Contents lists available at ScienceDirect

Expert Systems with Applications

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

A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests

Julián Luengo, Salvador García *, Francisco Herrera

Department of Computer Science and Artificial Intelligence, University of Granada, C/Daniel Saucedo Aranda S/N, 18071 Granada, Spain

ARTICLE INFO

Keywords:
Neural networks
Statistical tests
Radial basis function networks
Multi-layer perceptron
Support vector machines
Learning vector quantization
Classification

ABSTRACT

In this paper, we focus on the experimental analysis on the performance in artificial neural networks with the use of statistical tests on the classification task. Particularly, we have studied whether the sample of results from multiple trials obtained by conventional artificial neural networks and support vector machines checks the necessary conditions for being analyzed through parametrical tests. The study is conducted by considering three possibilities on classification experiments: random variation in the selection of test data, the selection of training data and internal randomness in the learning algorithm.

The results obtained state that the fulfillment of these conditions are problem-dependent and indefinite, which justifies the need of using non-parametric statistics in the experimental analysis.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

The report of results in artificial neural networks (ANNs) and support vector machines (SVMs) is usually given by means of simple and well-known statistical measures, such as mean and standard deviations. However, when comparing them on different cases of a certain problem, an aggregated analysis by using only these measures lacks of rigorousness and may hidden some information in the results obtained. One solution to this problem is offered by statistical validation of obtained results (Demšar, 2006).

Due to the increasing number of real-world applications and frameworks for machine learning (ML), the development or modifications of new algorithms is a relative easy task. In spite of this fact, every development made must offer a certain advantage with respect to previous proposals in the area of research of interest. Establishing a good procedure of comparing groups of methods by using empirical results is a necessary matter on a specify study.

When using ANNs or SVMs in ML for the classification problem (Qiu, Tao, Tan, & Wu, 2007; Wu, Huang, & Meng, 2008), the main intention is to propose a new algorithm which improves a certain aspect over existing algorithm(s), such as effectiveness, efficiency or interpretability. In classification, a number of data sets is selected for testing, the algorithms are run over them and the quality of the resulting models is evaluated by means of an appropriate measure (commonly, the accuracy in test data). A final step, more and more demanded in the scientific community and the topic that we want to illustrate, is the use of appropriate statistical tests depending on the properties of the sample of data obtained. Statistics allows us to determine whether the results obtained are significantly different in the algorithms compared and whether the conclusions remarked are supported by the experimentation carried out.

Indeed, a low proportion of publications uses statistical techniques to comparing the obtained results. Nevertheless, their presence is growing notoriously in terms of parametric tests (Sheskin, 2003). When we found statistical studies, they are based on the mean and variance, by using tests such as paired *t*-test or ANOVA (Alpaydin, 1999; Castillo-Valdivieso, Merelo, Prieto, Rojas, & Romero, 2002; Gao, Madden, Chambers, & Lyons, 2005; Kim, 2008; Lam, 2004; Li, Shiue, & Huang, 2006; Zekic-Susac & Horvat, 2005). In few cases, non-parametric tests are used for comparing ANNs (Groves et al., 1999; Pizarro, Guerrero, & Galindo, 2002) or immune networks (García-Pedrajas & Fyfe, 2007). A study on the distribution of performance in ANNs can be found in Lawrence, Back, Tsoi, and Giles (1997), in which the authors give some guidelines on the presentation of results and convergence of the learning algorithms of ANNs, supposing that the sample of results does not follow a normal distribution.

In this paper, we will focus on the use of statistical techniques for the analysis of ANNs and SVMs in classification tasks in order to establish comparisons among the algorithms, instead of providing a piece of advice on presenting the results, studying the appropriate use of parametric and non-parametric tests (Sheskin, 2003; Zar, 1999). In fact, we analyze the required conditions which allow the usage of parametric tests for comparing ANNs and SVMs in classification, depending on three main factors: the random variation in the selection of the test data, the selection of the training data and the internal randomness in the learning algorithm. In

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: julianlm@decsai.ugr.es (J. Luengo), salvagl@decsai.ugr.es (S. García), herrera@decsai.ugr.es (F. Herrera).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.11.041

general, we will show that the sample of results obtained is not appropriate for parametric statistical analysis and, for this reason, we will illustrate a case study on the multiple comparison with non-parametric tests.

To achieve the mentioned goal, we will use some well-known models of ANNs and SVMs applied to classification of data sets (Rojas & Feldman, 1996; Yingwei, Sundararajan, & Saratchandran, 1997). Thus, the paper is organized as follows. In Section 2 we describe the ANN models used in the study together with the SVM methods and we explain the experimental framework. Section 3 explores the needed conditions in order to correctly apply the parametric test to analyze the obtained results. The presentation of the case study involving some non-parametric tests is given in Section 4. Finally, in Section 5, we point out the conclusions of the paper.

2. Classification algorithms and experimentation framework

In this section, we will briefly describe the algorithms used in this study (see Subsection 2.1). In Subsection 2.2, the experimental framework is explained, including the data sets and types of validation chosen and the results obtained by the algorithms are presented through classic statistics metrics, such as mean and standard deviation.

2.1. Artificial neural networks and support vector machines

The experimentation in this paper is conducted by using the following models of ANNs:

- Multi-layer perceptron (MLP) with back-propagation (Rojas & Feldman, 1996): it is a classical model of ANN whose weights are adjusted trough a back-propagation algorithm. This type of network consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in a layer has directed connections to the neurons of the subsequent layer. In many applications, the units of these networks apply a sigmoid function as an activation function. The output values are compared with the correct answer to compute the value of some predefined error function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state in which the error of the computations is small. Typical problems of the back-propagation algorithm are the speed of convergence and the possibility of ending up in a local minimum of the error function. In our study, we have used the configuration of a hidden layer of neurons with 25 perceptrons.
- Radial basis function network (RBFN) (Rojas & Feldman, 1996): it is well-suited for function approximation and pattern recognition due to its simple topological structure and its ability to reveal how learning proceeds in an explicit manner. A radial basis function (RBF) is a function that has built into a distance criterion with respect to a centre. Different basis functions like thin-plate spline functions, multiquadratic functions, inverse multiquadratic functions and gaussian functions have been studied for the hidden layer neurons, but normally the selected one is the gaussian function. Compared with other types of ANNs, such as feed-forward networks, the RBFN requires less computation time for learning and also has a more compact topology. RBFs have been applied in the area of ANNs where they may be used as a replacement for the sigmodial hidden layer transfer characteristic in multi-layer perceptrons. The original RBF method has been traditionally used for strict multivariate function interpolation (Powell, 1987) and for this fact, it requires as many RBF neurons as data points. Broomhead and Lowe (1988) removed this strict interpolation restriction and provided a neural network architecture where the number of RBF neurons can be far less than the data points. A RBFN mainly consists of two layers, one hidden layer and one output layer. The number of hidden layer neurons is configurable and fixable by the user a priori. In our study, we have fixed the number of neurons at 50.
- RBFN Decremental (Yingwei et al., 1997): in the classical approach described above, the number of hidden units is fixed a priori based on the properties of input data. A significant contribution that overcomes these drawbacks was made through the development of an algorithm that adds hidden units to the network based on the novelty of the new data. One drawback of this approach is that once a hidden unit is created, it can never be removed. The authors have proposed an algorithm that adopts the basic idea of growing, and augments it with a pruning strategy. The pruning strategy removes those hidden neurons which consistently make little contribution to the network output. Pruning becomes imperative for the identification of nonlinear systems with changing dynamics, because failing to prune the network in such cases will result in numerous inactive hidden neurons, being present as the dynamics which caused that their creation initially becomes nonexistent. If inactive hidden units can be detected and removed while learning proceeds, a more well-suited network topology can be constructed. Also, when the neural networks are employed for control, the problem of over-parametrization should be avoided. In our study, we provide this model with 20 initial neurons, a percentage of 0.1 under the average of the weights used to decide whether a neuron must be removed or not, and a learning factor $\alpha = 0.3$ of the least mean square (LMS) algorithm for adjusting weights.
- RBFN incremental: this approach builds a RBFN composed of one hidden layer and one output layer. This topography is similar to non-incremental RBFN's one, but we do not know the number of neurons of the hidden layer. This idea is similar to RBFN Decremental, but in this model we will not set any limit to the hidden layer's neurons number.The network begins with no hidden units, and while observations are received, the network grows by using some of them as new RBFs. Two criteria must be met for an observation (x, y) to add a new hidden unit to the network:
  - The euclidean distance between x and its closest RBF must be greater than δ.
  - The error between the output of the net and y must be greater than δ.

