D. P., editors, *Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN'07), Part I*, volume LNCS 4668, pages 339–348. Springer.

del-Hoyo, R., Buldain, D., and Marco, A. (2003). Supervised classification with associative SOM. In *Proceedings of the 7th International Work-Conference on Artificial and Neural Networks, (IWANN)'03*, pages 334–341.

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

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

Everitt, B. (1977). *The Analysis of Contingency Tables*. Chapman & Hall/CRC.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Hansen, L. K. and Salamon, P. (2002). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Haykin, S. (2008). *Neural Networks and Learning Machines*. Prentice Hall, 3rd edition.

He, Q. and Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics, Computation*, 186(2):1407–1422.

Huang, G. B., Zhu, Q. Y., and Ziew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489–501.

Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Piscataway, NJ, USA.

Keskin, G. A. and Özkan, C. (2009). An alternative evaluation of FMEA: Fuzzy art algorithm. *Quality and Reliability Engineering International*, 25(6):647–661.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kohonen, T. (1988). The "neural" phonetic typewriter. *Computer*, 21(3):11–22.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, active learning. *Advances in Neural Information Processing Systems*, pages 231–238.

Loo, C. K., Law, A., Lim, W. S., and Rao, M. V. C. (2006). Probabilistic ensemble simplified fuzzy ARTMAP for sonar target differentiation. *Neural Computing & Applications*, 15(1):79–90.

Pedersen, M. E. H. and Chipperfield, A. J. (2010). Simplifying particle swarm optimization. *Applied Soft Computing*, 10(2):618–628.

Petrikieva, L. and Fyfe, C. (2002). Bagging and bumping self-organising maps. *Computing and Information Systems*, 9(2):69.

Rocha Neto, A. R. and Barreto, G. A. (2009). On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions*, 7(4):487–496.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328.

Santos, A. M. and Canuto, A. M. P. (2008). Using ARTMAP-based ensemble systems designed by three variants of boosting. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'08)*, pages 562–571.

Souza Júnior, A. H., Barreto, G. A., and Varela, A. T. (2011). A speech recognition system for embedded applications using the SOM and TS-SOM networks. In Mwasiagi, J. I., editor, *Self-Organizing Maps - Applications and Novel Algorithm Design*. InTech.

Wyns, B., Sette, S., Boullart, L., Baeten, D., Hoffman, I. E. A., and De Keyser, F. (2004). Prediction of diagnosis in patients with early arthritis using a combined Kohonen mapping and instance-based evaluation criterion. *Artificial Intelligence in Medicine*, 31(1):45–55.

Xiao, Y.-D., Clauset, A., Harris, R., Bayram, E., Santago, P., and Schmitt, J. D. (2005). Supervised self-organizing maps in drug discovery. 1. robust behavior with overdetermined data sets. *Journal of Chemical Information and Modeling*, 45(6):1749–1758.