Design of Artificial Neural Networks using a Memetic Pareto Evolutionary Algorithm using as objectives Entropy versus Variation Coefficient

J. C. Fernández[1], C. Hervás[1], F. J. Martínez[2], M. Cruz[1]
[1]*Department of Computer Science. University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain.*
[2]*Department of Management and Quantitative Methods, ETEA, Escritor Castilla Aguayo 4, 14005, Córdoba, Spain*
[1]*{jcfernandez, chervas, i42cram}@uco.es,* [2]*fjmestud@etea.com*

Abstract

This paper proposes a multi-classification pattern algorithm using multilayer perceptron neural network models which try to boost two conflicting main objectives of a classifier, a high correct classification rate and a high classification rate for each class. To solve this machine learning problem, we consider a Memetic Pareto Evolutionary approach based on the NSGA2 algorithm (MPENSGA2), where we defined two objectives for determining the goodness of a classifier: the cross-entropy error function and the variation coefficient of its sensitivities, because both measures are continuous functions, making the convergence more robust. Once the Pareto front is built, we use an automatic selection methodology of individuals: the best model in accuracy (upper extreme in the Pareto front). This methodology is applied to solve six benchmark classification problems, obtaining promising results and achieving a high classification rate in the generalization set with an acceptable level of accuracy for each class.

1. Introduction

One of the fundamental problems of machine learning is the classification of examples into two or more classes using traditionally the correct classification rate or accuracy to measure the performance of a classifier, generally avoiding the presentation of the classification level of each class in the results. However, the pitfalls of using only accuracy have been pointed out by several authors [1].

Neural networks [2] have been an object of renewed interest among researchers, both in statistics and computer science, owing to the significant results obtained in a wide range of classification and pattern recognition problems. Many different types of neural network architectures have been used, but the most popular one is Multilayer Perceptron, MLP. In the context of Artificial Neural Network (ANN) design, evolutionary optimization has led to the development of Evolutionary ANNs (EANNs), in which adaptation is performed primarily by means of evolution. EANNs have been shown to possess several advantages over conventional methods of training [3].

Traditionally, the design of ANNs involves the optimization of two competing objectives, the maximization of network capacity and the minimization of neural architecture complexity. Therefore, it is not surprising that Multi-Objective Evolutionary Algorithms (MOEAs) have been applied with great success to the concurrent optimization of both architecture and connection weights [4,5]. Although these two objectives are those habitually analyzed, this study deals with Cross-Entropy, E , and the Variation Coefficient of the Sensitivities, VC , to optimize the goodness of the classifier, which will be explained in detail in the following section.

To train ANNs an adaptive and improvement MOEA is used; we also introduce a local search method. Our algorithm determines the optimal structure of the networks while simultaneously optimizing its corresponding parameters.

In summary, the objective of this paper is to propose a new approach for classification based on a two-dimensional performance measure associated with multi-class problems by a Memetic MOEA to design ANN models, and whose purpose is to obtain classifiers with a high accuracy level and also an acceptable classification level for each class.

The organization of this paper is as follows. In the next section, E *and* VC measures are proposed and discussed. Section 3 presents an overview of Multi-Objective Evolutionary Neural networks. Section 4

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.153

describes the MPENSGA2 algorithm and our problem is portrayed as a multi-objective Pareto-based optimization problem. Section 5 shows the experimental design and Section 6 the main conclusions.

2. Cross-Entropy versus Variation Coefficient

Let's set a classification problem with Q classes and N training or testing patterns. We define the performance of a classifier, g, by means of the corresponding $Q \times Q$ contingency or confusion matrix $M(g) = \left\{ n_{ij}; \sum_{i,j=1}^{Q} n_{ij} = N \right\}$, where n_{ij} represents the number of times that the patterns are predicted by classifier g to be in class j when they really belong to class i. The diagonal corresponds to correctly classified patterns and the off-diagonal corresponds to mistakes in the classification task.