  When a new hidden neuron is added to the network, the centre is set to x, the radius is set to the distance between x and its closest RBF with an overlap, and the weight is set to the error between the output of the net and y. If any of these two criteria are not met. the LMS algorithm (used to adjust the weights) is applied to a random RBF. The growing process of the net stops when no RBF unit is added and all instances from the data set are processed.This method tries to find the correct number of neurons for a given data set, without an initial limitation like as in the RBFN model (which has its neurons number fixed) and the RBFN Decremental model (which also has a maximum number of neurons fixed a priori). However, if δ is too low, we can find that our network overfits the training data. The model that we have used is set with $\alpha = 0.3$ (see description of this parameter above), $\delta = 0.5$, which is the minimum distance allowed to introduce a new RBF, and $\epsilon = 0.1$, which is the minimum error allowed to introduce a new RBF. If the latter both values are lower, the probability to add a RBF grows.

- C-SVM (also C-support vector classification or C-SVC) (Cortes & Vapnik, 1995; Fan, Chen, & Lin, 2005): SVM (support vector machine) is a useful technique for data classification. Given training vectors $x_i \in R^n$, $i = 1, \ldots, l$ in two cases, and a vector $y \in R^l$ such that $y_i \in 1, -1$, C-SVC solves the following primal problem:

  $$\min_{w,b,\xi} \frac{1}{2} w^t w + C \sum_{i=1}^{l} \xi_i$$

  subject to

  $$y_i(w^t \phi(x_i) + b) \geqslant 1 - \xi_i,$$
  $$\xi_i \geqslant 0, i = 1, \ldots, l.$$

  Its dual is

  $$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

  subject to

  $$y^T \alpha = 0$$
  $$0 \leqslant \alpha_i \leqslant C, i = 1, .., l,$$

  where e is the vector of all ones, $C > 0$ is the upper bound, Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i\, y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel. Here, training vectors x_i are mapped to a higher (maybe infinite) dimensional space by the function ϕ. The decision function is

  $$\text{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b \right).$$

  For multi-class classification, the "one-against-one" approach is used, in which $k(k - 1)/2$ classifiers are constructed and each one trains data from two different classes.In classification, we use a voting strategy: each binary classification is considered to be a voting where votes can be cast for all data points x – in the end point is designated to be in a class with maximum number of votes. The parameters used by the method comprises a RBF Kernel, $C = 1000$, $\epsilon = 0.001$, $degree = 10$, $\gamma = 1$, $coef_0 = 1$ and no shrinking.
- ν-SVM (NU-SVM) (Fan et al., 2005; Schölkopf, Smola, Williamson, & Bartlett, 2000): the ν-support vector classification uses a new parameter ν which controls the number of support vectors and training errors. The parameter $\nu \in (0,1]$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. Given training vectors $x_i \in R^n$, $i = 1, \ldots, l$ in two cases, and a vector $y \in R^l$ such that $y_i \in 1, -1$, the primal form considered is

  $$\min_{w,b,\xi,\rho} \frac{1}{2} w^t w + \nu\rho \sum_{i=1}^{l} \xi_i$$

  subject to

  $$y_i(w^t \phi(x_i) + b) \geqslant \rho - \xi_i,$$
  $$\xi_i \geqslant 0, i = 1, \ldots, l; \rho \geqslant 0.$$

  The dual is

  $$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha$$

  subject to

  $$y^T \alpha = 0; e^T \alpha \geqslant \nu$$
  $$0 \leqslant \alpha_i \leqslant 1/l, i = 1, .., l,$$

  where $Q_{ij} \equiv y_i y_j K(x_i, x_j)$. The decision function is

  $$\text{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b \right).$$

Table 1
Data Sets used in the experimentation.

Data set	# Instances	# Attributes	# Classes
Breast	682	10	2
Cleveland	303	13	5
Crx	689	16	2
Glass	214	9	7
Iris	150	4	3
Pima	768	8	2
Wine	178	13	3
Wisconsin	699	10	2
Bupa	345	7	2
Lymphography	148	18	4
Monks	432	6	2
Page-blocks	5476	10	5
Pen-based	10992	16	10
Ringnorm	7400	20	2
Satimage	6435	36	7
Splice	3190	60	3

  It has been shown that $e^T\alpha \geqslant \nu$ can be replaced by $e^T \alpha = \nu$, so we can solve a scaled version of the dual primal form. In the end, the two margins obtained are the same as those of C-SVC. The parameters used by the method comprises a RBF Kernel, $\nu = 0.01$, $\rho = 1$, $\epsilon = 0.001$, $degree = 10$, $\gamma = 1$, $coef_0 = 1$ and no shrinking.
- Learning vector quantization (LVQ) (Bezdek & Kuncheva, 2001): The LVQ family comprises a large spectrum of competitive learning schemes. One of the basic designs that can be used for prototype generation is the LVQ1 algorithm. An initial set of labeled prototypes is picked by first specifying $n_P \geqslant c$. Then n_p elements are randomly selected from X (the data set) to be the initial prototypes, so each class is represented by at least one prototype. LVQ1 has three additional parameters specified by the user: the learning rate $\alpha_k \in (0,1)$, a constant $\eta \in (0,1)$, and the terminal number of iterations T. The standard competitive learning update equation is then used to alter the prototype set. If the closed prototype for input x_k is the vector $v_{\text{i,old}}$,

  $$v_{\text{i,new}} = v_{\text{i,old}} + \alpha_k(x_k - v_{\text{i,old}}) \quad \text{when } l(v_{\text{i,old}} = l(x_k))$$

  or

  $$v_{\text{i,new}} = v_{\text{i,old}} + \alpha_k(x_k - v_{\text{i,old}}) \quad \text{when } l(v_{i,old} \neq l(x_k)).$$

  First equation rewards the winning prototype for getting the correct label by letting it migrate towards the input vector, while second equation punishes the winning prototype for not labeling the current input correctly by repelling it away from the input vector. In our experiments, the learning rate was updated after each presentation of a new vector using the formula $\alpha_{k+1} = \eta\alpha_k$, $k = 1, \ldots, n - 1$; and was restored to the initial, user specified value of α_1 at the end of each pass through X. Before each new pass through LVQ1, X is randomly permuted to avoid dependence of the extracted prototypes on the order of inputs. LVQ1 terminates when either (i) there are no misclassifications in a whole pass through X (and hence, the extracted prototypes are a consistent reference set); or (ii) the prespecified terminal number of iterations is reached. In our experimentation, we have used a maximum number of iterations of 100 over X, $n_p = 20$, $\alpha = 0.3$ and $\eta = 0.8$.

2.2. Experimentation framework

We have selected a group of data sets taken from the UCI repository (Newman, Hettich, Blake, & Merz, 1998) and DELVE[1] project.

[1] http://www.cs.toronto.edu/delve/.

Table 2
Results obtained for the model used in HOV.

