Neurocomputing 353 (2019) 28–44

Contents lists available at ScienceDirect

Neurocomputing

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

Semi-wrapper feature subset selector for feed-forward neural networks: Applications to binary and multi-class classification problems

Antonio J. Tallón-Ballesteros[a,*], José C. Riquelme[a], Roberto Ruiz[b]

[a] Department of Languages and Computer Systems, University of Seville, Reina Mercedes Avenue, Seville 41012, Spain
[b] Area of Computer Science, Pablo de Olavide University, Km. 1, Utrera Road, Seville 41013, Spain

ARTICLE INFO

Article history:
Received 15 December 2017
Revised 8 April 2018
Accepted 26 May 2018
Available online 13 March 2019

Keywords:
Feed-forward artificial neural networks
Feature selection
Supervised machine learning
Feature subset selection
Computational intelligence
Semi-wrapper

ABSTRACT

This paper explores widely the data preparation stage within the process of knowledge discovery and data mining via feature subset selection in the context of two very well-known neural models: radial basis function neural networks and multi-layer perceptron. It is known the best performance of wrapper attribute selection methods based on the evaluation measure provided by a classifier, although the temporal complexity of learning neural networks practically precludes the use of wrapper techniques, especially in complex databases with high dimensionality and a large number of labels. In this paper, we propose the use of the Naïve Bayes classifier as a fitness function within a semi-wrapper feature selection approach. The Naïve Bayes classifier is a good fast approach to a neural network and utilising it as a measure of goodness in a backward search on a ranking provides a specific attribute selection method for neural networks in complex data. The test-bed consists of 34 binary and multi-class classification problems and 7 feature selectors. Of these, there are 6 data sets with upwards of 5 classes. According to the reported accuracy results that have been supported by non-parametric statistical tests in different scenarios, our method has been shown to be very suitable for both kinds of neural networks. Moreover, the reduced feature-space is around 20% of the full attribute space. The speedup with the aforementioned semi-wrapper is very outstanding and its value fluctuates, on average, from about 1.5 with radial basis function neural networks to around 30 with multi-layer perceptron.

1. Introduction

Neurocomputing deals with information processing. It involves a learning procedure within artificial neural network architecture. The trained networks can be utilised to perform certain tasks depending on the particular application that we are coping with. Processing in neurons can be very complex, although with the basic limitations of speed and accuracy imposed by the biophysical properties of ions and membranes. Integration of information in dendrites is often non-linear [1]. Neurocomputing can play an important role to solve certain difficult problems in science and engineering such as pattern recognition, optimisation, control and identification of nonlinear systems, and statistical analysis [2]. Computational neuroscience is a field that strives to simulate and understand the function of the nervous system. Among its disciplines, it encompasses artificial neural networks (also called neural networks) [3]. Neural networks tie, to some extent, with expert systems in the sense that for concrete domains a knowledge base in terms of rules could be incorporated to the initial neural networks [4]. The application scope of neural networks is very broad and we can find some degree of success in finances, engineering diagnosis, intelligent manufacturing, human resource management, medicine, failure detection and biology [5]. In this world of digital data, now more than ever, it is important to distinguish the most relevant properties or attributes, in order to maximise the objective, we are pursuing in the learning that we intend to carry out from the data. The advantages that can be obtained are already widely agreed, such as obtaining simpler and clearer learning models; equalise, and even improve, the results of the model, classification in our case; accelerate the learning process, although the cost of the selection in the pre-processing is added; and once the chosen attributes have been tested, it could be proposed for future cases, to capture only these attributes saving the effort of entering and storing the rest of the properties.

* Corresponding author.
E-mail address: atallon@us.es (A.J. Tallón-Ballesteros).

https://doi.org/10.1016/j.neucom.2018.05.133
0925-2312/

We have performed a scrupulous review of the research within the area bearing in mind feature selection and at the same time the application domain of classifiers based on neural networks. According to the study, we must state that most of the existing publications are focused on a specific type of classifier or just a few methods to reduce the input space have been reported. This paper aims to shed light on the performance of different filter-based selection methods, to suggest which pathways will be most promising conducting future experiments in the scope of aforementioned neural approaches. Particularly, we propose a semi-wrapper approach by means of a Naïve Bayes classifier that could help the neural network models to process only the subset of inputs which are the most relevant. At the same time, it promotes a way to overcome the performance of feature subset selection implemented as filters and a faster solution to wrapper methods, which are very specific with limited application to other variations or topologies of neural networks, and would require high amounts of memory and computing time.

In the context of feature selection there is a trade-off between efficacy and efficiency; some methods pick up a reduced number of features but the accuracy does not improve in comparison to the full set of features, whilst conversely some methods select a larger number of features and give an excellent classifier performance. Alternatively, some methods may even worsen the accuracy. Nonetheless the important issue is to demonstrate if there are significant differences. As proof of the research conducted, we can assert that feature selection is always convenient, especially in the computation of neural networks and semi-wrappers are strong approaches to deal with complex problems. It is also crucial to complete the paper with some prospective works.

As a test-bed for this paper, we consider two outstanding models of feed-forward neural networks: Multi-layer Perceptron (MLP) and Radial Basis Function Neural Networks (RBFNN) averaged with 30 runs into 34 binary and multi-class classification problems taking into account both full and reduced feature space. Seven feature selection approaches based on subset of attributes have been conducted.

The remaining part of this paper is organised as follows: Section 2 describes briefly the methods that are the core elements to follow easily the forthcoming parts of this paper. Section 3 starts with a motivation to the approach and introduces the proposal. Section 4 details the experimentation. Section 5 reports on the results and also undergoes with non-parametric statistical tests. Section 6 draws the conclusions. Lastly, Section 7 opens new research lines upon the basis of all that this paper concludes.

2. Methods

2.1. Neural networks

Artificial neural networks are computational models that emulate the human brain. The complexity of the brain is such that around 10^{11} neurons are present, on average, in the human brain. It also means that a huge amount of connections govern the body [6]. Furthermore, the word connectionism has been applied to the neural networks due to storage of the knowledge in a neural network as a set of connections with weights to excite or inhibit the signals coming from the previous layer [7]. Connectionist framework was one of the two main approaches in the 1950′s and early 1960′s within the emerging field of artificial intelligence [8]. Learning plays a crucial role in neural networks [9]. Firstly, the first task within neural networks is to train or learn the model to represent the input patterns. Secondly, the model should be assessed with unseen data; this phase is known as the generalisation task [10].

Neural networks could be divided into two types: feed-forward neural networks and feed-back or recurrent neural networks [11]. Nowadays, the latter type is the subject of many control systems and also lot of dynamic environments, whereas the former has mainly widespread applications in the areas of pattern recognition and business intelligence just to name a few. On the one hand, feed-forward neural networks are categorised mainly into three variations: (a) Single-layer perceptron that consists of one input layer and one output layer; nowadays this approach is rarely considered in practice, (b) multi-layer perceptron, typically this model follows a three-layered structure with an input layer, a hidden layer and an output layer, (c) radial basis function nets with a structure similar to that of the previous type. In particular, we adopt two forms of very well-known universal approximators namely MLP [12] and RBFNN [13].

The most popular class of multi-layer feed-forward networks is multi-layer perceptron in which each computational unit (or node) utilises either the threshold function or the sigmoid function [13]. There are a great number of different approaches to train an MLP architecture but all of them are based on gradients or sometimes on heuristics. Generally speaking, the basis of MLP is the Back-propagation (BP) learning algorithm to determine weights. We utilises the BP approach with momentum which is superior to the usual BP algorithm.

The Radial Basis Function (RBF) neural network consists of two layers and is a special type of multi-layer feed-forward network. Each unit in the hidden layer uses a radial basis function, represented by a Gaussian kernel, as its activation function [14]. The radial basis function (or kernel function) is centred at the point indicated by the weight vector associated with the unit. Both the positions and the widths of these kernels must be learned from training patterns. There are usually fewer kernels in the RBF network than there are training patterns. Each output unit performs a linear combination of these radial basis functions. From the viewpoint of function approximation, the hidden units construct a set of functions that constitute a base set for representing input patterns in the space spanned by the hidden units. The training is done by a *K*-means algorithm [16].

2.2. Feature selection

Feature selection pursues to determine a subset of variables from the input which can efficiently describe the input data and at the same time reducing effects from noise or irrelevant variables and still providing good prediction results. Articles such as [15,16] clearly expose different ways of classifying selection algorithms, and include descriptions comparing advantages and disadvantages. Feature selection algorithms have two essential elements, the evaluation measure used to quantify the goodness of the subsets of attributes, and the strategy followed by the search method in order to locate a good subset that comes as close as possible to a global maximum [17]. If we take into account the first component of this type of algorithm, the evaluation function, we can distinguish selectors that use a metric that is unrelated to the type of learning that we will later use, from selectors that use the learning algorithm itself as an evaluation measure for the subsets. The formers are called filters, the latter wrappers. Some algorithms offer a solution by selecting the attributes in the learning process itself, known as embedded methods. Within the filter category there are several widely utilised metrics, such as correlation measures, consistency measures, information gain and dependency, among others. In the state-of-the-art of the selection of attributes, we can find many hybrid selection methods, the result of mixing different ways of evaluating the subsets [18,19]. There are also different possibilities depending on the search strategy, the second component of a selector algorithm: exhaustive, heuristic, random, etc.

In the experiments detailed in this document we see the results obtained with different combinations of the following metrics:

- CBF - Correlation-based filter [20]: CFS assesses the quality of a feature subset bearing in mind the hypothesis that good feature subsets contain features highly correlated to the class.
- CNS - Consistency-based measure [21]: CNS is founded on the consistency measure, which estimates, for a given subset of features, the number of sources that matches all but their class labels. The inconsistency rate is then utilised to evaluate its quality.
- SOAP - Selection of attributes by projection [22]: It is a non-stochastic feature selection criterion based on the basic principle of counting the label changes of examples projected onto each feature. If the attributes are sorted in ascending order according to the number of label changes, we have a list that defines the priority of selection, from greater to smaller importance. The main advantages are its speed and simplicity in the assessment of the attributes.
- Naïve Bayes is a descriptive and predictive classification technique based on the probability theory of the analysis of Bayes [23]. It calculates the probability distributions of each class to establish the relationship between the attributes and the class. The Bayesian classifier is a simple and fast method.

From a different point of view, based on how the output of the attribute selectors is produced, we find methods that at the end of the process return a defined subset of attributes, compared to other methods that perform the process of ordering the attributes according to the criteria set, that is, the function that is used to evaluate can only be applied to each attribute individually. Hybrid algorithms use both types of output in different parts of the process have been used for a little more than a decade [24–28]. They usually start with a first ranking phase, which is followed by another phase of subset formation. In some versions this process is repeated whilst alternating both phases. As we can imagine, taking into account the classifications of the aforementioned selectors, there are numerous possible algorithms for each configuration. In this sense, perhaps, the most referenced algorithm that follows this type of hybrid strategy, FCBF - Fast Correlation-Based Filter [29]. This selector establishes the concept of a Predominate feature based on the non-linear correlation metric SU – Symmetrical Uncertainty.

3. Related works and contribution

3.1. Neural networks and feature selection

The topic of feature selection in the context of RBFNN and MLP has been touched by some researches. The idea of accelerating the training time by means of feature selection had an important contribution in 1997 by Setiono and Liu; firstly, they proposed to train the MLP neural network with an approach faster than BP [30] and secondly they introduced a procedure to prune the network which may be considered similar to a backward search in the sense that for each attribute is computed the accuracy of the whole network without the current attribute, later the feature with the smallest decrease is removed and then the process is repeated. This work tested the proposal with six artificial and four binary real-world datasets and among them Sonar problem that contains sixty feature and two labels is the most complex of their study. One of the pioneered works for RBFNN was published in 1999 by Basak and Mitra [31], who made a contribution to compute two new evaluation indices to select the optimal set of features which were derived from the worsening of the separation between and/or compactness of the classes due to the absence from the feature set. They assessed the approach in a couple of real-life problems such as Iris and Vowel (six classes, one for every of the Telugu vowel sounds) and provided desired feature rankings.

Probably, the work more connected to the sketch of the current paper could be one that utilised MLP and RBFNN to train only a concrete problem related with the security containing samples from four classes with a number of properties in the most difficult case around one hundred [32]; the contribution is based on the usage of a sequential forward selection to extract the relevant features and also is important to remark that the search starts with an empty candidate set and adds feature variables sequentially until the halt. Nonetheless, the forthcoming pages talk about the new contribution that we propose and, of course, the experiments to evaluate our proposal. We have found a contribution using several feature ranking methods and also testing two wrappers approach after an initial pre-screening in RBFNN [33]; the idea of this paper is to avoid one of the drawbacks of assessing the attributes individually because there are difficulties to get features with a good performance alone and also within a subset. This paper has received a very good welcome from the data mining community since has been cited around fifty times at the written time of this manuscript. From previous works, it has been shown [34] that filters based on subsets which reduce the space attribute too much could lead to the deletion of relevant properties in terms of entries to the neural network and also be accompanied by a not so good performance compared to other approaches that chose a moderate number of attributes.