Let us denote the number of patterns associated with class i by $f_i = \sum_{j=1}^{Q} n_{ij}, i = 1,\ldots,Q$. Then, $S_i = n_{ii} / f_i$ is the number of patterns correctly predicted to be in class i with respect to the total number of patterns in this class, that is, the Sensitivity for class i.

From the above quantities, two statistics based on those sensitivities are defined: Sensitivity S of the classifier as the minimum value of sensitivities for each class, $S = \min(S_i;\ i = 1,...,Q)$; and the correct classification rate or accuracy, C, defined as $C = \sum_{j=1}^{Q} \frac{f_j}{N} S_j$, that is, the rate of all the correct predictions or the weighted average of the sensitivities in which the weights depend on the data set.

It is straightforward to prove that these two measures verify that $S \le C \le 1 - (1 - S) p^*$, p^* being the minimum of the estimated *prior* probabilities, a value that has an important role in the relationship between the two measures. Therefore, from previous inequality, each classifier will be represented as a point in the region in Figure 1-B.

In general, C and S could be cooperative, but as we approach the $(1,1)$ point in (S,C) space, the objectives become competitive and an improvement in one objective tends to involve a decrease in the other one, which justifies the use of a MOEA.

This problem is especially significant when dealing with classification problems that differ in their prior class probabilities (class imbalances) or where there are a great number of classes. In this way, the (S,C) pair tries to find a point between the scalar accuracy measure and the multidimensional ones based on misclassification rates.

Once the relationship between Accuracy and Sensitivity has been explained, our objectives in this work are to try to optimize:

- The Cross-Entropy or E, defined as:

$$l(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{Q} y_n^{(l)} \log g_l(x_n, \boldsymbol{\theta}_l)$$

- The Variation Coefficient of the sensitivities of all classes defined as:

$$VC = \frac{\sqrt{\sum_{i=1}^{Q} \left(S_i - \overline{S}\right)^2}}{\overline{S}}$$

where $\overline{S}$ is the average of the sensitivities.

The first objective tries to maximize the accuracy of the MLP models. With the second objective we minimize the relative variance of Correctly Classified Rate percentages for each class with respect to the total number of examples in the corresponding class. Also both measures are continuous, which makes the search more parsimonious and the convergence more robust.

The (VC,E) pair expresses two features associated with a classifier: global performance, C, and the rate of the worst classified class, S (Figure 1-B). From this perspective we work with two objectives that are not being used for multiclass problems in the design of ANN models at this moment.

3. Multi-Objective Evolutionary Neural Networks

Training ANNs with Evolutionary Algorithms is a powerful approach to address the exploitation/exploration dilemma. Selecting the size architecture of a neural network for a particular application is a difficult task. The architecture of the neural network directly affects two of the most important factors of neural network training: generalization and training efficiency and efficacy.

EANNs can automatically find the best possible architecture for a neural network. There have been many applications for parametric learning (evolving the weights of the network) and for both parametric and structural learning (evolving the weights and the number of the hidden nodes and connections simultaneously) [6]. On the other hand, Pareto-based techniques, specifically the MOEAs, should provide a

homogeneous distribution of population along the Pareto frontier together with an improvement in solutions throughout successive generations. Also, these techniques present an uncountable set of solutions that, when evaluated, produce vectors whose components represent a trade-off in objective space. A decision maker then implicitly chooses an acceptable solution (or solutions) by selecting one or more of these vectors; in our case, the algorithm automatically chooses the upper extreme of the Pareto front.

A limited number of studies use a MOEA to train a population of multi-objective ANNs, which are usually used to minimize the error over the training set and the complexity of the network [7,8].

One improvement in the EA is the incorporation of local search procedures throughout evolution. Therefore, in the combination EAs carry out a global search inside the space of solutions, locating the ANNs near the global optimum, so the local procedure could get to the best solution quickly and efficiently. This type of algorithm receives the name of memetic or hybrid algorithm [9]. Gradient descent techniques are the most widely used class of local search algorithms for supervised learning in neural networks. In this paper a new version of the Resilient Backpropagation algorithm (*Rprop*) [10] is used, because it is the best of these techniques known to the authors with respect to its parameters in terms of convergence speed, accuracy and robustness.