Algorithm	MLP		RBFN		RBFN Decremental		RBFN Inc.		LVQ		*C*-SVM		*NU*-SVM	
Data set	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Breast	68.51	3.05	70.10	3.27	63.27	6.11	62.25	3.11	66.09	4.69	65.82	0.00	63.29	0.00
Bupa	61.65	4.59	62.14	3.61	59.98	4.22	66.17	3.25	51.99	4.40	77.67	0.00	46.60	0.00
Cleveland	51.93	3.73	31.86	5.70	34.70	6.76	34.98	6.29	47.07	4.99	57.95	0.00	57.95	0.00
Crx	85.41	1.10	71.90	2.49	46.05	8.59	65.57	3.25	78.71	4.12	83.33	0.00	81.77	0.00
Glass	52.18	9.23	26.25	6.57	43.29	14.67	59.96	7.46	45.77	5.97	76.79	0.00	60.71	0.00
Iris	82.05	6.53	94.23	4.63	90.65	11.57	95.72	2.46	91.25	2.98	97.67	0.00	100.00	0.00
Lymphography	37.14	5.62	32.43	0.00	37.73	5.96	31.03	4.89	34.33	4.05	40.54	0.00	40.54	0.00
Monks	77.64	4.43	81.06	3.69	97.48	2.02	99.98	0.11	67.22	3.71	99.19	0.00	99.19	0.00
Page-blocks	88.99	2.12	89.52	1.89	86.09	8.43	89.86	2.14	86.93	3.58	96.16	0.00	84.69	0.00
Penbased	51.64	4.67	44.71	3.37	27.02	5.84	86.16	1.87	69.99	4.22	99.58	0.00	99.67	0.00
Pima	70.03	2.09	71.32	3.68	69.04	3.61	61.75	2.93	66.35	3.63	74.56	0.00	32.02	0.00
Ringnorm	76.27	1.07	95.33	0.86	95.20	1.50	97.14	0.13	58.57	2.86	96.44	0.00	96.66	0.00
Satimage	62.35	10.37	61.56	3.07	45.12	7.38	77.00	2.06	71.03	5.53	90.97	0.00	58.57	0.00
Splice	47.99	24.60	58.35	3.87	32.52	17.16	84.33	0.38	55.85	3.71	60.21	0.00	60.31	0.00
Wine	95.50	2.08	69.12	3.46	71.65	3.90	75.31	5.46	92.85	3.11	94.23	0.00	94.23	0.00
Wisconsin	95.51	0.37	96.35	0.32	96.34	0.75	96.13	0.64	95.94	1.61	95.00	0.00	95.00	0.00

Table 3
Results obtained for the models used in 10FCV.

Algorithm	MLP		RBFN		RBFN Decremental		RBFN Inc.		LVQ		*C*-SVM		*NU*-SVM	
Data set	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Breast	68.81	8.80	72.08	4.57	69.22	11.58	65.08	9.20	63.61	5.90	66.83	4.83	64.27	7.11
Bupa	61.56	8.37	63.55	6.87	63.01	8.88	62.09	7.57	54.33	3.45	70.09	8.81	42.12	6.14
Cleveland	53.51	7.41	30.34	10.19	34.86	11.66	34.07	8.80	49.13	4.20	54.59	4.24	53.85	3.77
Crx	83.86	10.65	67.41	7.73	46.83	5.87	63.59	5.61	79.08	3.38	80.39	4.91	74.15	7.64
Glass	50.19	10.14	27.08	7.46	40.37	11.82	55.04	11.17	49.47	4.87	71.61	10.48	56.77	9.78
Iris	77.87	9.47	92.13	7.09	91.20	6.38	94.53	4.40	91.20	4.32	94.67	4.04	84.00	21.34
Lymphography	34.26	11.83	31.10	1.67	36.74	9.65	34.14	12.69	35.90	4.40	42.65	6.64	42.56	6.00
Monks	75.05	13.25	88.33	6.91	97.40	2.90	100.00	0.00	66.13	3.60	99.77	0.69	99.77	0.69
Page-blocks	89.17	2.64	88.32	3.27	88.34	3.59	90.07	3.66	84.59	3.25	96.67	0.62	87.83	6.21
Penbased	50.62	5.95	45.42	3.76	30.63	5.04	86.93	4.24	69.36	4.88	99.55	0.24	99.60	0.18
Pima	73.06	4.80	72.25	5.60	72.01	4.23	65.68	5.97	64.79	5.50	73.71	5.53	42.31	14.02
Ringnorm	75.63	1.57	96.00	1.11	95.41	2.74	97.36	0.85	59.04	3.09	96.81	0.49	96.20	0.54
Satimage	62.09	10.12	64.10	2.89	43.49	9.42	77.04	2.79	70.91	6.39	90.64	1.27	72.79	3.53
Splice	50.33	24.63	61.39	2.85	27.87	10.88	83.29	1.45	55.37	3.07	61.79	1.55	61.79	1.55
Wine	97.42	3.42	66.09	7.57	65.52	9.11	73.99	7.72	89.81	4.63	96.67	5.14	96.67	5.14
Wisconsin	96.48	3.23	97.03	2.54	96.92	2.45	96.31	2.28	94.84	2.30	95.61	2.39	95.61	2.39

Table 4
Results obtained for the models used in 5 × 2CV.

Algorithm	MLP		RBFN		RBFN Decremental		RBFN Inc.		LVQ		*C*-SVM		*NU*-SVM	
Data Set	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Breast	69.81	2.36	71.30	1.26	69.54	5.98	65.12	4.82	64.22	6.17	68.18	1.34	54.86	18.23
Bupa	50.23	4.92	30.40	6.33	38.17	7.18	35.07	4.81	55.68	4.27	69.49	2.10	41.99	3.41
Cleveland	50.23	4.92	30.40	6.33	38.17	7.18	35.07	4.81	50.65	6.33	51.41	3.20	50.87	2.95
Crx	85.49	1.37	67.22	2.56	48.77	7.73	63.31	2.85	77.37	5.16	78.07	1.25	77.52	2.71
Glass	49.76	4.10	24.93	6.43	37.74	11.21	50.11	5.37	52.34	5.19	67.10	1.62	49.91	7.35
Iris	78.27	7.45	90.59	4.21	83.36	8.39	93.09	2.98	91.04	3.42	94.13	1.37	95.20	3.02
Lymphography	78.27	7.45	90.59	4.21	83.36	8.39	93.09	2.98	37.05	5.68	39.19	4.88	38.65	4.62
Monks	71.99	1.85	70.83	1.85	71.17	1.70	65.66	4.33	66.96	4.61	95.95	0.99	95.29	1.25
Page-blocks	88.87	2.52	89.25	2.33	82.82	18.27	89.23	2.95	85.12	4.98	96.63	0.18	79.34	10.76
Penbased	50.24	5.40	43.79	3.86	25.95	5.25	84.53	2.01	70.06	4.52	99.50	0.01	99.55	0.03
Pima	71.99	1.85	70.83	1.85	71.17	1.70	65.66	4.33	66.59	3.68	70.68	2.15	44.06	10.66
Ringnorm	95.84	1.08	96.92	0.62	96.75	0.65	96.21	0.62	58.26	3.52	96.21	0.43	95.92	0.46
Satimage	62.43	9.53	63.64	2.61	44.57	10.20	75.91	1.64	72.59	5.02	89.75	0.39	72.71	3.82
Splice	58.92	22.58	48.95	5.89	45.23	21.42	80.92	0.48	54.64	3.63	59.68	1.00	59.68	1.00
Wine	49.76	4.10	24.93	6.43	37.74	11.21	50.11	5.37	90.54	3.88	97.53	1.32	97.75	1.02
Wisconsin	95.84	1.08	96.92	0.62	96.75	0.65	96.21	0.62	94.69	2.18	94.79	0.53	94.91	0.62

Altogether, we have used 16 data sets to carry out the study. In Table 1, we summarize the properties of these data sets. The 8 first data sets will be used to perform the study of the initial conditions (in Section 3) in order to not enlarge the size of the tables of results. The complete set of data will be used in the non-parametric statistical analysis (see Section 4).