Shifting mainly to the wrapper model, some researches may be found. In 2002, Hsu et al. introduced a framework called ANNIGMA [35], which stands artificial neural net input gain measurement approximation, and is founded in assigning a weight to every input according to the information gain measure. Another interesting work from the previous decade suggested computing the relative contribution of each feature to the target label and reached to the conclusions that not always an important reduction in the feature space is productive for the neural network model [36]. Yang et al. [37] presented a wrapper for MLP which has been assessed in eight problems and some of them are very remarkable such as two data sets concerned with binary problems with up to one hundred and sixty six features and another one with seven labels and nineteen attributes. In the current decade, there have been some contributions in the field of neural networks using wrapper approaches. As an example, it is of special interest the constructive approach for feature selection (CAFS) [38] which is tested in problems with up to sixty features and three classes or eighteen attributes and four classes as two representative cases with the highest computational cost. Zeng et al. [39] has proposed very recently a wrapper based on sensitivity for RBFNN that has been applied in three classification problems and the most complex one has been waveform which has forty features and three labels. A newly written review outlines further perspectives for feature selection and introduces an in-depth study about unsupervised feature selection both for the case of the filter and wrapper models [40].

We have explored many more contributions although the real novelty is more limited compared to all the papers cited below. Thus, scenarios with hundreds of features and more than five classes, which may be considered to have higher complexities that the aforementioned ones, have not been addressed yet in the context of wrapper-based feature selection. Moreover, the limited applicability to other classifiers, even to those following the same strategy of coping with the data representation or the elements to conduct the decision-making procedure, make the wrappers said to be as too specific approaches; for example, a solution for an RBF neural network is not valid for other RBF with a different kernel or even for an MLP.

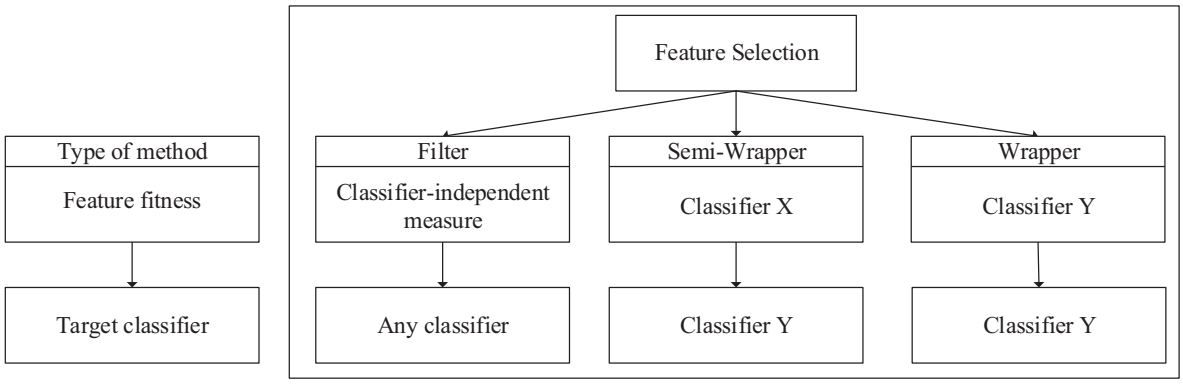

Fig. 1. Taxonomy of feature selection: filter, semi-wrapper and wrapper.

3.2. Proposal

Bearing in mind all the previous studies, we contemplate the possibility of incorporating intermediate methods between both extremes of the classical splitting of feature selection approaches into two large groups, namely, the filters and the wrappers. As a result of it, we propose the semi-wrapper feature selector as a faster method than wrappers and, at the same time, more accurate and reliable than filters. Additionally, a solution reached using a classifier to compute the feature fitness is suitable to be taken as the input for a target classification algorithm which must be different, falling in the same or a different category within the taxonomy of classifiers. Having said the above, we may apply a fast and, optionally, very competitive classification algorithm to get a good score with the available data and then to feed the target classifier with a very robust subset of attributes. To a certain extent, the semi-wrapper may promote the hybridisation at the classifier level within the feature selection task. Fig. 1 depicts the taxonomy of feature selection with the inclusion of the semi-wrapper in between the classical approaches such as filter and wrapper.

It is important to stress that we are contending with two types of feed-forward models independently. Thus, we are working not only with one neural approach but also with two different architectures and very extensively according to the test-bed which adds an extra contribution to this paper. Our focus is primarily on feature subset selection due to its better performance in comparison with feature ranking methods. Once we have considered working with feature subset selection, we have proposed an intermediate solution between filter and wrappers since the former are not able to reach solutions close to the optimal ones and the latter require an important computational burden for the neural network models.

The main idea that guides this work is to have a good method of feature selection that allows optimising the application of neural networks in high-dimensional environments. In recent years the common way of grouping feature selection algorithms has not changed much, and from above mentioned studies [15,16], we could gather:

1. First, feature selection, as a dimensionality reduction technique, focuses on choosing a subset of the **relevant attributes of the original set, discarding irrelevant, redundant and noisy attributes**.
2. **Wrapper** methods obtain better results from a classification accuracy point of view, using much more time [41], becoming prohibitive in the case of heavy classifiers as it happens with neural networks.
3. As regards the selector output, ranking type are much faster than those that provide subsets, but their drawback is that catches only the relationship of each individual attribute with the class, rather than the interrelationships between attributes. In addition, we have the issue of knowing how many attributes of the ranking we are left to form the subset of final attributes. **Hybrid algorithm (ranking + subset)** performs more agile searches, obtaining good results even in large data sets in which the algorithms of subsets may not arrive.
4. Regarding the search direction when the search space is traversed, that is, the relationship between the attributes of a subset with the next subset, **backward** direction approach remedies a disadvantage of the forward path, such as not detecting interesting basic interactions between attributes from a classification point of view. This is achieved by eliminating from the subset the least relevant attribute of all.

Taking into account the four aforementioned items, and especially the ideas outlined in bold, we propose a semi-wrapper version of the hybrid algorithm BIRS (Best Incremental Ranked Subset) [25] that we have called $BIRS_{SW}$ and is introduced concretely for the particular scenario for a feature selection prior to a classification process with neural networks. Fig. 2 shows the proposal.

The first part of the algorithm would be similar to other hybrid approaches; it generates a ranking where the attributes can be ordered according to any type of metric. The proposal is nevertheless to modify conveniently the second part of the algorithm. First, as noted above, the direction of the search algorithm is changed, starting with the complete set of features and through a backward search are successively eliminated. To do this, the attributes are chosen one by one in the reverse order of the previous ranking and if the evaluation of the subset without the attribute is greater or equal than with it, it is removed.

Due to the impossibility of using the neural networks themselves as an evaluation measure in the selection of subsets of attributes, we use a learning algorithm, Naïve Bayes (NB), as an evaluation metric providing a lower temporal cost. The benefits of the Naïve Bayes classifier are widely known, as listed in [42], computational efficiency, low variance, incremental learning, direct prediction of posterior probabilities, robustness in the face of noise, and robustness in the face of missing values. In our work, NB has been chosen as a subset evaluator mainly because of its similarity to neural networks. The resulting NB classifier uses a linear model, equivalent to that used by logistic regression (and therefore single-layer neural nets), differing only in the manner in which the parameters are chosen [42,43]. Because NB is not the final destination classifier, our proposal cannot be considered a wrapper, which is why we call it a semi-wrapper.

In addition, as discussed earlier in the first point, the use of NB as subset evaluator in the second phase reinforces the objective of any attribute selector, choosing a subset of relevant

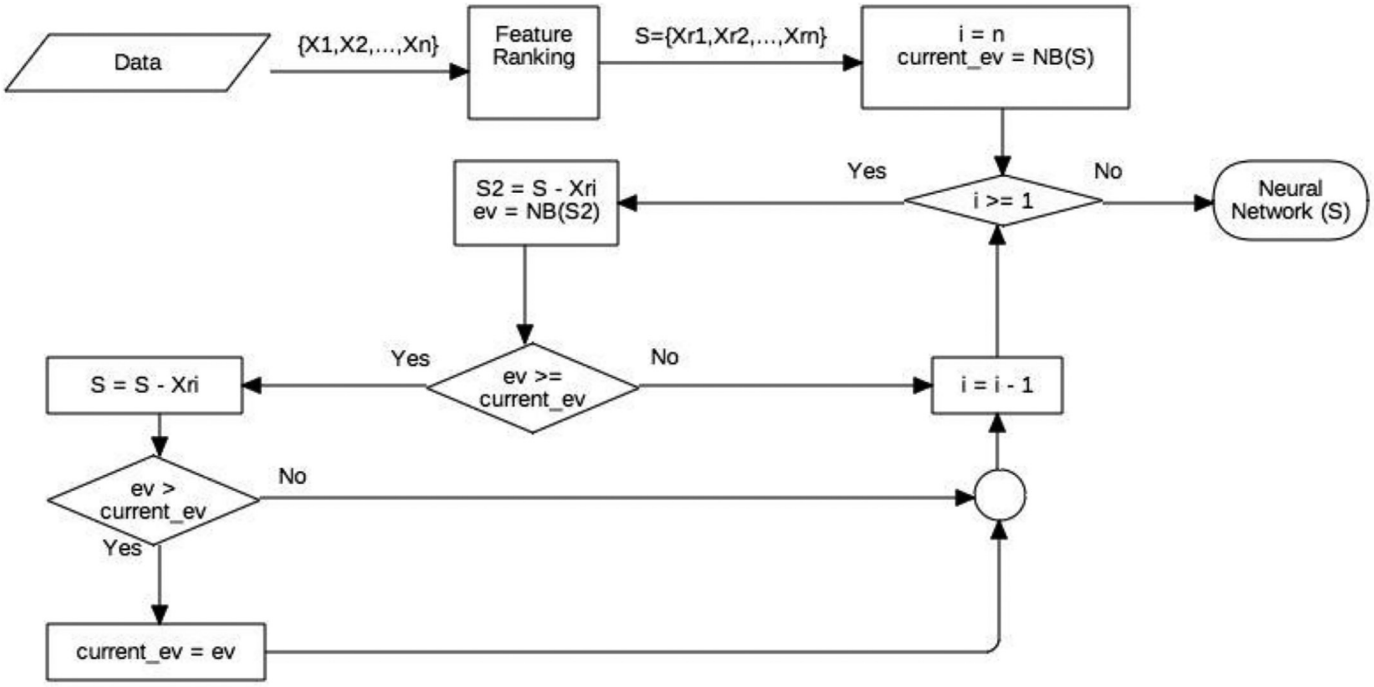

Fig. 2. Proposed approach: $BIRS_{SW}$. A semi-wrapper feature subset selector.

attributes of the original set by discarding irrelevant, redundant, and noisy attributes. First, irrelevant features have a very small or no correlation with the class variable, and so, have very little or no predictive power. Liu and Motoda [44] and Kohavi [45] have observed that theoretically, the irrelevant features should not affect the classification outcome for Naïve Bayes classification. They have argued that even though, theoretically, the removal of any feature cannot affect the classification performance of the (optimal) Bayesian classifier, the Naïve Bayes classifier should perform better when irrelevant features are removed. John et al. [46] have observed that in practice (empirically) the irrelevant features lead to degradation in classification performance. Second, NB is very sensitive to redundant attributes because if two or more attributes are correlated they receive too much weight in the final decision as to which class they belong to.

4. Experimentation

4.1. Data sets

Table 1 reports the problems that we have used for the experimentation and which are addressed to assess the performance of the proposal as well as in comparison with other approaches. The test-bed is very varied in terms of properties. The last row shows the mean values of every property. The top part of the table (from D1 to D22) is related mostly to real-world problems taken from the International Repository hosted by the University of California at Irvine (UCI) [47]. The bottom part of the table lists synthetic problems that are very challenging because although the goal is known, the method that should be used to achieve this goal remains unsolved. Third column provides some extra information about the problem in order to ease the reproducibility of the experiments: the original name for data sets from UCI as shown in the repository (for those problems not available on this server some additional details) and a brief description about synthetic ones.

The comparison using artificial data sets has been previously used in [48,49]. Each data set was constructed with n samples represented by pairs $(\overline{x_i}, y_i)$ where each $\overline{x_i}$ is a vector described by d quantitative variables and its corresponding $y_i \in \{-1, 1\}$ is a qualitative attribute that contains the class associated to the vector. The x_{ij} value represents the jth component of the ith example. Following the notation used in [45] the definition of each problem is based on *5* parameters and therefore a data set is characterised by means of the elements *(m,d,r,l,g)* where m represents the number of samples, d the total number of features, r the number of relevant features, l the type of classification rule, and g the noise rate in the features.