4. The MPENSGA2 Multi-Objective Evolutionary Algorithm

We consider standard feed forward MLP neural networks with one input layer with independent variables, one hidden layer and one lineal output layer, interpreting the outputs of the neurons on the output layer from a probability point of view which considers the softmax activation function given by:

$$g_l(\mathbf{x},\boldsymbol{\theta}_l)=\frac{\exp f_l(\mathbf{x},\boldsymbol{\theta}_l)}{\sum_{l=1}^{Q}\exp f_l(\mathbf{x},\boldsymbol{\theta}_l)},\ l=1,2,...,Q$$

where Q is the number of classes in the problem, $f_l(\mathbf{x},\boldsymbol{\theta}_l)$ the output of the j neuron for pattern $\mathbf{x}$ and $g_l(\mathbf{x},\boldsymbol{\theta}_l)$ the probability pattern $\mathbf{x}$ has of belonging to class q. Taking this consideration into account, it can be seen that the class predicted by the neuron net corresponds to the neuron on the output layer whose output value is the greatest. The optimum rule $C(\mathbf{x})$ is the following:

$$C(\mathbf{x})=\hat{l},\ \text{where}\ \hat{l}=\arg\max_l g_l(\mathbf{x},\hat{\boldsymbol{\theta}}),\ \text{for}\ l=1,2,...,Q$$

The functional model considered in this work is:

$$f_l(\mathbf{x},\boldsymbol{\theta}_l)=\beta_0^l+\sum_{j=1}^{M}\beta_j^l\sigma_j(\mathbf{x},\mathbf{w_j}),\ \ l=1,2,...,Q$$

where $\boldsymbol{\theta}=(\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_Q)^T$ is the transpose matrix containing all the neural net weights, $\boldsymbol{\theta}_l=(\beta_0^l,\beta_1^l,...,\beta_M^l,\mathbf{w}_1,...,\mathbf{w}_M)$ is the vector of weights of the l output node, $\mathbf{w}_j=(w_{1j},...,w_{Kj})$ is the vector of weights of the connections between the K nodes of the input layer and the j hidden node, Q is the number of classes in the problem, M is the number of sigmoidal units in the hidden layer, $\mathbf{x}$ is the input pattern,

$$\sigma_j(\mathbf{x},w_j)=\frac{1}{1+e^{-\left(w_0^j+\sum_{i=1}^{K}w_i^j x_i\right)}}$$

and K the number of features in each pattern.

As soon as the previous definitions and explanations have been made, the first fitness function used in this research to evaluate a classification model is the function of Cross-Entropy error and is given by the following expression for Q classes:

$$l(\boldsymbol{\theta},g)=-\frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{Q}y_n^{(l)}\log g_l(x_n,\boldsymbol{\theta}_l)$$

The advantage of using the error function $l(\boldsymbol{\theta})$ instead of $(1-C)$ is that it is a continuous function. Then, the fitness measure to maximize is a strictly decreasing transformation of the entropy error $l(\boldsymbol{\theta})$ given by $A_1(g)=\frac{1}{1+l(\boldsymbol{\theta},g)}$, where g is a sigmoidal basis function model given by the multivaluated function $g(\mathbf{x},\boldsymbol{\theta})=\left(g_1(\mathbf{x},\boldsymbol{\theta}_1),...,g_Q(\mathbf{x},\boldsymbol{\theta}_Q)\right)$. The second fitness function to be maximized that is used in this paper is then a strictly decreasing transformation of the Variation Coefficient of the sensitivities S_l, that is, $A_2(g)=\frac{1}{1+VC(g)}$. The use of two continuous functions as fitness functions allows the training to converge towards more optimal solutions and the evolutionary algorithm to converge more slowly, which is very necessary for databases with a low number of patterns like those used in this paper.