Making use of these data sets, different types of validation have been carried out depending on the possibility that we want to study in classification problems:

- The internal randomness of the learning algorithms only supposes to control the initial random seeds on an unique partition train-test. In order to perform this study, we use the hold-out validation (HOV), which consists of partitioning the data set into 2 subsets, one used for training and the other for test. We have used hold-out at 70–30%, in order to give more examples for learning the model.
- The random variation in the selection of the test data, which can be controlled by using different subsets of examples with no overlapping among themselves. The 10-fold cross-validation (10FCV) can be used to check this possibility, due to the each test subset contains examples which cannot belong to other subset.
- The selection of the training set, in the same way as above, controls the examples that belong to the training set. The best way to check this property and to not be against a good convergence of learning algorithms, is to use the 5 × 2 cross-validation (5 × 2CV) (Dietterich, 1998). This validation is conducted through 5 repetitions of a 2FCV, which obtains 5 train and 5 test subsets at 50%. In the training subsets, the overlapping of examples exists, but it is less notably than in the 10FCV.

In any case, we have run a number of trials enough to achieve a sample of results with a size of 50 elements. Thus, we have repeated the experiments for 10FCV and 5 × 2CV 5 times, and 50 times for HOV.

Tables 1–4 show the average results and standard deviations of test accuracy obtained for all the models used in this study.

3. Study on the initial conditions for parametric tests using artificial neural networks

In this section, we will analyze the conditions that must be satisfied in the use of parametric tests by using the data sets and methods previously defined. First, we introduce the three conditions. Then, we present the analysis of the normality and heteroscedasticity tests, and finally we show some cases study of the normality property.

3.1. Conditions for the use of parametric tests

In Sheskin (2003), the distinction done between parametric and non-parametric tests is based on the level of measure represented by the data that will be analyzed. In this way, a parametric test uses data with real values belonging to a range.

The latter does not involve that when we always dispose of this type of data, we should use a parametric test. It is possible that one or more initial assumptions for the use of parametric tests are not fulfilled, making that a statistical analysis loses credibility.

In order to use the parametric tests, is necessary to check the following conditions (Sheskin, 2003; Zar, 1999):

- *Independence*: in statistics, two events are independent when the fact that one occurs does not modify the probability of the other one occurring.
- *Normality*: an observation is normal when its behaviour follows a normal or Gaussian distribution with a certain value of mean μ and variance σ. A normality test applied over a sample can indicate the presence or absence of this condition in the observed data. We will use three normality tests:
  - *Kolmogorov–Smirnov*: it compares the accumulated distribution of observed data with the accumulated distribution expected for a Gaussian distribution, obtaining the *p*-value based on both discrepancies. Therefore, it is a quality of fit procedure that can be used to test the hypothesis of normality in the population distribution. However, this method performs poorly because it possess very low power.
  - *Shapiro–Wilk*: it analyzes the observed data for computing the level of symmetry and kurtosis (shape of the curve) in order to compute the difference with respect to a Gaussian distribution afterwards, obtaining the *p*-value from the sum of the squares of these discrepancies. The power of this test has been shown to be excellent. However, the performance of this test is adversely affected in the common situation where there is tied data.
  - *D'Agostino–Pearson*: it first computes the skewness and kurtosis to quantify how far from Gaussian the distribution is in terms of asymmetry and shape. It then calculates how far each of these values differs from the values expected with a Gaussian distribution, and computed a single *p*-value form the sum of these discrepancies. The performance of this test is not as good as that of Shapiro–Wilk's procedure, but it is not affected by tied data.
- *Heteroscedasticity*: this property indicates the existence of a violation of the hypothesis of equality of variances. Levene's test is used for checking if k samples present or not this homogeneity of variances (homoscedasticity). When observed data does not fulfill the normality condition, it is more reliable the result of using this test than Bartlett's test (Zar, 1999), which is another test that checks the same property.

With respect to the independence condition, Demšar (2006) suggests that independency is not truly verified in 10FCV and 5 × 2CV (a portion of samples is used either for training and testing in different partitions). Hold-out partitions can be safely take as independent, since training and tests partitions do not overlap. Furthermore, the independence of the events in terms of getting results is obvious, given that they are independent runs of the algorithm with randomly generated initial seeds. In the following, we show a normality analysis by using Kolmogorov–Smirnov's, Shapiro–Wilk's and D'Agostino–Pearson's tests, together with a heteroscedasticity analysis by using Levene's test.

3.2. Normality test over the group of data sets and algorithms

We apply the normality test of Kolmogorov–Smirnov by considering a level of confidence of $\alpha = 0.05$ (we employ SPSS statistical software package). Tables 5, 8 and 11 show the results in HOV, 10FCV and 5 × 2CV, respectively, where the symbol '*' indicates that the normality is not satisfied and the value in brackets is the *p*-value needed for rejecting the normality hypothesis. Tables 6, 9 and 12 show the results by applying the test of normality of Shapiro–Wilk. Finally, Tables 7, 10 and 13 show the results of the application of D'Agostino–Pearson's test. As we have indicated above, this study is performed by using the first group of data sets in Table 1. The algorithms *C*-SVM and *NU*-SVM are not included in this study due to the fact that they are non-probabilistic methods, thus the sample of results obtained do not depend on their randomness, but on the partitions used.

As we can observe in the run of the three tests, we can declare that the conditions needed for the application of parametric tests are not fulfilled in some cases. The normality condition is not always satisfied although the size of the sample of results would be enough (50 in this case). A main factor that influences this condition seems to be the nature of the problem, since there exist some problems in which it is never satisfied, such as in the iris

Table 5
Test of normality of Kolmogorov–Smirnov for HOV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	*(.00)	*(.01)	(.20)	(.10)	*(.00)	*(.00)	*(.00)	*(.00)
RBFN	*(.00)	(.18)	(.07)	*(.00)	*(.00)	(.20)	*(.01)	*(.00)
RBFN Decremental	*(.00)	(.20)	*(.00)	(.20)	*(.00)	(.16)	*(.00)	*(.00)
RBFN Inc.	*(.04)	(.20)	(.20)	*(.01)	*(.00)	*(.03)	(.20)	*(.00)
LVQ	(.11)	(.20)	*(.04)	*(.00)	*(.04)	*(.01)	(.07)	*(.00)

Table 6
Test of Normality of Shapiro–Wilk for HOV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	*(.01)	(.29)	(.05)	(.15)	*(.00)	(.14)	*(.00)	*(.00)
RBFN	*(.00)	(.25)	(.11)	*(.00)	*(.00)	(.68)	(.05)	*(.00)
RBFN Decremental	*(.00)	(.85)	*(.00)	(.06)	*(.00)	(.15)	*(.01)	*(.00)
RBFN Inc.	*(.03)	(.06)	(.45)	(.09)	*(.00)	(.10)	(.90)	*(.02)
LVQ	(.13)	(.43)	(.05)	*(.00)	(.05)	*(.00)	(.07)	*(.00)

Table 7
Test of Normality of D'Agostino–Pearson for HOV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.17)	(.65)	(.06)	(.14)	*(.00)	(.35)	*(.01)	(.12)
RBFN	*(.00)	(.18)	(.38)	*(.02)	*(.00)	(.59)	*(.01)	(.26)
RBFN Decremental	*(.00)	(.88)	*(.00)	(.10)	*(.00)	(.43)	(.40)	*(.00)
RBFN Inc.	(.24)	(.06)	(.50)	(.09)	(.09)	(.10)	(.94)	(.98)
LVQ	(.31)	(.59)	(.11)	*(.00)	(.21)	*(.00)	(.05)	*(.00)

Table 8
Test of normality of Kolmogorov–Smirnov for 10FCV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.20)	(.17)	*(.00)	*(.03)	*(.00)	*(.01)	*(.00)	*(.00)
RBFN	*(.02)	*(.01)	(.20)	(.20)	*(.00)	(.20)	*(.00)	*(.00)
RBFN Decremental	(.20)	(.20)	*(.00)	(.20)	*(.00)	(.18)	*(.00)	*(.00)
RBFN Inc.	(.10)	(.20)	(.20)	(.20)	*(.00)	(.06)	*(.03)	*(.00)
LVQ	(.20)	(.08)	(.20)	(.20)	(.20)	(.20)	*(.00)	*(.00)