A feature is considered to be relevant to the learning task if it is present in the definition of the classification rule that has been taken from [48] and to build the data set it was generated a $r \times l$ $(c_{j,k})$ random matrix with coefficients in $[-2,-1] \cup [+1,+2]$. We utilised this range to avoid coefficients with values close to *0*, which would falsify the subset of relevant attributes. Then, a polynomial of degree l is built, and for each example i we define:

$$p_i = \prod_{k=1}^{l} \left(\sum_{j=1}^{r} (c_{jk} x_{ij}) + b_k \right); \quad y_i = \begin{cases} +1 & \text{if } p_i > \mu \\ -1 & \text{otherwise} \end{cases}.$$

where b_k is a random independent term to assure that all monomials of a specific degree are generated, and μ is the median of p_i, $i = 1,\ldots,m$. Each element x_{ij} was drawn uniformly in [0, 1]. The label y_i of each example $\overline{x_i}$ was assigned considering the equations for $l = 1$ (linear case).

For the experiments conducted through this article, we used data sets with $m = 500$ samples, whilst the number of features were 50 and 100. The number of relevant features was fixed to $r = \{5, 10\}$ for $d = 50$ and $r = \{5, 10, 15, 20\}$ for $d = 100$. Additionally, in order to increase the difficulty, the input values of the data sets were altered by adding label noise that consisted in flipping the class value. The percentage of noise considered for the experimentation was fixed to $g = \{0, 5\}$. To estimate the quality of the hypothesis learned with each subset of features, we utilised the average classification success in a test set independently generated with one third of the size of the training set.

A stratified holdout method has been applied to divide the data sets into two stratified sets, one with 3 out of 4 quarters (the

Table 1
Test-bed summary.

Data set	Problem	Additional information	Instances	Perc_Training	Features	Classes
D1	Batch	Gas Sensor Array Drift, 2012	13,910	74.996	129	6
D2	Breast	Breast Cancer, 1988	286	75.174	9	2
D3	Column	Vertebral Column, 2011	310	74.830	6	2
D4	Heart	Statlog (Heart)	270	74.814	13	2
D5	Hepatitis	Hepatitis, 1988	155	75.483	19	2
D6	Ionos	Ionosphere, 1989	351	74.928	33	2
D7	Labor	Labor Relations, 1988	57	75.438	29	2
D8	Leaves	Data were collected on 16th July 2007; file is not currently stored on the server	180	75.000	43	3
D9	Messidor	Diabetic Retinopathy Debrecen, 2014	1151	74.978	19	2
D10	OBS	Burst Header Packet (BHP) flooding attack on Optical Burst Switching (OBS) Network, 2017	1075	74.976	21	4
D11	Parkinsons	Parkinsons, 2008	195	74.871	23	2
D12	Pasture	From Agricultural researchers in New Zealand, Dave Barker, AgResearch Grasslands, Palmerston North, 1995	36	75.000	21	3
D13	Pima	Pima Indians Diabetes, 1990	768	75.000	8	2
D14	Pollen	From David Coleman at RCA Laboratories in Princeton, N.J.	1372	75.000	47	7
D15	Promoter	Molecular Biology (Promoter Gene Sequences	106	75.471	58	2
D16	Soybean	Sample data set in Weka tool as soybean.arff	683	74.816	82	19
D17	Squash	Squash Harvest (unstored variation), Winna Harvey, Crop & Food Research, Christchurch, 1996	52	75.000	23	3
D18	Tokyo	About hardware failures; not currently available on the server	959	49.947	44	2
D19	Waveform	Waveform (Version 2, 1988)	5000	75.000	40	3
D20	Winequality-red	Wine Quality, 2009, red wine type	1599	74.796	11	6
D21	Winequality-white	Wine Quality, 2009, white wine type	4898	74.948	11	7
D22	Yeast	Yeast, 1996	1484	74.932	8	10
D23	d50r5g0	Five relevant attributes	500	75.000	50	2
D24	d50r5g5	Five relevant attributes and five noisy ones	500	75.000	50	2
D25	d50r10g0	Ten relevant attributes	500	75.000	50	2
D26	d50r10g5	Ten relevant attributes and five noisy ones	500	75.000	50	2
D27	d100r5g0	Five relevant attributes	500	75.000	100	2
D28	d100r5g5	Five relevant attributes and five noisy ones	500	75.000	100	2
D29	d100r10g0	Ten relevant attributes	500	75.000	100	2
D30	d100r10g5	Ten relevant attributes and five noisy ones	500	75.000	100	2
D31	d100r15g0	Fifteen relevant attributes	500	75.000	100	2
D32	d100r15g5	Fifteen relevant attributes and five noisy ones	500	75.000	100	2
D33	d100r20g0	Twenty attributes	500	75.000	100	2
D34	d100r20g5	Twenty attributes and five noisy ones	500	75.000	100	2
Average			1202.9	74.3	49.9	3.4

Table 2
List of methods employed in the extensive experimentation with and without feature selection.

Feature selector name	Type of method	Ranking method	Subset evaluation	Abb. Name
–	–	None	None	F0
spBI _CFS	Filter	SOAP	CFS	F1
cfBI _CFS	Filter	CFS	CFS	F2
spBI _CNS	Filter	SOAP	CNS	F3
cnBI _CNS	Filter	CNS	CNS	F4
FCBF	Filter	SU	SU	F5
nbBI _NB	Semi-wrapper	NB	NB	F6
cfBI _NB	Semi-wrapper	CFS	NB	F7

training set) and other with the remaining quarter (the testing set). We have specified the exact training percentage in order to facilitate reproducibility. Hereinafter, we only mention results related to the testing set. Thus, in order to assess a classifier we train the model with the training set, then we evaluate the achieved model with the unseen (testing) set. The exception is the computational cost which has been computed during the training phase and is the most time-consuming activity in supervised machine learning tasks. Initial pre-processing even before feature selection has been minimal in the sense that only the missing values have been replaced by the mean or the mode within the same class depending on whether the attribute is numerical or categorical.

4.2. Filters and semi-wrappers

Table 2 shows the selection methods used in the experiments, except for the first row which corresponds to the complete data set. The first column indicates the name of the selector. As explained in previous paragraphs, all of the selectors, except the one that appears in the sixth row (FCBF), correspond to different variants of the BIRS (BI) algorithm, so that the letters that precede it affect the metric with the ordering of the attributes in the initial phase is carried out (this is reaffirmed in the second column), while the letters that appear behind BI refer to the function of subset evaluation applied in the second phase of BIRS (this is reaffirmed in the third column). The last column shows the abbreviations with which we refer to the different selectors in the next sections, from F0 to F7, denoting that F0 corresponds to the absence of a selection method. Some preliminary own works assessed BIRS in the context of neural networks [50,51].

With regards to the problem indicated in the first cell within the row, every row in Table 3 shows the number of features of the original training set (refer to column labelled F0) and those which

Table 3
Original and selected features. Scalar values and reduction percentage.

Data set	F0	F1	F2	F3	F4	F5	F6	F7	Red_F1	Red_F2	Red_F3	Red_F4	Red_F5	Red_F6	Red_F7
D1	129	33	22	20	19	2	7	14	74.42	82.95	84.50	85.27	98.45	94.57	89.15
D2	9	4	4	2	2	3	2	4	55.56	55.56	77.78	77.78	66.67	77.78	55.56
D3	6	1	1	5	6	3	2	2	83.33	83.33	16.67	0.00	50.00	66.67	66.67
D4	13	7	7	8	9	6	6	6	46.15	46.15	38.46	30.77	53.85	53.85	53.85
D5	19	10	10	11	5	6	2	2	47.37	47.37	42.11	73.68	68.42	89.47	89.47
D6	33	17	13	12	9	6	9	9	48.48	60.61	63.64	72.73	81.82	72.73	72.73
D7	29	7	6	5	5	8	4	4	75.86	79.31	82.76	82.76	72.41	86.21	86.21
D8	43	10	3	12	11	3	6	6	76.74	93.02	72.09	74.42	93.02	86.05	86.05
D9	19	9	5	11	11	3	4	6	52.63	73.68	42.11	42.11	84.21	78.95	68.42
D10	21	2	2	9	7	2	4	7	90.48	90.48	57.14	66.67	90.48	80.95	66.67
D11	23	5	5	7	6	4	2	2	78.26	78.26	69.57	73.91	82.61	91.30	91.30
D12	21	3	3	4	4	4	4	4	85.71	85.71	80.95	80.95	80.95	80.95	80.95
D13	8	3	3	4	5	4	3	2	62.50	62.50	50.00	37.50	50.00	62.50	75.00
D14	47	9	9	8	8	13	13	19	80.85	80.85	82.98	82.98	72.34	72.34	59.57
D15	58	7	7	8	7	11	2	2	87.93	87.93	86.21	87.93	81.03	96.55	96.55
D16	82	44	23	35	22	18	29	27	46.34	71.95	57.32	73.17	78.05	64.63	67.07
D17	23	3	3	4	3	6	6	8	86.96	86.96	82.61	86.96	73.91	73.91	65.22
D18	44	14	8	16	19	2	6	11	68.18	81.82	63.64	56.82	95.45	86.36	75.00
D19	40	14	14	15	15	5	14	15	65.00	65.00	62.50	62.50	87.50	65.00	62.50
D20	11	5	5	8	8	4	5	5	54.55	54.55	27.27	27.27	63.64	54.55	54.55
D21	11	6	6	10	10	4	2	5	45.45	45.45	9.09	9.09	63.64	81.82	54.55
D22	8	5	4	7	7	6	7	7	37.50	50.00	12.50	12.50	25.00	12.50	12.50
D23	50	5	5	5	5	5	9	8	90.00	90.00	90.00	90.00	90.00	82.00	84.00
D24	50	4	4	3	2	4	9	9	92.00	92.00	94.00	96.00	92.00	82.00	82.00
D25	50	5	5	5	5	5	16	16	90.00	90.00	90.00	90.00	90.00	68.00	68.00
D26	50	7	6	7	6	6	13	15	86.00	88.00	86.00	88.00	88.00	74.00	70.00
D27	100	5	5	5	5	5	13	8	95.00	95.00	95.00	95.00	95.00	87.00	92.00
D28	100	4	4	3	2	4	14	12	96.00	96.00	97.00	98.00	96.00	86.00	88.00
D29	100	5	5	5	5	5	13	21	95.00	95.00	95.00	95.00	95.00	87.00	79.00
D30	100	7	6	7	6	6	13	13	93.00	94.00	93.00	94.00	94.00	87.00	87.00
D31	100	7	6	7	6	4	10	20	93.00	94.00	93.00	94.00	96.00	90.00	80.00
D32	100	7	6	6	6	6	23	19	93.00	94.00	94.00	94.00	94.00	77.00	81.00
D33	100	6	6	6	6	6	30	25	94.00	94.00	94.00	94.00	94.00	70.00	75.00
D34	100	8	7	8	7	7	19	25	92.00	93.00	92.00	93.00	93.00	81.00	75.00
Average	49.9	8.5	6.7	8.5	7.6	5.5	9.4	10.5	75.3	78.8	69.8	71.1	80.3	76.5	73.3

have been obtained, utilising only the training set, with seven feature subset selection methods (see columns labelled F1–F7) along with the reduction percentage in the feature space of each feature subset selection procedure compared to the original data set. The last row reports the average number of features and the reduction percentage of the test-bed for each trialled method on this paper. The reduction percentage of the number of features is defined by Eq. (1).

$$Reduction\_of\_features(\%) = \left(1 - \frac{Features(Fi)}{Features(F0)}\right)100; \;\; i = 1, \ldots, 7 \tag{1}$$

where i is the index of filter or feature selection procedure and *Features(j)* represents the number of features of a given data set with method j.

In all registers, filter-based feature selection approaches successfully and outstandingly reduce the data dimensionality by choosing, on average, far less than a quarter of the number of features originally available on the data set. F5 and F2 achieve means reductions of 80.3% and 78.8%, respectively, which are the highest overall average values obtained. The filter F7 keeps the highest number of features from the full data set.

4.3. Classifiers

All the previous settings of the problems have been assessed with two neural network-based classifiers, RBFNN and MLP, which are the core methods for the research conducted and for this reason we intend to shed light on the appropriateness of seven feature subset selection approaches operating jointly. Moreover, we have considered two powerful methods: the highly popular decision tree C4.5 and Support Vector Machines (SVM). We have trialled with the available implementations in Weka [52] version 3.7.10 of the aforesaid algorithms which are named MultilayerPerceptron (MLP), RBFNetwork (RBF), J48 and SMO.