The MOEA proposed is NSGA2 [11], adding the necessary operators to obtain new individuals in the evolutionary process, and the local search algorithm is the *improved Rprop—IRprop+* [12]. The algorithm in this work is called MPENSGA2 (Memetic Pareto Evolutionary NSGA2). The *IRprop+* algorithm is applied when we combine parent and offspring populations. Then, only the individuals from the first

Pareto front are optimized by *IRprop+*, reducing the computational cost considerably. *IRprop+* can be seen as a kind of life-time learning (for the first objective only) within a generation. Life-time learning is done in generations 2/7, 4/7 y 6/7 of the number of generations.

Operators used in this work are divided in structural mutation: add/delete neurons, add/delete connections, and parametric mutation: in this case a new parametric mutation that involves the alteration of all the weights in the network by adding a Gaussian noise, mean zero, where the variance of Gauss distribution follows a geometric decline (for details, see [13,14]).

5. Experiments

For the experimental design we consider six datasets taken from the UCI repository. The design was conducted using a stratified holdout procedure with 30 runs, where approximately 75% of the patterns were randomly selected for the training set and the remaining 25% for the test set.

In Table 1 we show the number of instances in training and testing, the number of inputs, the number of classes and the total number of instances per class. The experiments were carried out using a software package developed in JAVA by the authors, that can be obtained from KEEL [15], as an extension of the JCLEC framework (http://jclec.sourceforge.net/) [16].

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. For *IRprop+*, the adopted parameters are $\eta^- = 0.5$ (decreasing factor for step-size), $\eta^+ = 1.2$ (increasing factor for step-size), $\Delta_0 = 0.0125$ (the initial value of the step-size for the weights), $\Delta_{\min} = 0$ (minimum step-size for the weights), $\Delta_{\max} = 50$ (maximum step-size for the weights), and $Epochs = 25$ (number of epochs for the local optimization).

The algorithmic structure is similar to NSGA2[11] but once the Pareto front is built, an automatic selection methodology of individuals is used, obtaining the best model in E (upper extreme for the Pareto front). This allows us to compare the C and S values for the generalization set with other classification methods. As our procedures are stochastic, the MPENSGA2 algorithm is run 30 times. In each execution, once the first Pareto front is calculated, we chose the upper extreme value in training, that is, the best individual on E. This individual is called EI (Entropy Individual). Once this is done, we get the values of accuracy C and Sensitivity S in the generalization set for the best EI individual, thus giving an individual $EI_{testing} = (C_{EI\_testing}, S_{EI\_testing})$ for one run. This is repeated 30 times, then the average and standard deviation obtained from 30 EI individuals is calculated, $\overline{EI}_{testing} = (\overline{C}_{EI\_testing}, \overline{S}_{EI\_testing})$. Hence, we have called MPENSGA2-EVC to the procedure to obtain the average $\overline{EI}_{testing}$.

TABLE 1. CHARACTERISTICS FOR UCI DATASETS.

Dataset	#Tra	#Test	#Input	#Clas	#per class
ACard	517	173	51	2	307-383
BreastC	215	71	15	2	201-85
Ionosp.	263	88	34	2	126-225
Iris	111	39	4	3	50-50-50
Newth	161	54	5	3	150-35-30
Vote	326	109	16	2	267-168

In Table 2 we present the values of the mean and the standard deviation (mean±SD) for C and S obtained for the best models in each run through the generalization set. MPENSGA2-EVC is compared with three algorithms for designing ANNs: an EA with two stages, called $E + A$ [17], a version of NSGA2 for Neural Networks [18] and a MOEA based on Differential Evolution [4]. From analyzing the results obtained, we can draw the following conclusions:

- VC and E usually guide the algorithm towards regions in the (S,C) space with high C and acceptable sensitivity S. The results confirm that E and VC are robust fitness functions for obtaining classifiers with a high level of classification in all classes and for each class.
- The MPENSGA2-EVC approach improves accuracy levels, maintaining high values of sensitivity in Australian Card, Breast Cancer, Ionosphere, Newthyroid and Vote.

To analyze the statistical significance of the differences, in mean, observed in each dataset performance, we have carried out an ANOVA statistical method or the non parametric Kruskal-Wallis, K-W, test, depending on the satisfaction of the normality hypothesis of the C and S measures of the best models as the generalization set (previously evaluating if the C and S values follow normal distribution, using a Kolmogorov-Smirnov test). The tests show that the effect of the different methodologies in the C and S measures is statistically significant, in the mean of C and S, for Ionos, Newthyroid and Vote data sets and for BreastC, only in C, at a significance level, α, of 5%, i.e.,

$\alpha > p$-value (see first row for C and S in Table 3).