Table 9
Test of Normality of Shapiro–Wilk for 10FCV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.38)	(.71)	*(.00)	(.21)	*(.00)	*(.00)	*(.00)	*(.00)
RBFN	(.09)	(.05)	(.56)	(.57)	*(.00)	(.19)	*(.00)	*(.00)
RBFN Decremental	(.06)	(.48)	*(.00)	(.50)	*(.00)	(.26)	*(.00)	*(.00)
RBFN Inc.	(.20)	(.13)	(.51)	(.57)	*(.00)	*(.01)	(.06)	*(.01)
LVQ	(.73)	*(.00)	(.08)	(.63)	*(.03)	*(.02)	*(.00)	*(.00)

Table 10
Test of Normality of D'Agostino–Pearson for 10FCV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.21)	(.70)	*(.00)	(.51)	*(.03)	(.06)	*(.03)	*(.00)
RBFN	(.63)	(.20)	(.61)	(.60)	*(.00)	(.27)	*(.00)	*(.03)
RBFN Decremental	(.06)	(.56)	*(.00)	(.63)	(.15)	(.10)	*(.00)	*(.39)
RBFN Inc.	(.36)	(.65)	(.90)	(.11)	(.38)	*(.04)	(.53)	(.07)
LVQ	(.78)	*(.00)	*(.02)	(1.00)	(.18)	(.23)	*(.00)	*(.00)

Table 11
Test of Normality of Kolmogorov–Smirnov for 5 × 2CV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.18)	(.20)	(.20)	*(.04)	(.20)	(.20)	*(.04)	(.20)
RBFN	(.20)	(.20)	(.09)	*(.00)	(.20)	(.20)	*(.00)	*(.01)
RBFN Decremental	*(.00)	(.05)	*(.00)	*(.00)	*(.00)	(.20)	*(.00)	*(.01)
RBFN Inc.	*(.01)	(.20)	(.20)	(.20)	*(.01)	(.20)	(.20)	*(.04)
LVQ	(.20)	*(.04)	(.05)	(.07)	*(.03)	(.05)	*(.00)	(.07)

Table 12
Test of Normality of Shapiro–Wilk for 5 × 2CV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.59)	(.72)	(.12)	(.22)	(.19)	(.20)	(.22)	(.08)
RBFN	(.08)	(.59)	*(.04)	*(.00)	(.07)	(.14)	*(.01)	*(.00)
RBFN Decremental	*(.00)	*(.03)	*(.00)	*(.01)	*(.00)	(.73)	*(.01)	(.21)
RBFN Inc.	*(.02)	(.43)	(.35)	(.86)	(.13)	(.16)	(.87)	(.42)
LVQ	(.27)	(.11)	*(.03)	(.12)	*(.05)	(.53)	*(.01)	*(.00)

Table 13
Test of Normality of D'Agostino–Pearson for 5 × 2CV.

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
MLP	(.92)	(.60)	*(.03)	(.53)	(.11)	(.46)	(.53)	(.14)
RBFN	(.90)	(.63)	*(.22)	*(.02)	(.03)	(.06)	(.11)	*(.02)
RBFN Decremental	*(.00)	*(.17)	*(.00)	(.11)	*(.00)	(.82)	*(.02)	(.25)
RBFN Inc.	*(.02)	(.34)	(.34)	(.90)	(.56)	(.18)	(.90)	(.66)
LVQ	(.42)	(.09)	(.11)	(.65)	(.30)	(.76)	*(.03)	*(.00)

and wisconsin problems in HOV and 10FCV, and the general trend is not predictable. D'Agostino–Pearson's test is the most suitable test in these situations, where it is frequent that the sample of results would contain some ties. Note that in 5 × 2CV the number of rejections is usually lower than in HOV or 10FCV, so this validation obtains good-fitted sample of results to the Gaussian distribution.

In relation to the heteroscedasticity study, Table 14 shows the results by applying Levene's test, where the symbol '*' indicates that the variances of the distributions of the different algorithms for a certain data set are not homogeneous (we reject the null hypothesis).

The homoscedasticity property is even more difficult to be fulfilled, since the variances associated to each problem also depend on the algorithm's results, that is, the capacity of the algorithms for offering similar results with random seed variations. This fact implies that an analysis of performance of ANN methods performed through parametric statistical treatment could mean erroneous conclusions.

3.3. Case studies of the normality property

In the following, we present a case study done for a given sample of results. From the Figs. 1–4, different examples of graphical representations of histograms and Q–Q graphics are shown. A histogram represents a statistical variable by using bars, so that the area of each bar is proportional to the frequency of the represented values. A Q–Q graphic represents a confrontation between the quartiles from data observed and those from the normal distribution.

In Fig. 1 we observe a typical case of absolute lack of normality. In Fig. 2, the normality condition is not rejected by the D'Agostino–Pearson test, which is the best-suitable test for normality condition. In this case, Shapiro–Wilk test is unable to detect the

Table 14
Test of Heteroscedasticity of Levene (based on means).

	Breast	Cleveland	Crx	Glass	Iris	Pima	Wine	Wisconsin
HOV	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)
10FCV	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)	(.20)	*(.00)	*(.01)
5 × 2CV	*(.00)	*(.01)	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)	*(.00)

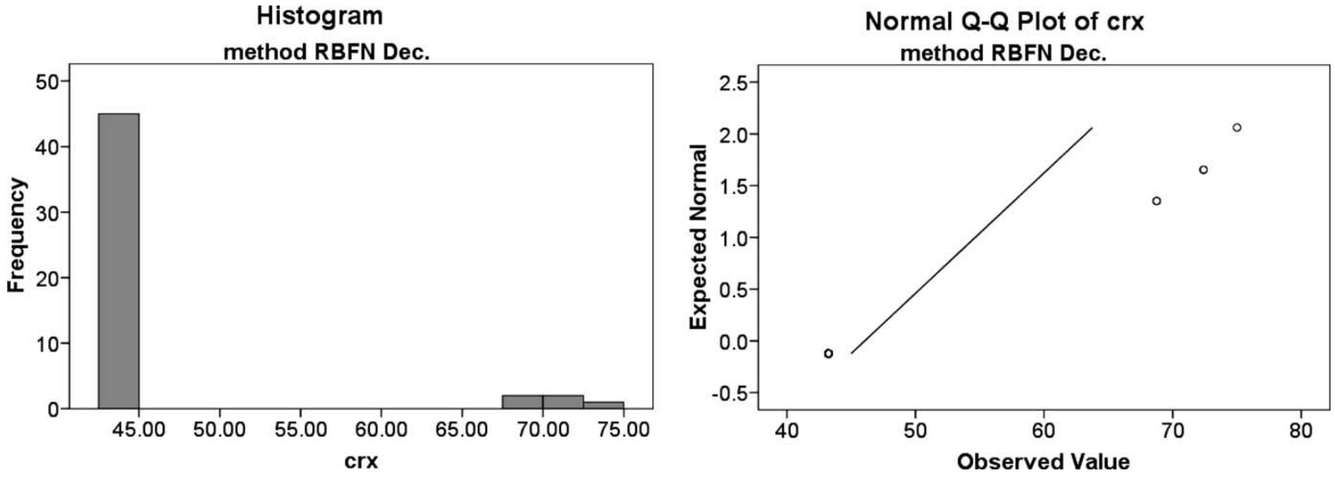

Fig. 1. Results of RBFN Decremental over crx data set in HOV: histogram and Q–Q graphic.