Neural networks performance is highly dependent on the parameter setting. We have conducted a preliminary experimentation keeping in mind the training set to analyse the behaviour of some parameter values. Firstly, in MLP the momentum and the learning rate are two basic parameters; in a previous contribution we did a grid search for them in a good number of problems and we determined that the best values are 0.3 and 0.2, respectively [53]. Going further, the training time (number of epochs) and the number of nodes in the hidden layer are of the utmost importance; for the first parameter we managed three options (250, 500 and 1000 following similar paths as described in [54]) and in accordance with the initial experiments using only the training data we set it to 500; additionally, for the number of nodes in the hidden layer we have taken into account two possible values, (attributes+classes)/2 and (attributes+classes), ultimately opting for the former although the latter could be very convenient for problems with just a few attributes such as the classical Iris data set. Secondly, for RBF the main parameter is the number of clusters. We tried with values from 2 to 5; the preliminary experimentation led us to choose 2 as the most suitable value. Finally, for J48 and SMO we would like to remark that the behaviour of the former is excellent with the default values, with nothing new being found and for the latter it is very important to stress that we have kept the kernel as the polynomial kernel (PolyKernel) with its default values.

5. Results

We have run the stochastic algorithms (RBF and MLP) using the training and testing sets 30 times just to smooth the results and to reduce bias as much as possible. Therefore, averages of the results on the forthcoming tables have been taken and the mean (Avg) and the standard deviation (SD) are shown to reflect well on the stability of the models achieved. Bearing in mind that there are 34 data sets, one baseline method and 7 feature selection methods, the total number of runs for each neural classifier is about 8000.

5.1. Global results in the test-bed

5.1.1. Results and statistical analysis with RBFNN

Table 4 reports the results obtained with RBFNN for every problem given the original data set (column F0) and the reduced data sets (columns labelled F1, F2,... F7). The top part shows quantitative information, measured in test accuracy illustrating the mean and SD. For the mean higher is better, whereas for the SD lower is better. The bottom part depicts qualitative information based on statistical tests; generally speaking, for this zone lower is better (e.g. rank and T). The results with F0 are taken as the baseline approach. Columns five to eleven represent the results with the application of feature selection. On average, the best data preparation method seems to be F7 for two reasons: the mean and the number of wins. We need to swap to the qualitative perspective to get more substantial conclusions. An Iman-Davenport test is utilised to prove the existence of significant differences according the ranks of the feature selection approaches; a low value is indicative of a good performance and a high value indicative of a poor performance. Since the null-hypothesis is rejected, the performance of every pair of classifiers is not significantly different and we cannot

Table 4
RBFNN test accuracy results with the whole test-bed.

Data set	Problem	Accuracy	F0	F1	F2	F3	F4	F5	F6	F7
D1	Batch	Avg	65.47	68.88	70.01	65.19	66.79	69.67	76.77	75.25
		SD	0.91	1.20	0.71	0.60	15.49	0.10	0.67	1.03
D2	Breast	Avg	68.78	67.46	67.46	69.01	69.01	67.65	65.49	65.77
		SD	1.57	1.13	1.13	0.00	0.00	1.50	0.72	1.70
D3	Column	Avg	81.15	79.62	79.62	83.93	81.15	80.77	82.86	82.86
		SD	1.47	0.62	0.62	1.67	1.47	0.00	0.63	0.63
D4	Heart	Avg	78.53	78.24	78.24	75.39	77.60	75.92	74.71	74.71
		SD	1.92	1.98	1.98	1.22	1.43	1.57	2.02	2.02
D5	Hepatitis	Avg	89.30	89.30	89.30	89.91	88.42	89.53	88.42	88.42
		SD	2.29	2.76	2.76	1.84	1.64	2.40	1.31	1.31
D6	Ionos	Avg	92.46	95.49	94.73	94.51	93.39	94.09	93.11	93.56
		SD	0.70	0.21	0.63	0.79	1.84	0.63	0.29	1.56
D7	Labor	Avg	71.67	71.43	85.71	64.29	64.29	67.56	71.43	71.43
		SD	1.30	0.00	0.00	0.00	0.00	4.88	0.00	0.00
D8	Leaves	Avg	67.48	65.19	67.56	70.52	67.45	67.41	77.56	64.74
		SD	2.37	3.21	5.79	2.97	5.36	2.63	3.42	2.84
D9	Messidor	Avg	59.92	60.94	60.17	60.07	63.68	61.50	59.06	59.25
		SD	0.51	1.59	0.74	0.00	5.09	0.36	0.83	1.25
D10	OBS	Avg	69.32	74.99	74.99	70.21	71.20	74.68	70.55	74.11
		SD	1.68	2.45	2.45	0.59	0.95	3.25	1.46	2.61
D11	Parkinsons	Avg	70.27	77.55	77.55	74.56	73.47	81.42	81.63	83.67
		SD	1.67	0.00	0.00	1.17	0.00	3.07	0.00	0.00
D12	Pasture	Avg	64.81	74.44	74.44	91.48	91.48	70.37	73.70	91.11
		SD	11.70	10.98	10.98	9.54	9.54	7.90	6.83	8.95
D13	Pima	Avg	77.34	79.17	79.17	77.50	75.64	79.29	79.10	73.32
		SD	2.17	0.53	0.53	0.52	1.07	2.29	0.53	1.37
D14	Pollen	Avg	91.73	91.89	91.21	89.72	89.72	92.03	89.66	93.27
		SD	0.29	0.44	0.27	0.32	0.32	1.19	0.64	0.42
D15	Promoter	Avg	79.36	83.46	83.46	76.03	85.00	79.01	80.77	80.77
		SD	5.30	2.70	2.70	4.70	4.33	4.20	0.00	0.00
D16	Soybean	Avg	93.84	93.47	93.20	92.81	89.61	91.38	93.24	94.17
		SD	0.92	0.99	0.80	0.88	1.94	0.81	0.60	0.19
D17	Squash	Avg	80.77	85.64	85.64	82.05	75.38	80.77	70.26	80.00
		SD	6.92	3.90	3.90	5.83	5.85	4.76	5.62	4.78
D18	Tokyo	Avg	89.56	88.76	87.94	88.35	89.65	89.65	89.49	91.45
		SD	1.25	0.95	2.31	1.48	2.06	2.06	0.11	0.52
D19	Waveform	Avg	82.14	82.24	82.24	82.55	82.22	76.89	82.63	82.55
		SD	0.08	0.13	0.13	0.08	0.05	1.42	0.03	0.07
D20	Winequality-red	Avg	57.11	59.00	57.53	59.19	59.19	58.88	57.59	57.59
		SD	2.28	0.74	1.88	0.83	0.83	2.55	1.85	1.85
D21	Winequality-white	Avg	48.04	51.39	51.39	48.90	48.90	51.08	52.15	50.85
		SD	0.37	0.70	0.70	0.56	0.56	0.44	0.56	0.72
D22	Yeast	Avg	58.33	58.41	54.97	58.90	58.90	58.48	54.97	54.97
		SD	1.09	1.04	1.22	1.34	1.34	2.61	1.22	1.22
D23	d50r5g0	Avg	82.27	86.37	86.37	86.37	86.37	86.37	85.39	83.12
		SD	2.57	2.24	2.24	2.24	2.24	2.24	2.17	2.73
D24	d50r5g5	Avg	78.60	80.83	80.83	75.78	67.56	80.83	83.25	81.55
		SD	2.72	2.57	2.57	2.07	2.13	2.57	2.80	2.55
D25	d50r10g0	Avg	73.64	72.47	72.47	72.47	72.47	72.47	79.01	79.52
		SD	2.75	2.01	2.01	2.01	2.01	2.01	2.31	2.11
D26	d50r10g5	Avg	76.88	68.63	66.40	68.63	66.40	66.40	75.41	73.89
		SD	2.06	3.13	2.18	3.13	2.18	2.18	3.08	2.65
D27	d100r5g0	Avg	78.25	86.37	86.37	86.37	86.37	86.37	85.12	86.88

(continued on next page)

Table 4 (*continued*)

Data set	Problem	Accuracy	F0	F1	F2	F3	F4	F5	F6	F7
		SD	2.98	2.24	2.24	2.24	2.24	2.24	2.20	2.46
D28	d100r5g5	Avg	76.61	80.83	80.83	75.78	67.56	80.83	81.47	83.12
		SD	2.73	2.57	2.57	2.07	2.13	2.57	2.96	2.92
D29	d100r10g0	Avg	73.59	72.47	72.47	72.47	72.47	72.47	77.81	82.77
		SD	3.04	2.01	2.01	2.01	2.01	2.01	2.42	2.23
D30	d100r10g5	Avg	74.63	70.04	66.40	70.04	66.40	66.40	75.41	70.72
		SD	2.44	3.32	2.18	3.32	2.18	2.18	3.08	2.66
D31	d100r15g0	Avg	77.36	72.28	71.86	72.28	66.62	71.86	67.65	81.57
		SD	2.54	2.17	2.69	2.17	2.66	2.69	2.68	2.96
D32	d100r15g5	Avg	75.84	67.01	67.83	63.67	67.83	67.83	73.44	70.51
		SD	2.70	2.41	3.06	2.52	3.06	3.06	2.28	3.16
D33	d100r20g0	Avg	74.04	68.14	68.14	68.14	68.14	68.14	77.36	69.73
		SD	2.80	2.34	2.34	2.34	2.34	2.34	2.76	2.11
D34	d100r20g5	Avg	76.45	64.52	65.43	64.52	65.43	65.43	68.35	75.23
		SD	2.59	2.70	2.93	2.70	2.93	2.93	2.93	2.94
Average			75.16	75.50	75.65	74.75	73.99	74.81	76.32	77.13
Wins				18	18	17	15	17	20	20
Mean rank.			4.75	4.19	4.44	4.82	5.10	4.63	4.26	3.79
Pairwise comparison										
F0 vs F7			$T = 220$				F0 = F7			
F0 vs F1			$T = 264$				F0 = F1			
F4 vs F7			$T = 139$ (*)				F7 > F4 (*)			
Compared methods			Statistical test results				Statistical conclusion			

Critical value: 201.
* Statistically significant difference with $\alpha = 0.05$.

apply a post-hoc test. We apply other types of non-parametric tests to add more discoveries to the experiments. Concretely, a pairwise comparison following a Wilcoxon signed-ranks test is conducted. We have taken as candidates for the statistical test the baseline method and the two best filters according to their ranks; then the best and worst filters are compared. Since there are 34 data sets, the T value at $\alpha = 0.05$ should be less than or equal to 201 (critical value) to reject the null hypothesis. The condition is met only in the comparison between F4 and F7 and therefore the null hypothesis is rejected. It means that F7 is statistically better than F4. The other comparisons state that F7 and F1, individually, get at least similar results to F0 from a statistical point of view.

5.1.2. Results and statistical analysis with MLP

Table 5 depicts the results concerning the MLP classifier. From the quantitative perspective, F7 appears to be the best although the margin is closer than in the RBF (as depicted in Table 4, value 3.79 compared to 4.75) classifier because, now, the average differences (3.63 compared to 4.07) and the number of times that F7 performs better than F0 is lower (18). Moreover, there are two methods with a ranking over 5 and we need to study them carefully. As a practical matter, we have now done five statistical comparisons which are performed between F4 and F5 compared to F0 and F7 and also the pairwise confrontation between the baseline method (F0) and the approach which seems to be the best (F7). The critical value is again 201 because compared to the previous section, the only element that we have changed is the supervised machine learning. F7 does not significantly overcome F0. F4 and F5 should not be used with likelihood of success because F0 is significantly better; another issue could be the computational cost. As the reader may have thought, F7 enhances F4 and F5 with significant differences. The variety of approaches better than F0 has been outstandingly reduced to one

Table 5
MLP test accuracy results with the whole test-bed.