Based in the last results, for BreastC (only for C), Ionos and Vote, a post-hoc means multiple comparison Tukey test [19] is carried out. For Newthyroid dataset significant differences exist based on a K-W test (p-value>0.05), and then we perform a pair-wise Mann-Witney test [20]. Table 3 shows the results obtained (in the second row for C and S for each dataset).

If we analyze the test results for accuracy C, we observe that the MPENSGA2-EVC methodology obtains mean values significantly better than those obtained with E+A in Vote, with MPENSGA2-E in Newthyroid and with MPANN in BreastC and Newthyroid, and it obtained similar results with respect to other methodologies in the rest of datasets. On the other hand, the results of mean sensitivity S show that the MPENSGA2-EVC obtains a performance that is significantly better in mean than those obtained with E+A in BreastC and Newthyroid, with MPENSGA2-E in Newthyroid and with MPANN in Newthyroid and Ionos, and similar results in the rest of datasets.

The computational complexity of MPENSGA2 is greater than other algorithms that are not based on evolutionary metaheuristics, although this disadvantage is offset by great robustness of the method. If we compare the complexity with other EAs with metaheuristics, in terms of number of evaluations and computational time, it is comparable or even less, since *IRprop+* only used three times throughout evolution.

Therefore, we can conclude that the methodology used for obtaining MLP models based on the Pareto-front are well suited to basically improve Accuracy without, in general, decreasing the Sensitivity. These results are consistent with Pareto fronts in E versus VC for training sets and the graphics C versus S for the generalization sets that are shown in Figure 1. In these figures we can see the results obtained for Newthyroid in (VC,E) and (S,C) spaces.

6. Conclusions

In our opinion, the (S,C) perspective and the memetic NSGA2 approach reveal a new point of view for dealing with multi-class classification problems, and provide the opportunity to improve the accuracy of a multiclassifier for a wide range of databases, which is good enough for each and every class.

7. Acknowledgements

This work has been partially subsidized by the TIN 2008-06681-C06-03 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The research done by J.C. Fernández has been financed by the FPI (grant reference BES-2006-12543) Predoctoral Program (Spanish Ministry of Education and Science).

8. References

[1] F. Provost and T. Fawcett, "Robust classification system for imprecise environments," Proc. Proccedings of the Fithteenth National Conference on Artificial Intelligence, pp. 706-713, 1998.

[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[3] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 694-713, 1997.

[4] H. Abbass, "Speeding Up Backpropagation Using Multiobjective Evolutionary Algorithms," Neural Computation, vol. 15, pp. 2705-2726, 2003.

[5] N. Garcia-Pedrajas, C. Hervas-Martinez and D. Ortiz-Boyer, "Cooperative coevolution of artificial neural network ensembles for pattern classification," IEEE Transaction on Evolutionary Computation, vol. 9, no. 3, pp. 271-302, 2005.

[6] X. Yao, "Evolving Artificial Neural Networks," Proc. Proceedings of the IEEE, pp. 1423-1447, 1999.

[7] J.E. Fieldsend and S. Singh, "Pareto evolutionary neural networks," IEEE Transaction on Neural Networks, vol. 16, no. 2, pp. 338–354, 2005.

[8] S. Roth, A. Gepperth and C. Igel, "Multi-objective neural network optimization for visual object detection," Multi-objective Machine Learning, Studies in Computational Intelligence, 2006.

[9] J.E. Smith, "Coevolving Memetic Algorithms: A Review and Progress Report," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 37, no. 1, pp. 6 - 17, 2007.

[10] J.C. Fernández, P.A. Gutiérrez, C. Hervás and F.J. Martínez, "Memetic Pareto Evolutionary Artificial Neural Networks for the determination of growth limits of Listeria Monocytogenes," Proc. Eighth International Conference on Hybrid Intelligent Systems, IEEE, pp. 631-636, 2008.