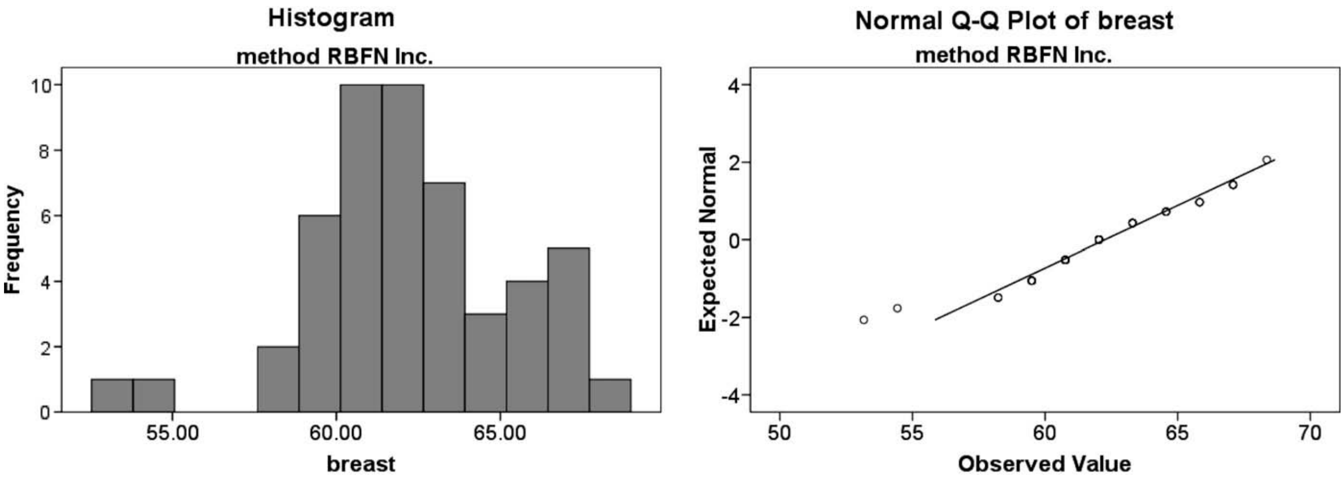

Fig. 2. Results of RBFN incremental over breast data set in HOV: histogram and Q–Q graphic.

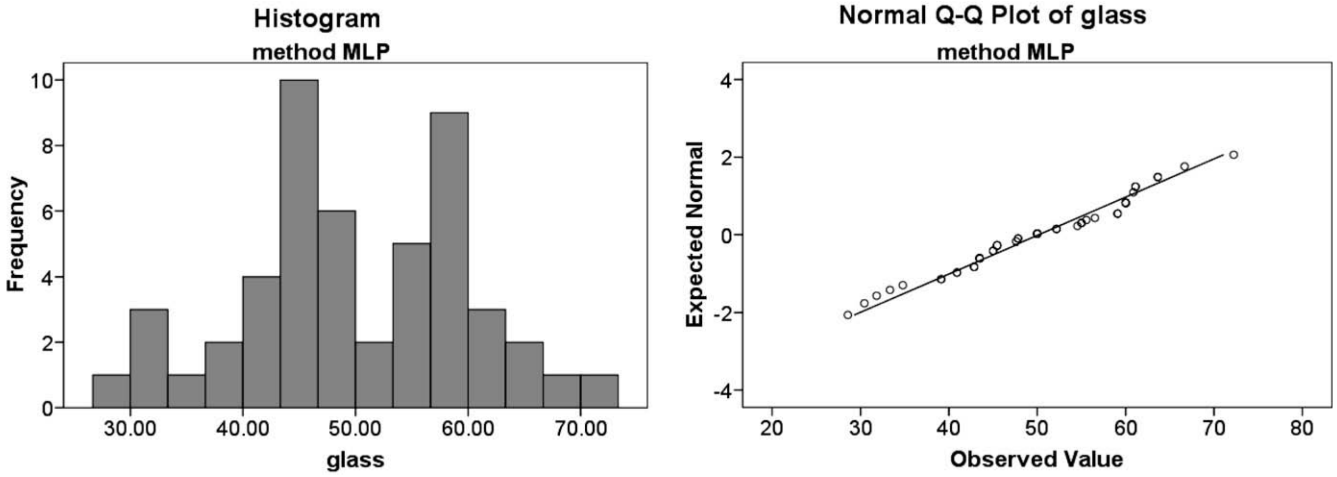

Fig. 3. Results of MLP BackProp over glass data set in 10FCV: histogram and Q–Q graphic.

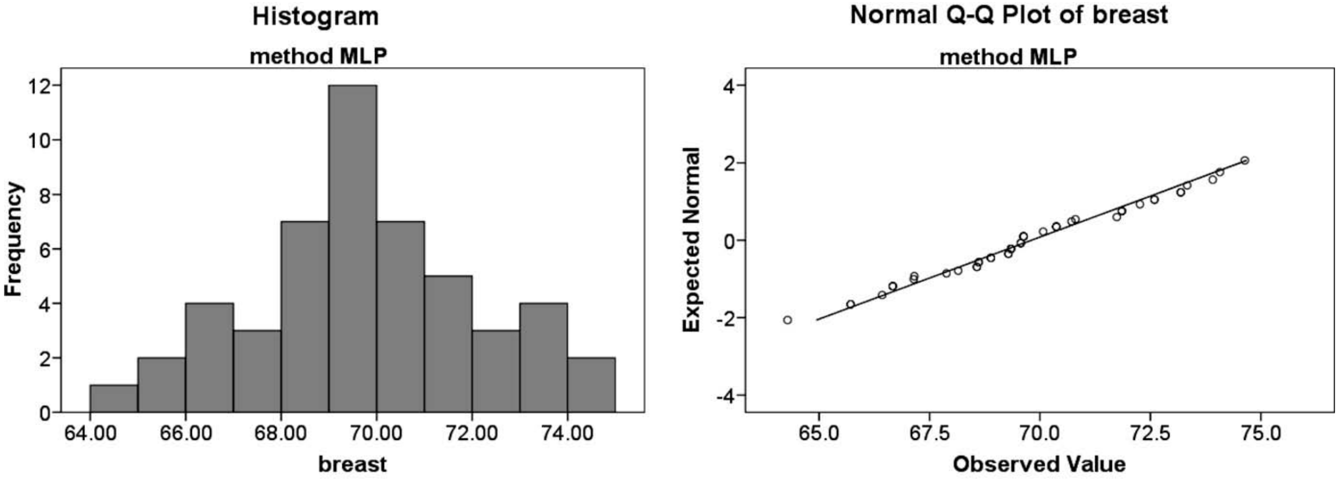

Fig. 4. Results of MLP BackProp over breast data set in 5 × 2CV: histogram and Q–Q graphic.

normality of the distribution due to, as we can observe in the Q–Q graphic of the Fig. 2, there are few points around the line, so this implies that the distribution contains many ties.

Fig. 3 represents a case in which a normality test rejects the null hypothesis whereas the other tests cannot do it. Kolmogorov–Smirnov's test rejects the normality hypothesis but Shapiro–Wilk and D'Agostino–Pearson do no reject it. Note that the histogram of the Fig. 3 seems to adopt an approximate Gaussian shape, except for the bar in the centre of the graphic; given that the Shapiro–Wilk's and D'Agostino–Pearson tests are usually more powerful than the Kolmogorov–Smirnov's (Zar, 1999), it is a case in which Kolmogorov–Smirnov's test obtains a false negative error.

Finally, in Fig. 4 we show an example in which the normality hypothesis is not rejected by the three tests used.

4. On the use of rank-based non-parametric tests: a short experimental study

In this section, we briefly introduce non-parametric tests used and we present an experimental study using the seven algorithms described. Non-parametric tests can use the mean values obtained for each data set, so they can treat with probabilistic and non-probabilistic methods without any restriction. We will use a simple multiple comparison procedure to show a case study of comparing simultaneously more than two methods through non-parametric tests.