Data set	Problem	Accuracy	F0	F1	F2	F3	F4	F5	F6	F7
D1	Batch	Avg	92.11	96.64	94.25	92.75	85.65	69.07	90.19	96.45
		SD	5.18	1.38	1.60	1.45	4.41	1.81	1.62	1.04
D2	Breast	Avg	61.13	69.01	69.01	69.01	69.01	69.53	64.79	67.32
		SD	3.87	1.81	1.81	0.00	0.00	1.46	0.00	1.67
D3	Column	Avg	82.44	80.68	80.68	83.50	82.44	82.69	78.25	78.25
		SD	2.02	2.28	2.28	2.35	2.02	3.85	2.64	2.64
D4	Heart	Avg	74.80	72.65	72.65	73.14	74.85	74.85	73.33	73.33
		SD	2.40	2.14	2.14	1.81	2.57	2.63	1.38	1.38
D5	Hepatitis	Avg	85.00	87.28	87.28	86.75	84.21	87.72	86.84	86.84
		SD	2.60	3.85	3.85	2.01	0.00	1.44	0.00	0.00
D6	Ionos	Avg	88.94	92.01	89.85	92.12	92.84	89.24	91.29	95.15
		SD	1.37	1.21	2.35	1.23	1.58	1.73	1.50	0.79
D7	Labor	Avg	69.52	64.29	78.57	78.57	78.57	71.43	71.43	71.43
		SD	3.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D8	Leaves	Avg	71.48	70.89	68.07	68.89	72.89	66.96	72.44	63.63
		SD	3.87	4.68	5.61	2.86	4.34	2.59	2.71	2.64
D9	Messidor	Avg	72.53	71.82	71.40	70.28	71.27	61.27	60.66	71.85
		SD	2.24	1.33	0.89	1.67	2.08	0.93	1.31	1.72
D10	OBS	Avg	81.00	75.50	75.50	79.49	79.55	72.64	76.20	78.19

(*continued on next page*)

Table 5 (*continued*)

Data set	Problem	Accuracy	F0	F1	F2	F3	F4	F5	F6	F7
		SD	3.02	2.27	2.27	2.52	2.03	3.52	2.93	2.15
D11	Parkinsons	Avg	77.62	81.56	81.56	75.92	75.65	84.83	80.61	80.61
		SD	0.37	0.37	0.37	2.48	2.62	1.58	1.29	2.32
D12	Pasture	Avg	66.67	72.22	72.22	77.78	77.78	56.67	88.89	88.89
		SD	0.00	6.36	6.36	0.00	0.00	3.39	0.00	0.00
D13	Pima	Avg	76.74	78.18	78.18	74.27	76.91	79.03	76.04	78.25
		SD	1.60	1.23	1.23	1.73	1.43	1.49	2.25	1.04
D14	Pollen	Avg	96.39	91.67	91.59	88.74	88.74	92.84	93.51	94.65
		SD	0.69	0.82	0.86	1.75	1.75	0.92	1.13	0.80
D15	Promoter	Avg	86.03	84.49	84.49	65.00	75.51	78.46	80.77	80.77
		SD	2.14	0.70	0.70	2.34	1.89	2.39	0.00	0.00
D16	Soybean	Avg	92.87	92.13	90.72	92.44	88.93	88.78	92.69	92.27
		SD	1.14	1.19	1.10	0.73	1.70	1.47	1.14	1.16
D17	Squash	Avg	80.26	76.92	76.92	84.62	80.51	76.92	81.79	94.36
		SD	4.82	0.00	0.00	0.00	3.90	0.00	5.14	3.46
D18	Tokyo	Avg	91.37	91.92	91.13	91.44	90.05	90.05	90.74	92.15
		SD	0.78	0.78	0.54	0.89	0.71	0.71	0.80	0.46
D19	Waveform	Avg	80.41	83.24	83.24	83.42	83.13	77.58	82.81	82.64
		SD	0.77	1.79	1.79	1.18	1.13	0.99	1.35	1.17
D20	Winequality-red	Avg	56.22	59.45	58.95	57.04	57.04	59.61	59.04	59.04
		SD	1.63	1.52	2.02	1.61	1.61	1.19	1.07	1.07
D21	Winequality-white	Avg	52.21	53.02	53.02	52.65	52.65	51.40	51.80	51.41
		SD	1.74	1.86	1.86	1.98	1.98	1.68	1.63	1.91
D22	Yeast	Avg	59.84	60.10	55.11	60.06	60.06	59.01	55.11	55.11
		SD	2.18	1.50	1.85	2.00	2.00	1.60	1.85	1.85
D23	d50r5g0	Avg	96.45	99.15	99.15	99.15	99.15	99.15	99.04	98.93
		SD	2.19	0.59	0.59	0.59	0.59	0.59	0.61	0.38
D24	d50r5g5	Avg	88.75	84.24	84.24	75.71	68.48	84.24	92.08	89.55
		SD	1.60	0.79	0.79	2.44	2.76	0.79	1.14	1.89
D25	d50r10g0	Avg	88.72	76.69	76.69	76.69	76.69	76.69	79.84	80.35
		SD	0.71	2.16	2.16	2.16	2.16	2.16	2.64	1.73
D26	d50r10g5	Avg	77.79	70.24	68.32	70.24	68.32	68.32	84.64	77.33
		SD	1.44	2.84	2.76	2.84	2.76	2.76	1.85	2.39
D27	d100r5g0	Avg	90.13	99.15	99.15	99.15	99.15	99.15	97.63	98.75
		SD	0.85	0.59	0.59	0.59	0.59	0.59	0.61	0.72
D28	d100r5g5	Avg	83.01	84.24	84.24	75.71	68.48	84.24	88.67	88.32
		SD	2.77	0.79	0.79	2.44	2.76	0.79	1.84	2.04
D29	d100r10g0	Avg	85.25	76.69	76.69	76.69	76.69	76.69	77.23	95.36
		SD	1.79	2.16	2.16	2.16	2.16	2.16	2.11	1.87
D30	d100r10g5	Avg	78.67	69.65	68.32	69.65	68.32	68.32	84.64	69.71
		SD	1.52	2.12	2.76	2.12	2.76	2.76	1.85	2.59
D31	d100r15g0	Avg	89.81	72.75	70.72	72.75	70.72	70.72	66.80	85.71
		SD	0.59	2.56	2.29	2.56	2.29	2.29	2.86	1.98
D32	d100r15g5	Avg	80.43	65.44	62.64	62.53	62.64	62.64	77.87	70.24
		SD	1.45	1.69	2.75	1.91	2.75	2.75	2.25	3.44
D33	d100r20g0	Avg	85.31	67.60	67.60	67.60	67.60	67.60	84.29	72.77
		SD	0.88	2.97	2.97	2.97	2.97	2.97	1.65	2.51
D34	d100r20g5	Avg	78.77	63.52	61.57	63.52	61.57	61.57	70.21	81.55
		SD	1.97	3.89	3.47	3.89	3.47	3.47	2.50	1.87
Average			79.96	77.50	77.17	76.68	76.06	75.29	79.48	80.62
Wins				15	14	15	14	12	16	18
Mean rank.			4.07	4.10	4.90	4.66	5.07	5.26	4.28	3.63
Pairwise comparison										
F0 vs F7			$T = 267$				F0 = F7			
F5 vs F7			$T = 105$ (*)				F7 > F5 (*)			
F4 vs F7			$T = 124$ (*)				F7 > F4 (*)			
F5 vs F0			$T = 129$ (*)				F0 > F5 (*)			
F4 vs F0			$T = 178$ (*)				F0 > F4 (*)			
Compared methods			Statistical test results				Statistical conclusion			

Critical value: 201.

* Statistically significant difference with $\alpha = 0.05$.

method in terms of rank. Clearly, it may suggest seeking new ways of feature selection for the MLP classifier.

5.2. Analysis of the results on real-world problems

This section aims at extracting some extra conclusions focusing only on real-world problems. In particular, we carry out additional comparisons for the most promising feature selection methods according to the global results from both RBFNN and MLP

5.2.1. Results with RBFNN

Table 6 shows the results regarding real-world problems in the context of RBFNN. The top part indicates the information related to a descriptive analysis of the results. As usual, the bottom part reports qualitative information. For the section on global results we observe two very strong candidates as feature selection approaches. Therefore, we must undergo tests at least for F1 and F7 due to its superior performance within the feature selection arena in the RBFNN scope. There are three methods with a performance around 2 percentage units higher than F0 (baseline method). There

Table 6
Summary of RBFNN test accuracy results and statistical test results on real-world problems.

Data sets	Problems	Average accuracy							
		F0	F1	F2	F3	F4	F5	F6	F7
D1-D22	Batch, Breast, Column, Heart, Hepatitis, Ionos, Labor, Leaves, Messidor, OBS, Parkinsons, Pasture, Pima, Pollen, Promoter, Soybean, Squash, Tokyo, Waveform, Winequality-red, Winequality-white, Yeast	74.43	76.22	76.66	75.68	75.55	75.36	75.69	76.54
Pairwise comparison									
F0 vs F7		$T = 98$				F0 = F7			
F0 vs F1		$T = 61^{(*)}$				F1 > F0$^{(*)}$			
F0 vs F2		$T = 65^{(*)}$				F2 > F0$^{(*)}$			
Compared methods		Statistical test results				Statistical conclusion			

Critical value: 75.
*Statistically significant difference with $\alpha = 0.05$.

are 22 data sets and then the critical value at $\alpha = 0.05$ is 75. F1 and F2 are significantly more accurate than F0. F7 has the second best mean although there are not significant differences with F0. F1 is not as good as F7 on average, but is statistically preferable to F0.

5.2.1.1. Computational cost. The comparison between the baseline method and the filter-based selection methods using RBFNN is completed by means of a computational cost analysis. Table 7 reports the training time results concerning the average computational cost per run measured in seconds (s). Experiments have been run in a server equipped with an Intel Xeon E5-2630 v3 processor at 2.4 GHz. Columns five to eleven depict the training time, although the time to apply the feature selection has not been taken into account because it is insignificant (it may be no more than 10 s) and must only be done once, and not for every run. The last row contains the sum of the average values reported in the column. Obviously, the weight of the Yeast problem is huge. Since F1 and F2 are the significant best methods with regards to accuracy, we only stress them. F1 and F2 are around 3 or 4 times faster than F0.

5.2.2. Results with MLP

Table 8 shows the results obtained on real-world problems coming mostly from the UCI repository with an MLP neural approach. The best enhancement with feature selection is less than 2 percentage units. From a purely descriptive analysis of the results, F7 seems the best option. We also picked up as a candidate from Section 5.1.2 the feature selection approach F7. Moving to the qualitative field of the results with MLP into real-worlds problem, the critical value is 75 as aforementioned and T is equal to 97. It means that there are not significant differences between F0 and F7. In practical terms, F7 could be safely applied because the results are not going to be worse and the number of features is considerably reduced in comparison to the original feature space for every problem.

5.2.2.1. Computational cost. Table 9 depicts the elapsed time to train the neural network following the MLP algorithm. It has been averaged on 30 runs. It is worth mentioning that the sum of every average time is around one thousand seconds for all the problems and Batch needs about 940 s. The speedup obtained with feature selection is at least 8 times in the worst case (F1) and around 30 times in the best case (F7). Continuing in the line of the comments about the best filter in terms of accuracy, we can assert that the best feature selection method for MLP is also one of the top 3 fastest.

5.3. Analysis of the results on synthetic data sets

5.3.1. Results with RBFNN

Table 10 reports the results regarding the application of RBFNN to synthetic data sets. F1 and F7 have been promising in certain contexts of the previous parts of this research such as the scenario with the whole test-bed and with real-world problems, both for RBFNN. We now move on to the significance angle. Since there 12 data sets, the critical value is 17 at $\alpha = 0.05$ significance level. It means that F1 is the closest method regarding T although its value is greater than the critical value and we can assert that F1 and F0 exhibit a similar performance according to the statistical tests.

5.3.1.1. Computational cost. The total RBFNN learning time with the original twelve data sets was 7570 ms. This time includes the generation of the thirty models, one for each execution with a different seed for each of the data sets, reaching an average of 21.03 ms. However, the average of the same learning with the reduced data sets is 6.33 ms, slightly less than a third of the previous time.

5.3.2. Results with MLP

Table 11 shows the results obtained by MLP classifiers on synthetic data sets. The situation is very complex because not many feature selection methods are able to maintain the performance of the MLP models, bearing in mind the full feature space. At the same time, it is proof that the chosen data sets are not very easy for feature selection and are important challenges. The first sights are for F7, especially to the behaviour in the previous scenarios. From previous comparisons with MLP neural networks, we also noticed that the single strong candidate for this type of feed-forward neural networks is F7. Then, we conduct now comparisons between F0 and F7. T is again higher than the critical value (17) and the conclusion drawn is that F7 is not significantly better than F0. The good news is that F7 is a strong candidate for most of the situations where feature selection is a very convenient or even necessary initial step.

5.3.2.1. Computational cost. In this case, the total MLP learning time with the twelve original data sets was just over one hour. As with RBFNN, this time includes the generation of the thirty models, one for each execution with a different seed for each of the data sets, reaching an average of 10.35 s. However, the average of

Table 7
Average RBFNN training time measured in seconds on real-worlds problems.