[11] K. Deb, A. Pratab, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA2," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.

[12] C. Igel and M. Hüsken, "Improving the Rprop Learning Algorithm," Proc. Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), ICSC Academic Press, pp. 115-121, 2000.

[13] C. Hervás and F.J. Martínez-Estudillo, "Logistic regression using covariates obtained by product-unit neural network models," Pattern Recognition, vol. 40, pp. 52-64, 2007.

[14] A. Martinez-Estudillo, F. Martinez-Estudillo, C. Hervas-Martinez and N. Garcia-Pedrajas, "Evolutionary product unit based neural networks for regression," Neural Networks, vol. 19, no. 4, pp. 477-486, 2006.

[15] J. Alcala-Fdez, L. Sánchez, S. García, M.J.d. Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández and F. Herrera, "KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems," Soft Computing, vol. 13, pp. 307-318, 2009.
[16] S. Ventura, C. Romero, A. Zafra, J.A. Delgado and C. Hervás-Martínez, "JCLEC: a JAVA framework for evolutionary computation", Soft Computing, vol. 12, pp. 381-392, 2008.
[17] F.J. Martínez-Estudillo, P.A. Gutiérrez, C. Hervás and J.C. Fernández, "Evolutionary Learning by a Sensitivity-Accuracy Approach for Multi-class Problems " Proc. IEEE World Congress on Computational Intelligence, CEC 08, pp. 1581-1588, 2008.
[18] G.F. Miller, P.M. Todd and S.U. Hedge, "Designing Neural Networks Using Genetic Algorithms," Proc. 3rd International Conference on Genetic Algorithm and Their Applications, pp. 379-384, 1989.
[19] R.G. Miller, Beyond ANOVA, Basics of App. Statistics, Chapman & Hall, 1996.
[20] A.C. Tamhane and D.D. Dunlop, Statistics and Data Analysis., Prentice Hall, 2000.

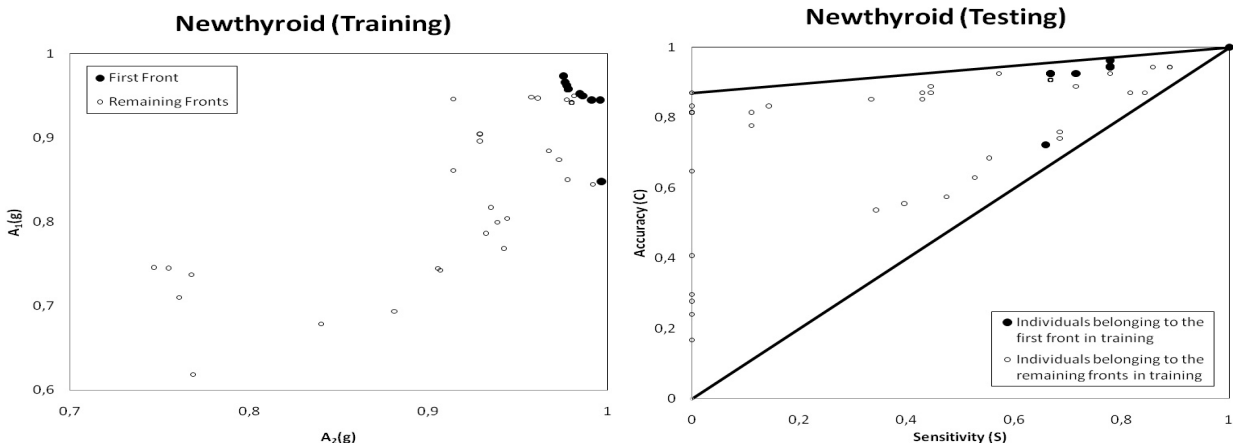

Figure 1. A) Pareto front in training. B) Feasible Region in the (S, C) space for testing