4.1. Rank-based non-parametric tests

A non-parametric test is such that uses nominal data, ordinal data or ranked data. However, this does not mean that other data types cannot be used. It could be interesting to transform real data from an interval into ranked data by means of their order, so non-parametric tests can be applied on data which is typically used by parametric tests (when conditions for parametric tests application are not verified). Usually, a non-parametric test is less restrictive than parametric one, but less robust than a parametric test applied over data which verifies all needed conditions.

Next, we show the basis of each non-parametric tests used in this study:

- Friedman test (Sheskin, 2003), which is a non-parametric test equivalent of the repeated-measures ANOVA. Under the null hypothesis, it states that all the algorithms are equivalent, so a rejection of this hypothesis implies the existence of differences among the performance of all the algorithms studied. After this, a post-hoc test could be used in order to find whether the control or proposed algorithm presents statistical differences with regards to the remain of methods into the comparison. One of them is the Bonferroni–Dunn test.Friedman test way of working is described as follows: It ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2, and so on. In case of ties average ranks are assigned.Let r_i^j be the rank of the jth of k algorithms on the ith of N data sets. The Friedman test compares the average ranks of algorithms, $R_j = \frac{1}{N}\sum_i r_i^j$. Under the null hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum jR_j^2 - \frac{k(k+1)^2}{4}\right] \quad (1)$$

  is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough (as a rule of a thumb, $N > 10$ and $k > 5$).
- Iman and Davenport test (Iman & Davenport, 1980), which is a non-parametric test, derived from the Friedman test, less conservative than the Friedman statistic:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2} \quad (2)$$

  which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. Statistical tables for critical values can be found at (Sheskin, 2003; Zar, 1999).
- Bonferroni–Dunn is a post-hoc test that can be used after Friedman or Iman–Davenport tests when they reject the null hypothesis. It is similar to Dunnet's test for ANOVA. This method assumes that the performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference:

$$\mathrm{CD} = q_\alpha \Big/ \sqrt{\frac{k(k+1)}{6N}} \quad (3)$$

  q_α value is the critical value Q' for a multiple non-parametrical comparison with a control (see Table B.16 in Zar (1999)).
- Holm's test (Holm, 1979): it is a multiple comparison procedure that can work with a control algorithm and compares it with the remaining methods. The test statistics for comparing the ith and jth method using this procedure is

$$z = (R_i - R_j) \Big/ \sqrt{\frac{k(k+1)}{6N_{ds}}}$$

  The z value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate level of confidence α. In Bonferroni–Dunn comparison, this α value is always $\alpha/(k-1)$, but Holm's test adjusts the value for α in order to compensate for multiple comparison.Holm's test is a step-up procedure that sequentially tests the hypotheses ordered by their significance. We will denote the ordered p-values by $p_1, p_2, \ldots,$ so that $p_1 \leqslant p_2 \leqslant \ldots \leqslant p_{k-1}$. Holm's test compares each p_i with $\alpha/(k-i)$, starting from the most significant p value. If p_1 is below $\alpha/(k-1)$, the corresponding hypothesis is rejected and we allow to compare p_2 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 remain hypotheses are retained as well.
- Hochberg's procedure (Hochberg, 1988): it is a step-up procedure that works in the opposite direction to Holm's method, comparing the largest p-value with α, the next largest with $\alpha/2$ and so forth until it encounters a hypothesis that it can reject. All hypotheses with smaller p-values are then rejected as well.

The post-hoc procedures described above allow us to know whether or not a hypothesis of comparison of means could be rejected at a specified level of significance α. However, it is very interesting to compute the p-value associated to each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. In this manner, we can know whether two algorithms are significantly different and also a metric of how different they are.

In the following, we will describe the method for computing these exact p-values for each test procedure, which are called "adjusted p-values" (Wright, 1992).

- The adjusted p-value for Bonferroni–Dunn's test (also known as the Bonferroni correction) is calculated by $p_{\mathrm{Bonf}} = (k-1)p_i$.
- The adjusted p-value for Holm's procedure is computed by $p_{\mathrm{Holm}} = (k-i)p_i$. Once computed all of them for all hypotheses, it is not possible to find an adjusted p-value for the hypothesis i lower than for the hypothesis j, $j < i$. In this case, the adjusted p-value for hypothesis i is set equal to the associated to the hypothesis j.

Table 15
Results for Friedman and Iman–Davenport tests.

Method	Friedman Value	Value of χ^2	Iman–Davenport Value	F_F Value
HOV	**21.609**	12.592	**4.357**	2.201
10FCV	**24.188**	12.592	**5.052**	2.201
5×2CV	**13.333**	12.592	**2.419**	2.201

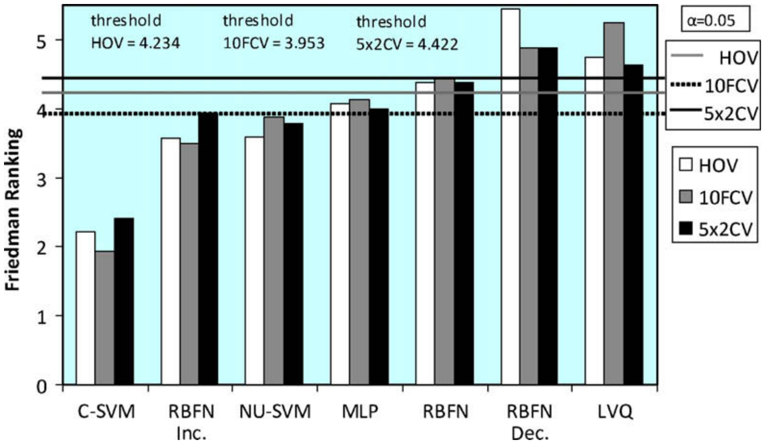

Fig. 5. Bonferroni–Dunn graphic for all validations.

- The adjusted *p*-value for Hochberg's method is computed with the same formula as Holm's, and the same restriction is applied in the process, but in the opposite sense, that is, it is not possible to find an adjusted *p*-value for the hypothesis *i* lower than for the hypothesis *j*, $j > i$.

4.2. Experimental study: results and analysis

In Table 15 we show the results of applying the tests of Friedman and Iman–Davenport in order to detect whether differences in the results exist. In bold appears the highest value of the compared ones, and if it is the corresponding statistical (column named "Friedman value" or "Iman–Davenport value"), then the null hypothesis is rejected. In case contrary, the null hypothesis is not rejected, so the results are not significantly different. Both test are applied with a level of confidence $\alpha = 0.05$.

In our case, both Friedman's and Iman–Davenport's tests indicate that significant differences in the results are found in the three validations used in this study. Due to these results, a post-hoc statistical analysis is required. In this snalysis, we choose the best performing method, *C*-SVM, as the control method for being compared with the rest of algorithms.

In Fig. 5, we illustrate the application of Bonferroni–Dunn's test. This graphic represents a bar chart, whose bars have a height proportional to the average rank obtained for each algorithm by following the procedure of Friedman. In each type of validation used, if we sum the value of ranking of the lowest bar (which is associated with the best algorithm, the control algorithm) to the Critical Difference (CD) value, we obtain a horizontal line (denoted as "Threshold"), which is displayed along the graphic. Those bars that exceed this line are the associated ones with the algorithms whose performance is significantly worse than the control algorithm (associated with the lowest bar).

As we can see in Fig. 5, three threshold lines are drawn corresponding to each type of validation. Bonferroni–Dunn's test indicates us that

- In HOV, the method *C*-SVM used as control is statistically better than RBFN, RBFN Dec. and LVQ.
- In 10FCV, the control method improves MLP, RBFN, RBFN Dec. and LVQ significantly.
- Bonferroni–Dunn's test indicates that, in 5 × 2CV, *C*-SVM outperforms RBFN Dec. and LVQ.