Data set	Problem	t (s)	F0	F1	F2	F3	F4	F5	F6	F7
D1	Batch	Avg	15.0579	9.5955	6.9001	6.5232	6.3569	3.7870	4.6443	4.3166
		SD	16.7654	1.6754	1.2056	2.7809	1.8018	0.3092	1.0197	0.9140
D2	Breast	Avg	0.0165	0.0047	0.0047	0.0048	0.0048	0.0038	0.0093	0.0043
		SD	0.0186	0.0006	0.0006	0.0044	0.0044	0.0006	0.0104	0.0050
D3	Column	Avg	0.0141	0.0051	0.0051	0.0064	0.0141	0.0059	0.0065	0.0065
		SD	0.0210	0.0007	0.0007	0.0009	0.0210	0.0008	0.0007	0.0007
D4	Heart	Avg	0.0485	0.0061	0.0061	0.0059	0.0059	0.0185	0.0109	0.0109
		SD	0.0699	0.0006	0.0006	0.0019	0.0009	0.0198	0.0123	0.0123
D5	Hepatitis	Avg	0.0303	0.0033	0.0033	0.0043	0.0022	0.0041	0.0080	0.0080
		SD	0.0434	0.0021	0.0021	0.0077	0.0004	0.0040	0.0090	0.0090
D6	Ionos	Avg	0.0827	0.0135	0.0109	0.0083	0.0354	0.0078	0.0099	0.0073
		SD	0.1199	0.0054	0.0101	0.0079	0.0392	0.0080	0.0111	0.0079
D7	Labor	Avg	0.0174	0.0032	0.0046	0.0050	0.0050	0.0097	0.0026	0.0026
		SD	0.0242	0.0022	0.0042	0.0112	0.0112	0.0084	0.0059	0.0059
D8	Leaves	Avg	0.0629	0.0068	0.0063	0.0057	0.0099	0.0042	0.0083	0.0058
		SD	0.0905	0.0079	0.0078	0.0076	0.0077	0.0070	0.0079	0.0012
D9	Messidor	Avg	0.3713	0.0249	0.0181	0.0228	0.0427	0.0132	0.0162	0.0159
		SD	0.5500	0.0097	0.0012	0.0024	0.0515	0.0012	0.0037	0.0023
D10	OBS	Avg	0.9812	0.3162	0.3162	0.3581	0.2733	0.4864	0.2639	0.2749
		SD	0.3421	0.1835	0.1835	0.0998	0.1207	0.1913	0.1005	0.1205
D11	Parkinsons	Avg	0.0285	0.0070	0.0070	0.0049	0.0047	0.0106	0.0026	0.0032
		SD	0.0393	0.0119	0.0119	0.0017	0.0023	0.0035	0.0059	0.0064
D12	Pasture	Avg	0.0042	0.0026	0.0026	0.0019	0.0019	0.0025	0.0026	0.0031
		SD	0.0070	0.0011	0.0011	0.0007	0.0007	0.0011	0.0011	0.0063
D13	Pima	Avg	0.0960	0.0105	0.0105	0.0093	0.0138	0.0191	0.0098	0.0086
		SD	0.1356	0.0010	0.0010	0.0008	0.0084	0.0098	0.0076	0.0063
D14	Pollen	Avg	5.8916	1.4760	1.2827	1.6599	1.6599	2.4699	2.5765	1.3287
		SD	3.3299	0.3615	0.2844	0.3144	0.3144	1.1121	0.6758	0.3378
D15	Promoter	Avg	0.0318	0.0023	0.0023	0.0020	0.0019	0.0084	0.0021	0.0021
		SD	0.0467	0.0005	0.0005	0.0005	0.0004	0.0010	0.0054	0.0054
D16	Soybean	Avg	26.8724	11.5601	16.6117	16.1181	10.4677	16.4179	10.1938	14.6807
		SD	23.4924	5.8328	8.8640	6.6383	6.7157	6.0535	4.7070	5.9496
D17	Squash	Avg	0.0606	0.0045	0.0045	0.0053	0.0047	0.0110	0.0088	0.0047
		SD	0.0892	0.0012	0.0012	0.0019	0.0020	0.0118	0.0098	0.0074
D18	Tokyo	Avg	0.3935	0.0173	0.0168	0.0152	0.0261	0.0161	0.0125	0.0141
		SD	0.5941	0.0018	0.0047	0.0026	0.0106	0.0011	0.0064	0.0171
D19	Waveform	Avg	1.0663	0.4367	0.4367	0.4729	0.4596	0.6779	0.3330	0.4006
		SD	1.3758	0.0583	0.0583	0.0168	0.0576	0.0852	0.0185	0.0308
D20	Winequality-red	Avg	0.9104	0.6398	0.5491	0.5862	0.5862	0.7121	0.6336	0.6336
		SD	0.6189	0.2073	0.1615	0.2157	0.2157	0.2500	0.2150	0.2150
D21	Winequality-white	Avg	6.5734	6.3958	6.3958	4.0289	4.0289	2.7704	3.2584	3.1167
		SD	3.8929	2.0864	2.0864	2.5913	2.5913	0.9692	1.1854	0.8499
D22	Yeast	Avg	691.2589	208.5572	149.2557	548.5572	548.1572	275.1067	548.5572	548.5572
		SD	413.7464	181.7115	135.1397	481.7115	481.1115	217.6527	481.7115	481.7115
Sum			749.8702	239.0892	181.8508	578.4064	572.1630	302.5630	570.5707	573.4061

Table 8
Summary of MLP test accuracy results and statistical test results on real-world problems.

Data sets	Problems	Average Accuracy							
		F0	F1	F2	F3	F4	F5	F6	F7
D1-D22	Batch, Breast, Column, Heart, Hepatitis, Ionos, Labor, Leaves, Messidor, OBS, Parkinsons, Pasture, Pima, Pollen, Promoter, Soybean, Squash, Tokyo, Waveform, Winequality-red, Winequality-white, Yeast	77.07	77.53	77.47	77.18	77.19	74.57	77.24	78.75
Pairwise comparison									
F0 vs F7			$T = 97$			F0 = F7			
Compared methods			Statistical test results			Statistical conclusion			

Critical value: 75.
*Statistically significant difference with $\alpha = 0.05$.

Table 9
Average MLP training time measured in seconds on real-worlds problems.

Data set	Problem	t (s)	F0	F1	F2	F3	F4	F5	F6	F7
D1	Batch	Avg	941.8999	82.7310	50.0339	44.7622	40.8820	11.1049	12.8499	24.3683
		SD	35.6358	1.6390	0.3410	0.8821	0.5209	0.1438	0.1827	0.1584
D2	Breast	Avg	0.4913	0.1415	0.1415	0.0984	0.0984	0.1000	0.0808	0.1086
		SD	0.0351	0.0173	0.0173	0.0175	0.0175	0.0099	0.0342	0.0189
D3	Column	Avg	0.1503	0.0536	0.0536	0.1173	0.1503	0.0814	0.0881	0.0881
		SD	0.0202	0.0067	0.0067	0.0124	0.0202	0.0101	0.0187	0.0187
D4	Heart	Avg	0.3779	0.1785	0.1785	0.2119	0.2106	0.1536	0.1445	0.1445
		SD	0.0356	0.0143	0.0143	0.0157	0.0119	0.0123	0.0446	0.0446
D5	Hepatitis	Avg	0.3341	0.1739	0.1739	0.1713	0.0823	0.0996	0.0485	0.0485
		SD	0.0266	0.0275	0.0275	0.0115	0.0073	0.0083	0.0479	0.0479
D6	Ionos	Avg	1.5902	0.5672	0.3974	0.3823	0.2656	0.2042	0.2133	0.2205
		SD	0.0586	0.0451	0.0255	0.0238	0.0275	0.0142	0.0232	0.0220
D7	Labor	Avg	0.3778	0.0437	0.0416	0.0314	0.0314	0.0553	0.0198	0.0198
		SD	0.0513	0.0047	0.0101	0.0037	0.0037	0.0043	0.0072	0.0072
D8	Leaves	Avg	1.3469	0.1687	0.0787	0.2099	0.2011	0.0828	0.0888	0.0978
		SD	0.0570	0.0152	0.0113	0.0228	0.0223	0.0124	0.0114	0.0208
D9	Messidor	Avg	2.6802	0.9531	0.5232	1.2312	1.1879	0.3400	0.3879	0.4375
		SD	0.0491	0.0389	0.0214	0.0457	0.0472	0.0155	0.0367	0.0257
D10	OBS	Avg	3.9574	0.5740	0.5740	1.2605	1.0066	0.5707	0.5056	0.6360
		SD	0.0655	0.0323	0.0323	0.0472	0.0400	0.0241	0.0250	0.0124
D11	Parkinsons	Avg	0.5755	0.0955	0.0955	0.1229	0.1146	0.0838	0.0489	0.0521
		SD	0.0378	0.0037	0.0037	0.0151	0.0100	0.0106	0.0080	0.0139
D12	Pasture	Avg	0.1015	0.0147	0.0147	0.0262	0.0262	0.0165	0.0256	0.0260
		SD	0.0197	0.0026	0.0026	0.0044	0.0044	0.0028	0.0113	0.0085
D13	Pima	Avg	0.6158	0.2427	0.2427	0.3037	0.3230	0.3012	0.1827	0.1606
		SD	0.0267	0.0193	0.0193	0.0130	0.0134	0.0134	0.0201	0.0104
D14	Pollen	Avg	13.3768	1.9614	1.9735	1.7277	1.7277	2.6611	2.4610	3.6057
		SD	0.1831	0.0873	0.0935	0.0776	0.0776	0.1092	0.0458	0.0471
D15	Promoter	Avg	5.0274	0.0635	0.0635	0.0799	0.0624	0.1052	0.0250	0.0250
		SD	0.3795	0.0097	0.0097	0.0115	0.0049	0.0114	0.0088	0.0088
D16	Soybean	Avg	24.6388	11.1830	5.9164	8.7835	5.6115	4.8149	5.8995	5.4244
		SD	0.3836	0.1178	0.0722	0.0916	0.1076	0.0698	0.4022	0.1971
D17	Squash	Avg	0.5214	0.0234	0.0234	0.1735	0.1570	0.1912	0.0270	0.0484
		SD	0.0512	0.0090	0.0090	0.0203	0.0210	0.0229	0.0116	0.0119
D18	Tokyo	Avg	4.7027	0.8965	0.4767	1.0597	1.3050	0.1920	0.3411	0.6110
		SD	0.1588	0.0399	0.0283	0.0438	0.0492	0.0098	0.0294	0.0427
D19	Waveform	Avg	32.9012	7.7809	7.7809	8.9106	8.8561	3.1716	6.0376	7.3420
		SD	1.2880	0.1279	0.1279	0.1586	0.1198	0.0685	0.4422	0.3784
D20	Winequality-red	Avg	2.8236	1.7081	1.3979	2.5500	2.5500	1.7905	1.4393	1.4393
		SD	0.0469	0.0462	0.0260	0.0898	0.0898	0.0482	0.0521	0.0521
D21	Winequality-white	Avg	10.9337	6.6371	6.6371	9.5272	9.5272	5.4741	3.8583	5.3741
		SD	0.4309	0.1246	0.1246	0.0916	0.0916	0.0804	0.1504	0.1571
D22	Yeast	Avg	2.8817	2.0219	1.9156	2.4039	2.4039	2.4117	2.4039	2.4039
		SD	0.0892	0.0480	0.0371	0.0821	0.0821	0.1048	0.0821	0.0821
Sum			1052.3062	118.2138	78.7341	84.1452	76.7809	34.0062	37.1773	52.6822

Table 10
Summary of RBFNN test accuracy results and statistical test results on synthetic data sets.

Data sets	Problems	Average Accuracy							
		F0	F1	F2	F3	F4	F5	F6	F7
D23-D34	d50r5g0, d50r5g5, d50r10g0, d50r10g5, d100r5g0, d100r5g5, d100r10g0, d100r10g5, d100r15g0, d100r15g5, d100r20g0, d100r20g5	76.51	74.16	73.78	73.04	71.14	73.78	77.47	78.22
Pairwise Comparison									
F0 vs F7		$T=26$				F0 = F7			
F0 vs F1		$T=21$				F1 = F0			
Compared methods		Statistical test results				Statistical conclusion			

Critical value: 17.
*Statistically significant difference with $\alpha = 0.05$.

the same learning with the reduced data sets is 0.15 s, about seventy times less.

5.4. Joint results with state-of-the-art classifiers

Table 12 depicts the RBF and MLP results alongside those obtained with two typical methods within the data mining community, one coming from decision tree approaches (J48) and another that is very popular due to its excellent performance (SMO). According to the aforesaid comments about neural classifiers, F7 is very convenient for RBF and MLP. For J48, the best option is F6 as well as some methods such as F3 in spite of the low performance with certain neural models. For SMO, F7 is again a good filter selection method based on subsets.

Table 11
Summary of MLP test accuracy results and statistical test results on synthetic data sets.

Data sets	Problems	Average Accuracy							
		F0	F1	F2	F3	F4	F5	F6	F7
D23-D34	d50r5g0, d50r5g5, d50r10g0, d50r10g5, d100r5g0, d100r5g5, d100r10g0, d100r10g5, d100r15g0, d100r15g5, d100r20g0, d100r20g5	85.26	77.45	76.61	75.78	73.98	76.61	83.58	84.05
Pairwise comparison									
F0 vs F7		$T=27$				F0 = F7			
Compared methods		Statistical test results				Statistical conclusion			

Critical value: 17.
*Statistically significant difference with $\alpha = 0.05$.