TABLE 2. STATISTICAL RESULTS FOR E+A, MPENSGA2-E, MPENSGA2-EVC AND MPANN

Dataset	METHODOLOGY	*C*(%)	*S*(%)	Dataset	METHODOLOGY	*C*(%)	*S*(%)
ACard	*E + A*	88.02±1.70	85.44±2.94	BreastC	*E + A*	68.92±2.30	25.55±7.02
	MPENSGA2-E	88.07±1.57	86.13±2.73		MPENSGA2-E	69.34±2.30	28.89±9.10
	MPENSGA2-EVC	**88.34±0.95**	**86.36±1.75**		MPENSGA2-EVC	**69.62±2.83**	**30.63±6.81**
	MPANN	87.59±1.18	85.95±1.98		MPANN	66.53±3.07	28.73±14.23
Ionos	*E + A*	92.31±1.63	**85.00±4.05**	Iris	*E + A*	95.72±1.82	7.94±3.87
	MPENSGA2-E	92.65±2.22	82.71±5.36		MPENSGA2-E	**97.18±0.78**	**91.54±2.35**
	MPENSGA2-EVC	**93.26±1.96**	84.17±4.99		MPENSGA2-EVC	96.92±1.24	90.76±3.72
	MPANN	91.10±2.37	79.17±5.99		MPANN	94.35±9.38	83.07±28.16
Newth	*E + A*	95.49±1.73	73.33±9.94	Vote	*E + A*	93.76±1.06	91.78±2.06
	MPENSGA2-E	95.12±2.31	74.81±10.08		MPENSGA2-E	94.74±0.87	**93.42±1.70**
	MPENSGA2-EVC	**95.62±2.09**	**75.34±11.60**		MPENSGA2-EVC	**94.83±0.92**	92.87±2.34
	MPANN	94.87±3.81	72.11±22.29		MPANN	94.43±1.02	92.88±2.19

TABLE 3. P-VALUES OF THE SNEDECOR'S F OR K-W TEST. RANKING OF AVERAGES OF THE TUKEY TEST OR M-W PAIR TEST FOR ACCURACY (C) AND SENSITIVITY (S) FOR MPENSGA2-EVC, (EVC), MPNSGA2-E (E) , ENTROPY PLUS AREA ALGORITHMS, E+A, AND MPANN (MP).

	ACard	BreastC	Ionos	Iris	Newth	Vote
			C			
F or K-W test	0.213	0.000 (*)	0.001 (*)	0.889 KW	0.000 (*)KW	0.000 (*)
Tukey or M-W Ranking of averages	$\mu_{EVC}\geq\mu_E\geq\mu_{E+A}\geq\mu_{MP}$	$\mu_{EVC}\geq\mu_E\geq\mu_{E+A}>\mu_{MP}$	$\mu_{EVC}\geq\mu_E\geq\mu_{E+A}\geq\mu_{MP}$; $\mu_E>\mu_{MP}$	$\mu_E\geq\mu_{EVC}\geq\mu_{E+A}\geq\mu_{MP}$	$\mu_{EVC}>\mu_E$; $\mu_{EVC}\geq\mu_{E+A}$; $\mu_{EVC}>\mu_{MP}$	$\mu_{EVC}\geq\mu_E\geq\mu_{MP}>\mu_{E+A}$
			S			
F or K-W test	0.495	0.244	0.000 (*)	0.832 KW	0.002 (*)KW	0.025 (*)
Tukey or M-W Ranking of averages	$\mu_{EVC}\geq\mu_E\geq\mu_{MP}\geq\mu_{E+A}$	$\mu_{EVC}\geq\mu_E\geq\mu_{MP}\geq\mu_{E+A}$; $\mu_{EVC}>\mu_{E+A}$	$\mu_{E+A}\geq\mu_{EVC}\geq\mu_E>\mu_{MP}$	$\mu_E\geq\mu_{EVC}\geq\mu_{E+A}\geq\mu_{MP}$	$\mu_{EVC}>\mu_E$; $\mu_{EVC}>\mu_{E+A}$; $\mu_{EVC}>\mu_{MP}$	$\mu_E\geq\mu_{MP}\geq\mu_{EVC}\geq\mu_{E+A}$; $\mu_E>\mu_{E+A}$