We will apply more powerful procedures, such as Holm's and Hochbergs's, for comparing the control algorithm with the rest of algorithms. Tables 16–18 show all the adjusted *p*-values for each comparison which involves the control algorithm. The *p*-value is indicated in each comparison and we stress in bold the algorithms which are worse than the control, considering a level of significance $\alpha = 0.05$.

Tables 16–18 indicate us that:

- In HOV, the method *C*-SVM used as control is statistically better than RBFN, RBFN Dec., MLP and LVQ.
- In 10FCV, the control method outperforms all the remaining methods.
- Holm's and Hochberg's tests indicate that, in 5 × 2CV, *C*-SVM outperforms RBFN Dec., RBFN and LVQ.

Table 16
Adjusted *p*-values in HOV (*C*-SVM is the control).

i	Algorithm	Unadjusted *p*	p_{Bonf}	p_{Holm}	p_{Hoch}
1	RBFN Decremental	$2.505 \cdot 10^{-5}$	$1.503 \cdot 10^{-4}$	$1.503 \cdot 10^{-4}$	$1.503 \cdot 10^{-4}$
2	LVQ	$9.191 \cdot 10^{-4}$	0.00551	0.0046	0.0046
3	RBFN	0.00475	0.02853	0.01902	0.01902
4	MLP	0.01578	0.09466	0.04733	0.04733
5	*NU*-SVM	0.07181	0.43088	0.14363	0.07851
6	RBFN Inc.	0.07851	0.47108	0.14363	0.07851

Table 17
Adjusted *p*-values in 10FCV (*C*-SVM is the control).

i	Algorithm	Unadjusted *p*	p_{Bonf}	p_{Holm}	p_{Hoch}
1	LVQ	$1.443 \cdot 10^{-5}$	$8.663 \cdot 10^{-5}$	$8.663 \cdot 10^{-5}$	$8.663 \cdot 10^{-5}$
2	RBFN Decremental	$1.2 \cdot 10^{-4}$	$7.201 \cdot 10^{-4}$	$6.001 \cdot 10^{-4}$	$6.001 \cdot 10^{-4}$
3	RBFN	0.00106	0.00638	0.00425	0.00425
4	MLP	0.00418	0.02509	0.01255	0.01255
5	*NU*-SVM	0.01119	0.06713	0.02238	0.02238
6	RBFN Inc.	0.04078	0.24466	0.04078	0.04078

Table 18
Adjusted *p*-values in 5 × 2CV (*C*-SVM is the control).

i	Algorithm	Unadjusted *p*	p_{Bonf}	p_{Holm}	p_{Hoch}
1	RBFN Decremental	0.00128	0.00737	0.00737	0.00737
2	LVQ	0.00367	0.02203	0.01836	0.01836
3	RBFN	0.00995	0.05968	0.03978	0.03978
4	MLP	0.03691	0.22149	0.11074	0.07181
5	RBFN Inc.	0.04498	0.26986	0.11074	0.07181
6	*NU*-SVM	0.07181	0.43088	0.11074	0.07181

Holm's and Hochberg's tests find more significant differences than Bonferroni–Dunn's and their use is as correct as using the latter. However, they are more difficult to conduct and understand. Note that, in this case study, we have considered a level of significance $\alpha = 0.05$ and we have used 16 data sets for analyzing 7 algorithms. These three factors are very important in the non-parametric statistical analysis since they have much influence in the computation of the rankings and in the search of the critical values in the statistical tables.

5. Conclusions

The present work studies the use of statistical techniques for analyzing Artificial Neural Networks in classification problems and a further analysis of parametric and non-parametric tests.

The need of using non-parametric tests for results' analysis in classification with ANNs is very clear, since initial conditions required for obtaining safe conclusions from parametric tests are not met.

On the use of non-parametric tests, we have shown an example of performing a multiple comparison among several algorithms. In this study, we have employed the tests of Friedman, Iman–Davenport, Bonferroni–Dunn, Holm and Hochberg and we recommend them as a good set of testing algorithms.

References

Alpaydin, E. (1999). Combined 5×2 cv F test for comparing supervised classification learning algorithms. *Neural Computation, 11*, 1885–1892.

Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems, 16*(12), 1445–1473.

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems, 2*, 321–355.

Castillo-Valdivieso, P. A., Merelo, J. J., Prieto, A., Rojas, I., & Romero, G. (2002). Statistical analysis of the parameters of a neuro-genetic algorithm. *IEEE Transactions on Neural Networks, 13*, 1374–1394.

Cortes, C., & Vapnik, V. (1995). Support-vector network. *Machine Learning, 20*, 273–297.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 130.

Dietterich, D. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1924.

Fan, R.-E., Chen, P.-H., & Lin, C.-J. (2005). Working set selection using second order information for training SVM. *Journal of Machine Learning Research, 6*, 1889–1918.

Gao, D., Madden, M., Chambers, D., & Lyons, G. (2005). Bayesian ANN classifier for ECG arrhythmia diagnostic system: A comparison study. *Proceedings of the International Joint Conference on Neural Networks, 4*, 2383–2388.

García-Pedrajas, N., & Fyfe, C. (2007). Immune network based ensembles. *Neurocomputing, 70*, 1155–1166.

Groves, D. J., Smye, S. W., Kinsey, S. E., Richards, S. M., Chessells, J. M., Eden, O. B., et al. (1999). A comparison of cox regression and neural networks for risk stratification in cases of acute lymphoblastic leukaemia in children. *Neural Computing and Applications, 8*, 257–264.

Hochberg, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika, 75*, 800–803.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics, 6*, 65–70.

Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics, 9*, 571–595.

Kim, Y. S. (2008). Comparison of the decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size. *Expert Systems with Applications, 34*(2), 1227–1234.

Lam, M. (2004). Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decision Support Systems, 37*, 567–581.

Lawrence, S., Back, A. D., Tsoi, A. C., & Giles, C. L. (1997). On the distribution of performance from multiple neural-network trials. *IEEE Transactions on Neural Networks, 8*(6), 1507–1517.

Li, S.-T., Shiue, W., & Huang, M.-H. (2006). The evaluation of consumer loans using support vector machines. *Expert Systems with Applications, 30*, 772–782.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). UCI Repository of machine learning databases. Irvine: Department of Information and Computer Sciences, University of California. Availabel from http://www.ics.uci.edu/~mlearn/MLRepository.html.

Pizarro, J., Guerrero, E., & Galindo, P. L. (2002). Multiple comparison procedures applied to model selection. *Neurocomputing, 48*, 155–173.

Powell, M. J. D. (1987). Radial basis function for multivariate interpolation: A review. In J. C. Mason & M. G. Cox (Eds.), *Algorithm for approximation* (pp. 143–168). Oxford: Clarendon Press.

Qiu, X., Tao, N., Tan, Y., & Wu, X. (2007). Constructing of the risk classification model of cervical cancer by artificial neural network. *Expert Systems with Applications, 32*(4), 1094–1099.

Rojas, R., & Feldman, J. (1996). *Neural networks: A systematic introduction*. Springer.

Schölkopf, B., Smola, A., Williamson, R., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation, 12*, 1207–1245.

Sheskin, D. J. (2003). *Handbook of parametric and non-parametric statistical procedures*. CRC Press.

Wright, S. P. (1992). Adjusted *p*-values for simultaneous inference. *Biometrics, 48*, 1005–1013.

Wu, T.-K., Huang, S.-C., & Meng, Y.-R. (2008). Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities. *Expert Systems with Applications, 34*(3), 1846–1856.

Yingwei, L., Sundararajan, N., & Saratchandran, P. (1997). A sequential learning scheme for function approximation using minimal radial basis function neural networks. *Neural Computation, 9*, 361–478.

Zar, J. H. (1999). *Biostatistical analysis*. Prentice Hall.

Zekic-Susac, M., & Horvat, J. (2005). Modeling computer and web attitudes using neural networks. *Proceedings of the International Conference on Information Technology Interfaces, 4*, 2383–2388.