Table 12
Joint test accuracy results with the whole test-bed for four classifiers.

Data set	Problem	Classifier	F0	F1	F2	F3	F4	F5	F6	F7
D1	Batch	RBF	65.47	68.88	70.01	65.19	66.79	69.67	76.77	75.25
		MLP	92.11	96.64	94.25	92.75	85.65	69.07	90.19	96.45
		J48	97.50	96.58	95.17	96.75	95.43	75.01	96.98	97.70
		SMO	97.21	87.67	81.66	81.97	69.84	58.97	80.07	88.70
D2	Breast	RBF	68.78	67.46	67.46	69.01	69.01	67.65	65.49	65.77
		MLP	61.13	69.01	69.01	69.01	69.01	69.53	64.79	67.32
		J48	70.42	69.01	69.01	69.01	69.01	69.01	64.79	69.01
		SMO	64.79	66.20	66.20	64.79	64.79	64.79	64.79	64.79
D3	Column	RBF	81.15	79.62	79.62	83.93	81.15	80.77	82.86	82.86
		MLP	82.44	80.68	80.68	83.50	82.44	82.69	78.25	78.25
		J48	80.77	78.21	78.21	80.77	80.77	71.79	71.79	71.79
		SMO	76.92	67.95	67.95	76.92	76.92	69.23	67.95	67.95
D4	Heart	RBF	78.53	78.24	78.24	75.39	77.60	75.92	74.71	74.71
		MLP	74.80	72.65	72.65	73.14	74.85	74.85	73.33	73.33
		J48	70.59	73.53	73.53	73.53	72.06	73.53	75.00	75.00
		SMO	76.47	76.47	76.47	76.47	76.47	77.94	75.00	75.00
D5	Hepatitis	RBF	89.30	89.30	89.30	89.91	88.42	89.53	88.42	88.42
		MLP	85.00	87.28	87.28	86.75	84.21	87.72	86.84	86.84
		J48	84.21	84.21	84.21	89.47	89.47	89.47	89.47	89.47
		SMO	89.47	86.84	86.84	89.47	89.47	89.47	86.84	86.84
D6	Ionos	RBF	92.46	95.49	94.73	94.51	93.39	94.09	93.11	93.56
		MLP	88.94	92.01	89.85	92.12	92.84	89.24	91.29	95.15
		J48	92.05	92.05	92.05	94.32	94.32	90.91	89.77	90.91
		SMO	88.64	87.50	88.64	84.09	82.95	82.95	86.36	88.64
D7	Labor	RBF	71.67	71.43	85.71	64.29	64.29	67.56	71.43	71.43
		MLP	69.52	64.29	78.57	78.57	78.57	71.43	71.43	71.43
		J48	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71
		SMO	78.57	78.57	85.71	78.57	78.57	71.43	71.43	71.43
D8	Leaves	RBF	67.48	65.19	67.56	70.52	67.45	67.41	77.56	64.74
		MLP	71.48	70.89	68.07	68.89	72.89	66.96	72.44	63.63
		J48	71.11	64.44	44.44	66.67	62.22	57.78	60.00	57.78
		SMO	66.67	62.22	66.67	66.67	62.22	60.00	62.22	64.44
D9	Messidor	RBF	59.92	60.94	60.17	60.07	63.68	61.50	59.06	59.25
		MLP	72.53	71.82	71.40	70.28	71.27	61.27	60.66	71.85
		J48	62.15	59.03	57.99	62.85	62.15	60.76	59.38	62.15
		SMO	63.19	61.46	61.46	62.15	62.50	60.42	61.46	63.19
D10	OBS	RBF	69.32	74.99	74.99	70.21	71.20	74.68	70.55	74.11
		MLP	81.00	75.50	75.50	79.49	79.55	72.64	76.20	78.19
		J48	86.99	81.04	81.04	81.78	81.78	79.93	87.36	81.78
		SMO	75.84	73.61	73.61	71.00	72.49	59.85	72.12	71.00
D11	Parkinsons	RBF	70.27	77.55	77.55	74.56	73.47	81.42	81.63	83.67
		MLP	77.62	81.56	81.56	75.92	75.65	84.83	80.61	80.61
		J48	71.43	75.51	75.51	75.51	79.59	81.63	81.63	75.51
		SMO	75.51	75.51	75.51	75.51	75.51	79.59	79.59	81.63
D12	Pasture	RBF	64.81	74.44	74.44	91.48	91.48	70.37	73.70	91.11
		MLP	66.67	72.22	72.22	77.78	77.78	56.67	88.89	88.89
		J48	77.78	77.78	77.78	77.78	77.78	77.78	88.89	88.89
		SMO	77.78	77.78	77.78	77.78	77.78	66.67	77.78	66.67
D13	Pima	RBF	77.34	79.17	79.17	77.50	75.64	79.29	79.10	73.32
		MLP	76.74	78.18	78.18	74.27	76.91	79.03	76.04	78.25
		J48	74.48	76.04	76.04	69.79	74.48	76.04	75.52	73.96
		SMO	78.13	77.60	77.60	73.96	78.65	79.17	76.56	77.60
D14	Pollen	RBF	91.73	91.89	91.21	89.72	89.72	92.03	89.66	93.27
		MLP	96.39	91.67	91.59	88.74	88.74	92.84	93.51	94.65

(*continued on next page*)

Table 12 (*continued*)

Data set	Problem	Classifier	F0	F1	F2	F3	F4	F5	F6	F7
		J48	88.92	87.76	86.59	88.34	88.34	91.84	87.76	88.63
		SMO	93.88	88.34	88.05	78.72	78.72	89.50	88.34	93.59
D15	Promoter	RBF	79.36	83.46	83.46	76.03	85.00	79.01	80.77	80.77
		MLP	86.03	84.49	84.49	65.00	75.51	78.46	80.77	80.77
		J48	69.23	73.08	73.08	76.92	80.77	73.08	80.77	80.77
		SMO	88.46	84.62	84.62	76.92	84.62	73.08	80.77	80.77
D16	Soybean	RBF	93.84	93.47	93.20	92.81	89.61	91.38	93.24	94.17
		MLP	92.87	92.13	90.72	92.44	88.93	88.78	92.69	92.27
		J48	93.02	94.19	92.44	91.86	89.53	90.12	94.77	94.77
		SMO	93.60	94.19	94.19	94.77	93.02	93.02	95.35	95.35
D17	Squash	RBF	80.77	85.64	85.64	82.05	75.38	80.77	70.26	80.00
		MLP	80.26	76.92	76.92	84.62	80.51	76.92	81.79	94.36
		J48	69.23	76.92	76.92	76.92	76.92	76.92	76.92	84.62
		SMO	92.31	69.23	69.23	76.92	84.62	84.62	69.23	84.62
D18	Tokyo	RBF	89.56	88.76	87.94	88.35	89.65	89.65	89.49	91.45
		MLP	91.37	91.92	91.13	91.44	90.05	90.05	90.74	92.15
		J48	90.63	89.38	92.50	89.38	89.38	89.38	90.83	90.83
		SMO	91.67	92.08	90.83	91.67	91.04	91.04	90.21	91.46
D19	Waveform	RBF	82.14	82.24	82.24	82.55	82.22	76.89	82.63	82.55
		MLP	80.41	83.24	83.24	83.42	83.13	77.58	82.81	82.64
		J48	74.80	74.40	74.40	74.88	74.40	74.72	74.56	74.40
		SMO	86.24	86.88	86.88	87.12	87.12	78.80	85.52	85.92
D20	Winequality-red	RBF	57.11	59.00	57.53	59.19	59.19	58.88	57.59	57.59
		MLP	56.22	59.45	58.95	57.04	57.04	59.61	59.04	59.04
		J48	53.85	50.87	50.37	50.12	50.12	51.36	55.58	55.58
		SMO	59.55	59.80	57.07	58.81	58.81	59.31	59.06	59.06
D21	Winequality-white	RBF	48.04	51.39	51.39	48.90	48.90	51.08	52.15	50.85
		MLP	52.21	53.02	53.02	52.65	52.65	51.40	51.80	51.41
		J48	46.21	44.17	44.17	42.05	42.05	43.85	48.17	43.93
		SMO	52.97	52.57	52.57	52.89	52.89	51.83	52.24	50.94
D22	Yeast	RBF	58.33	58.41	54.97	58.90	58.90	58.48	54.97	54.97
		MLP	59.84	60.10	55.11	60.06	60.06	59.01	55.11	55.11
		J48	54.84	53.49	54.30	54.03	54.03	52.69	54.30	54.30
		SMO	55.91	54.03	53.76	54.84	54.84	51.61	53.76	53.76
D23	d50r5g0	RBF	82.27	86.37	86.37	86.37	86.37	86.37	85.39	83.12
		MLP	96.45	99.15	99.15	99.15	99.15	99.15	99.04	98.93
		J48	76.00	80.00	80.00	80.00	80.00	80.00	76.80	76.80
		SMO	94.40	97.60	97.60	97.60	97.60	97.60	98.40	99.20
D24	d50r5g5	RBF	78.60	80.83	80.83	75.78	67.56	80.83	83.25	81.55
		MLP	88.75	84.24	84.24	75.71	68.48	84.24	92.08	89.55
		J48	70.40	72.80	72.80	72.80	63.20	72.80	70.40	65.60
		SMO	87.20	85.60	85.60	74.40	68.80	85.60	89.60	91.20
D25	d50r10g0	RBF	73.64	72.47	72.47	72.47	72.47	72.47	79.01	79.52
		MLP	88.72	76.69	76.69	76.69	76.69	76.69	79.84	80.35
		J48	64.00	71.20	71.20	71.20	71.20	71.20	76.00	69.60
		SMO	90.40	84.80	84.80	84.80	84.80	84.80	83.20	84.00
D26	d50r10g5	RBF	76.88	68.63	66.40	68.63	66.40	66.40	75.41	73.89
		MLP	77.79	70.24	68.32	70.24	68.32	68.32	84.64	77.33
		J48	64.00	60.80	62.40	60.80	62.40	62.40	67.20	63.20
		SMO	79.20	68.80	71.20	68.80	71.20	71.20	80.80	75.20
D27	d100r5g0	RBF	78.25	86.37	86.37	86.37	86.37	86.37	85.12	86.88
		MLP	90.13	99.15	99.15	99.15	99.15	99.15	97.63	98.75
		J48	71.20	80.00	80.00	80.00	80.00	80.00	76.00	79.20
		SMO	92.00	97.60	97.60	97.60	97.60	97.60	97.60	95.20
D28	d100r5g5	RBF	76.61	80.83	80.83	75.78	67.56	80.83	81.47	83.12
		MLP	83.01	84.24	84.24	75.71	68.48	84.24	88.67	88.32
		J48	67.20	72.80	72.80	72.80	63.20	72.80	72.00	72.80
		SMO	79.20	85.60	85.60	74.40	68.80	85.60	89.60	92.80
D29	d100r10g0	RBF	73.59	72.47	72.47	72.47	72.47	72.47	77.81	82.77
		MLP	85.25	76.69	76.69	76.69	76.69	76.69	77.23	95.36
		J48	60.00	71.20	71.20	71.20	71.20	71.20	74.40	65.60
		SMO	86.40	84.80	84.80	84.80	84.80	84.80	84.80	96.00
D30	d100r10g5	RBF	74.63	70.04	66.40	70.04	66.40	66.40	75.41	70.72
		MLP	78.67	69.65	68.32	69.65	68.32	68.32	84.64	69.71
		J48	63.20	60.80	62.40	60.80	62.40	62.40	67.20	63.20
		SMO	76.00	70.40	71.20	70.40	71.20	71.20	80.80	67.20
D31	d100r15g0	RBF	77.36	72.28	71.86	72.28	66.62	71.86	67.65	81.57
		MLP	89.81	72.75	70.72	72.75	70.72	70.72	66.80	85.71
		J48	66.40	67.20	68.00	67.20	61.60	68.00	67.20	63.20
		SMO	87.20	73.60	77.60	73.60	70.40	77.60	67.20	87.20
D32	d100r15g5	RBF	75.84	67.01	67.83	63.67	67.83	67.83	73.44	70.51
		MLP	80.43	65.44	62.64	62.53	62.64	62.64	77.87	70.24
		J48	57.60	61.60	60.80	60.80	60.80	60.80	57.60	58.40
		SMO	82.40	70.40	69.60	66.40	69.60	69.60	80.80	74.40

(*continued on next page*)

Table 12 (*continued*)

Data set	Problem	Classifier	F0	F1	F2	F3	F4	F5	F6	F7
D33	d100r20g0	RBF	74.04	68.14	68.14	68.14	68.14	68.14	77.36	69.73
		MLP	85.31	67.60	67.60	67.60	67.60	67.60	84.29	72.77
		J48	58.40	61.60	61.60	61.60	61.60	61.60	61.60	63.20
		SMO	88.00	75.20	75.20	75.20	75.20	75.20	85.60	74.40
D34	d100r20g5	RBF	76.45	64.52	65.43	64.52	65.43	65.43	68.35	75.23
		MLP	78.77	63.52	61.57	63.52	61.57	61.57	70.21	81.55
		J48	55.20	64.00	64.00	64.00	64.00	64.00	56.80	60.00
		SMO	80.00	71.20	70.40	71.20	70.40	70.40	72.00	82.40
Average		RBF	75.16	75.50	75.65	74.75	73.99	74.81	76.32	77.13
		MLP	79.96	77.50	77.17	76.68	76.06	75.29	79.48	80.62
		J48	71.75	72.98	72.43	73.28	72.70	72.37	73.92	73.19
		SMO	80.77	77.26	77.48	76.21	76.01	75.43	77.85	78.90

6. Conclusions

This paper studied in detail the performance of two neural models, RBFNN and MLP, in the context of feature subset selection including filters and semi-wrappers. Experiments were conducted in 34 data sets, pertaining to real-world problems and also to synthetic data sets. Seventeen statistical tests have been done to extract as many important facts as possible for future research on this trending topic, namely, feature selection.

In essence, F7 (an approach based on Best Incremental Ranked Subset semi-wrapper -$BIRS_{SW}$- with Naïve Bayes as a subset evaluator, under the framework of Correlation-based Feature Selection as ranking method) is a good option for both RBFNN and MLP. F1 (Selection Of Attributes by Projection -SOAP- as a measure to assess the quality of the subsets within Best Incremental Ranked Subset integrated into Correlation-based Feature Selection) would be a secondary solution for RBFNN. In some specific scenarios, F2 (similar to F1 with the difference that a correlation measure is used instead of SOAP) is a real alternative to F1 in the context of RBFNN. For real-world problems F1 and F2 are significantly more accurate than F0. Synthetic data sets require F7 although for RBFNN F1 could be appropriate. The key issue for MLP is that an outstanding reduction in computational cost has been achieved via F7. The results are kept compared to the full attribute space with a lower computational cost.

F4 (Consistency-based Feature Selection) and F5 (Fast Correlation-based Feature Selection) are methods that could be discarded in the context of feed-forward neural networks, especially if any of the remaining feature selection methods are available. The greater convenience to apply the classifier without feature reduction either F4 or F5 could not be anticipated. It depends on the goal: if the solution is required within a certain time frame or if there is no time frame on finding a solution.

The application of the filter selection procedures to SMO concluded that F7 is a promising filter. J48 met a perfect subset selection procedure in F6 (a filter based on $BIRS_{SW}$ and Naïve Bayes as ranking method) and F3 (a feature subset selection based on Consistency and also integrated with Best Incremental Ranked Subset) as its follower.

7. Prospective works

Since the research has been concluded it is important to now remark upon some new research lines. F7 has exhibited a good performance. The number of selected attributes is the highest compared to the remaining feature selection methods. It could draw the attention that a prune in the feature space is convenient but we need to be careful because the prune could accidentally remove features that do not seem very important but contribute to have a strong solution. For the future, it could be very interesting to attempt a strategy of merging solutions could be very interesting to be applied to F7 or even F6, since these are the methods with the higher number of selected attributes. F7 probably could need some extra attributes that could be incorporated by adding them to features collected by another feature selection method. F4 and F5 are methods whose performance is uncertain.

For J48, F6 yields an appropriate number of features although a higher number of features is not convenient. It may be that, a reduction of attributes starting from the solution provided by F6 is a possible way forward.

Acknowledgements

This work has been partially subsidised by TIN2014-55894-C2-R project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7528 project of the “Junta de Andalucía” (Spain).

References

[1] T.J. Sejnowski, Neural network learning algorithms, Neural Computers, Springer, Berlin, Heidelberg, 1989, pp. 291–300.
[2] F.M. Ham, I. Kostanic, Principles of Neurocomputing for Science and Engineering, McGraw-Hill Higher Education, 2000.
[3] T. Trappenberg, Fundamentals of Computational Neuroscience, OUP Oxford, 2009.
[4] G.G. Towell, J.W. Shavlik, Knowledge-based artificial neural networks, Artif. Intell. 70 (1–2) (1994) 119–165.
[5] H. Duda, P. Hart, G. David, Pattern Classification, Stork, 2001.
[6] K.L. Du, M.N.S. Swamy, Fundamentals of machine learning, Neural Networks and Statistical Learning, Springer, London, 2014, pp. 15–65.
[7] S.E. Fahlman, G.E. Hinton, Connectionist architectures for artificial intelligence, Computer (United States) 20 (1) (1987).
[8] R. Pfeifer, Z. Schreter, F. Fogelman-Soulié, L. Steels (Eds.), Connectionism in Perspective, Elsevier, 1989.
[9] H. White, Learning in artificial neural networks: a statistical perspective, Learning 1 (4) (2008).
[10] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford university press, 1995.
[11] R. Rojas, Neural networks: A Systematic Introduction, Springer Science & Business Media, 2013.
[12] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[13] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, Computer 29 (3) (1996) 31–44.
[14] D.S. Broomhead, D. Lowe, Radial Basis functions, Multi-Variable Functional Interpolation and Adaptive Networks, Royal Signals and Radar Establishment Malvern, United Kingdom, 1988 (No. RSRE-MEMO-4148).
[15] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[16] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[17] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.
[18] H.H. Hsu, C.W. Hsieh, M.D. Lu, Hybrid feature selection by combining filters and wrappers, Expert Syst. Appl. 38 (7) (2011) 8144–8150.
[19] C.H. Yang, L.Y. Chuang, C.H. Yang, IG-GA: a hybrid filter/wrapper method for feature selection of microarray data, J. Med. Biol. Eng. 30 (1) (2010) 23–28.

[20] M. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: P. Langley (Ed.), Proceedings of the Seventeenth International Conference on Machine Learning (ICML), Morgan Kaufmann, San Francisco, CA, 2000, pp. 359–366.
[21] H. Liu, R. Setiono, A probabilistic approach to feature selection– a filter solution, in: L. Saitta (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996), Morgan Kaufmann, Italy, 1996, pp. 319–327.
[22] R. Ruiz, J. Riquelme, J. Aguilar-Ruiz, Projection-based measure for efficient feature selection, J. Intell. Fuzzy Syst. 12 (3–4) (2002) 175–183.
[23] T. Bayes, An essay towards solving a problem in the doctrine of chances, Philos. Trans. 53 (1763) 370–418.
[24] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (Oct) (2004) 1205–1224.
[25] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recognit. 39 (12) (2006) 2383–2392.
[26] P. Bermejo, J.A. Gamez, J.M. Puerta, Improving incremental wrapper-based subset selection via replacement and early stopping, Int. J. Pattern Recognit. Artif. Intell. 25 (05) (2011) 605–625.
[27] P. Bermejo, J.A. Gámez, J.M. Puerta, Speeding up incremental wrapper feature subset selection with Naive Bayes classifier, Knowl. Based Syst. 55 (2014) 140–147.
[28] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, M. García-Torres, Fast feature selection aimed at high-dimensional data via hybrid-sequential-ranked searches, Expert Syst. Appl. 39 (12) (2012) 11094–11102.
[29] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the Twentieth International Conference on Machine Learning (ICML-03), 2003, pp. 856–863.
[30] R. Setiono, H. Liu, Neural-network feature selector, IEEE Trans. Neural Netw. 8 (3) (1997) 654–662.
[31] J. Basak, S. Mitra, Feature selection using radial basis function networks, Neural Comput. Appl. 8 (4) (1999) 297–302.
[32] N. Srilatha, G. Yesuratnam, Security assessment and enhancement using RBFNN with feature selection, in: *Proceedings of the* North American Power Symposium (NAPS), 2014, IEEE, 2014, pp. 1–5.
[33] K. Zhang, Y. Li, P. Scarf, A. Ball, Feature selection for high-dimensional machinery fault diagnosis data using multiple models and radial basis function networks, Neurocomputing 74 (17) (2011) 2941–2952.
[34] A.J. Tallón-Ballesteros, C. Hervás-Martínez, J.C. Riquelme, R. Ruiz, Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems, Neurocomputing 114 (2013) 107–117.
[35] C.N. Hsu, H.J. Huang, S. Dietrich, The ANNIGMA-wrapper approach to fast feature selection for neural nets, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 32 (2) (2002) 207–212.
[36] E. Gasca, J.S. Sánchez, R. Alonso, Eliminating redundancy and irrelevance using a new MLP-based feature selection method, Pattern Recognit. 39 (2) (2006) 313–315.
[37] J.B. Yang, K.Q. Shen, C.J. Ong, X.P. Li, Feature selection for MLP neural network: the use of random permutation of probabilistic outputs, IEEE Trans. Neural Netw. 20 (12) (2009) 1911–1922.
[38] M.M. Kabir, M.M. Islam, K. Murase, A new wrapper feature selection approach using neural network, Neurocomputing 73 (16–18) (2010) 3273–3283.
[39] X. Zeng, Z. Zhen, J. He, L. Han, A feature selection approach based on sensitivity of RBFNNs, Neurocomputing 275 (2018) 2200–2208.
[40] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing (2019) (in press).
[41] M. Sebban, R. Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognit. 35 (4) (2002) 835–846.
[42] C. Sammut, G.I. Webb, Encyclopedia of Machine Learning and Data Mining, Springer, 2017.
[43] T. Mitchell, Generative and discriminative classifiers: naive Bayes and logistic regression, Machine Learning, McGraw-Hill, 2005.
[44] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1998.
[45] R. Kohavi, Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid, in: *Proceedings of the* KDD, 96, 1996, pp. 202–207.
[46] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Machine Learning Proceedings, 1994, 1994, pp. 121–129.
[47] K. Bache, M. Lichman, (2015). UCI machine learning repository online.
[48] K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, M. Sebag, Ensemble feature ranking, in: *Proceedings of the* Knowledge Discovery in Databases: PKDD 2004, 2004, pp. 267–278.
[49] J.R. Quevedo, A. Bahamonde, O. Luaces, A simple and efficient method for variable ranking according to their usefulness for learning, Comput. Stat. Data Anal. 52 (1) (2007) 578–595.
[50] A.J. Tallón-Ballesteros, J.C. Riquelme, R. Ruiz, Accuracy increase on evolving product unit neural networks via feature subset selection, in: Proceedings of the International Conference on Hybrid Artificial Intelligence Systems (HAIS 2016), Cham, Springer, 2016, April, pp. 136–148.
[51] A.J. Tallón-Ballesteros, J.C. Riquelme, R. Ruiz, Merging subsets of attributes to improve a hybrid consistency-based filter: a case of study in product unit neural networks, Connect. Sci. 28 (3) (2016) 242–257.
[52] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newslett. 11 (1) (2009) 10–18.
[53] A.J. Tallón-Ballesteros, C. Hervás-Martínez, P.A. Gutiérrez, An extended approach of a two stage evolutionary algorithm in artificial neural networks for multiclassification tasks, Innovations in Intelligent Machines-3, Springer, Cham, 2013, pp. 139–153.
[54] A.J. Tallón-Ballesteros, J.C. Riquelme, Data mining methods applied to a digital forensics task for supervised machine learning, Computational Intelligence in Digital Forensics: Forensic Investigation and Applications, Springer, Cham, 2014, pp. 413–428.

Antonio J. Tallón-Ballesteros received the Technical Engineer degree in Systems Computer Science from the University of Córdoba, Spain, in 2002, the M.Sc. degree in Computer Science from the University of Granada, Spain, in 2004 and the Ph.D. degree in Computer Science in 2013 from the University of Seville (Spain). He obtained the Accreditation to Senior Lecturer in 2013, a couple of months after the dissertation. He has been a Lecturer since 2005 with the University of Seville in the Department of Languages and Computer Systems. His current research lines include data preparation, neural networks, evolutionary algorithms and machine learning.

José C. Riquelme received the M.Sc. degree in Mathematics and the Ph.D. degree in Computer Science from the University of Seville, Spain. Since 1987 he has been with the Department of Computer Science, University of Seville, where he is currently Professor. His primary areas of interest are data mining, machine learning techniques and evolutionary computation.

Roberto Ruiz received the M.Sc. degree in Computer Science in 2000 and the Ph.D. degree in Computer Science in 2006, both from the University of Seville, Spain. He is an Associate Professor at Pablo de Olavide University of Seville, Spain. He is a member of the Bioinformatics Research Group of Seville (BIGS). He conducts research in genetic programming, feature selection and data mining.