Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems

Antonio J. Tallón-Ballesteros, César Hervás-Martínez, José C. Riquelme, Roberto Ruiz

ABSTRACT

Keywords:
Artificial neural networks
Product units
Evolutionary algorithms
Classification
Feature selection
High error problems

This paper combines feature selection methods with a two-stage evolutionary classifier based on product unit neural networks. The enhanced methodology has been tried out with four filters using 18 data sets that report test error rates about 20 % or above with reference classifiers such as C4.5 or 1-NN. The proposal has also been evaluated in a liver-transplantation real-world problem with serious troubles in the data distribution and classifiers get low performance. The study includes an overall empirical comparison between the models obtained with and without feature selection using different kind of neural networks, like RBF, MLP and other state-of-the-art classifiers. Statistical tests show that our proposal significantly improves the test accuracy of the previous models. The reduction percentage in the number of inputs is, on average, above 55 %, thus a greater efficiency is achieved.

1. Introduction

There are several machine learning techniques to deal with a classification problem, such as neural networks, radial basis functions, rules and decision trees. A review of them can be found in [1]. The explosion of available information complicates this problem. Moreover, redundancy or noise may be present on data [2]. Neural networks models play a crucial role in pattern recognition [3]. For many practical problems, the possible inputs to an Artificial Neural Network (ANN) can be huge. There may be some redundancy among different inputs. A large number of inputs to an ANN increase its size and thus require more training data and longer training times in order to achieve reasonable generalization ability. Pre-processing is often needed to reduce the number of inputs to an ANN. The application of feature selection (FS) approaches has become a real prerequisite for model building due to the multi-dimensional nature of many modelling task in some fields. Theoretically, having more features should give us more discriminating power. However, this can cause several problems: an increased computational complexity and cost, too many redundant or irrelevant features, and degradation in the classification error estimation.

Our objective is to improve the accuracy and to reduce the complexity (measured by means of the number of inputs) of the models of Evolutionary ANNs (EANNs) with product units (PUs) that have been employed to date by us. The training of databases for classification, which have different numbers of patterns, features and classes, is dealt with by means of ANNs. The computational cost is very high if Evolutionary Algorithms (EAs) with different parameter settings are employed for the training of the above-mentioned networks. However, in this paper we use a specialization of an EA called TSEA (Two-Stage Evolutionary Algorithm) [4] which add broader diversity at the beginning of the evolution. First of all, FS is applied to the data sets in order to eliminate redundant and irrelevant variables. In this way, the complexity could be reduced and the accuracy could be increased. The reduction in the number of inputs could decrement the number of nodes in the hidden-layer and, hence, also simplify the associated model. Several runs of the TSEA have been performed to smooth the stochastic character using mean values in order to complete a statistical analysis of the results obtained. This paper is organized as follows: Section 2 describes some concepts about FS and the classification with TSEA in evolutionary product unit neural networks (PUNNs); Section 3 presents the description of our proposal; Section 4 details the experimentation process; then Section 5 shows and analyzes the results obtained; finally, Section 6 states the concluding remarks.

2. Methodology

2.1. Feature selection

The selection of features and the removal or reduction of redundant information unrelated to the classification task on hand will not only reduce the complexity of the problem and improve the efficiency of the processing but also simplify significantly the design of the classifier. The FS is one of the essential and frequently used techniques in machine learning. A FS method generates different candidates from the feature space and assesses them based on an evaluation criterion to find the best feature subset [5]. On the basis of the evaluation criterion, FS can be divided into filter methods and wrapper methods. Filters assess the relevance of features by looking only at the intrinsic properties of the data, such as distance, consistency, and correlation [5–7]. These criteria are independent of any inductive learning algorithm. In contrast, the wrapper approach requires one predetermined mining algorithm and uses its performance to evaluate and determine which features are selected [8]. Wrappers often select features that have a higher accuracy; however, they are criticized for their high computational cost and low generality. To take advantage of the above two approaches, a hybrid model was proposed to handle large data sets [9]. Moreover, some methods, known as embedded, use internal information of the classification model to perform FS [10,11].

Based on the generation procedure, FS can be divided into individual feature ranking (FR) and feature subset selection (FSS) [10,12]. FR measures the relevance of each feature to the class and then ranks features by their scores and selects the top-ranked features. These methods are widely used because of their simplicity, scalability, and good empirical success [10,13]. However, FR is criticized because it can capture only the relevance of the features to the target concept, whereas the redundancy and basic interactions between features are not discovered. Additionally, the number of features retained is difficult to determine; as a result, a threshold is required. In contrast, FSS attempts to find a set of features that have good performance. This method integrates the metric for measuring the feature–class relevance and the feature–feature interactions. In [14] Liu and Yu, a large number of selection methods are categorized, in which different algorithms address these issues distinctively. We found different search strategies, namely exhaustive, heuristic and random searches, and combined them with several types of measures to form different algorithms. The time complexity is exponential in terms of the data dimensionality for an exhaustive search, and it is quadratic for a heuristic search. The complexity can be linear with the number of iterations in a random search, but experiments show that, to find the best feature subset, the number of iterations required is usually at least quadratic to the number of features [15]. In this categorization, to handle large data sets, a hybrid model was also proposed to combine the advantages of the FR and FSS techniques. These methods decouple relevance analysis and redundancy analysis, and they have been proven to be more effective than ranking methods and more efficient than subset evaluation methods on many traditional high-dimensional data sets. In this framework, [16] proposed a hybrid search algorithm. Yu and Liu [17] proposed a fast correlation-based filter algorithm (FCBF) that used a correlation measure to obtain relevant features and to remove redundancy. Ding and Peng [18] used mutual information for gene selection, finding maximum relevance with minimal redundancy by solving a simple two-objective optimization.

2.2. Classification with evolutionary product unit neural networks based on a two-stage algorithm

There are several kinds of neural networks, being the single-hidden-layer feed-forward network architecture the most popular one. Multiplicative neural networks contain nodes that multiply their inputs instead of adding them. This class of neural networks comprises such types as sigma-pi networks and product unit networks. The latter type was introduced by R. Durbin and D. Rumelhart [19]. The methodology employed here consists of the use of an EA as a tool for learning the architecture and weights of a PUNN model [20]. More details about PUNNs, such as some of the advantages, the universal approximation theorem, problems and learning methods, can be found in [4,21].

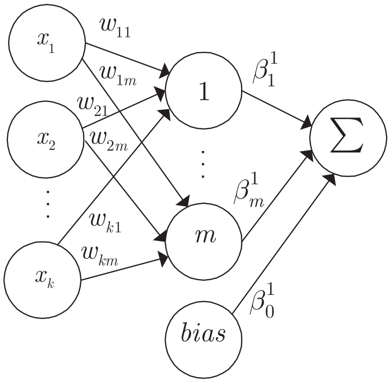


Fig. 1. Structure of a product unit neural network model for a bi-classification problem.

Fig. 1 shows the structure of a PUNN model with a *k:m:*1 architecture for a bi-classification problem; this is a three-layer architecture, that is, *k* nodes in the input layer, *m* ones (product units) and a bias one in the hidden layer and one node in the output layer.

The transfer function of each node in the hidden and output layers is the identity function. Thus, the functional model obtained by each of the nodes in the output layer with J classes is given by:

$$f(x_1,x_2,\ldots,x_k)=\beta_0^l+\sum_{j=1}^{m}\beta_j^l\prod_{i=1}^{k}x_i^{w_{ij}}\Bigg)\ l=1,2,\ldots,J;w_{ij}\in\Re \tag{1}$$

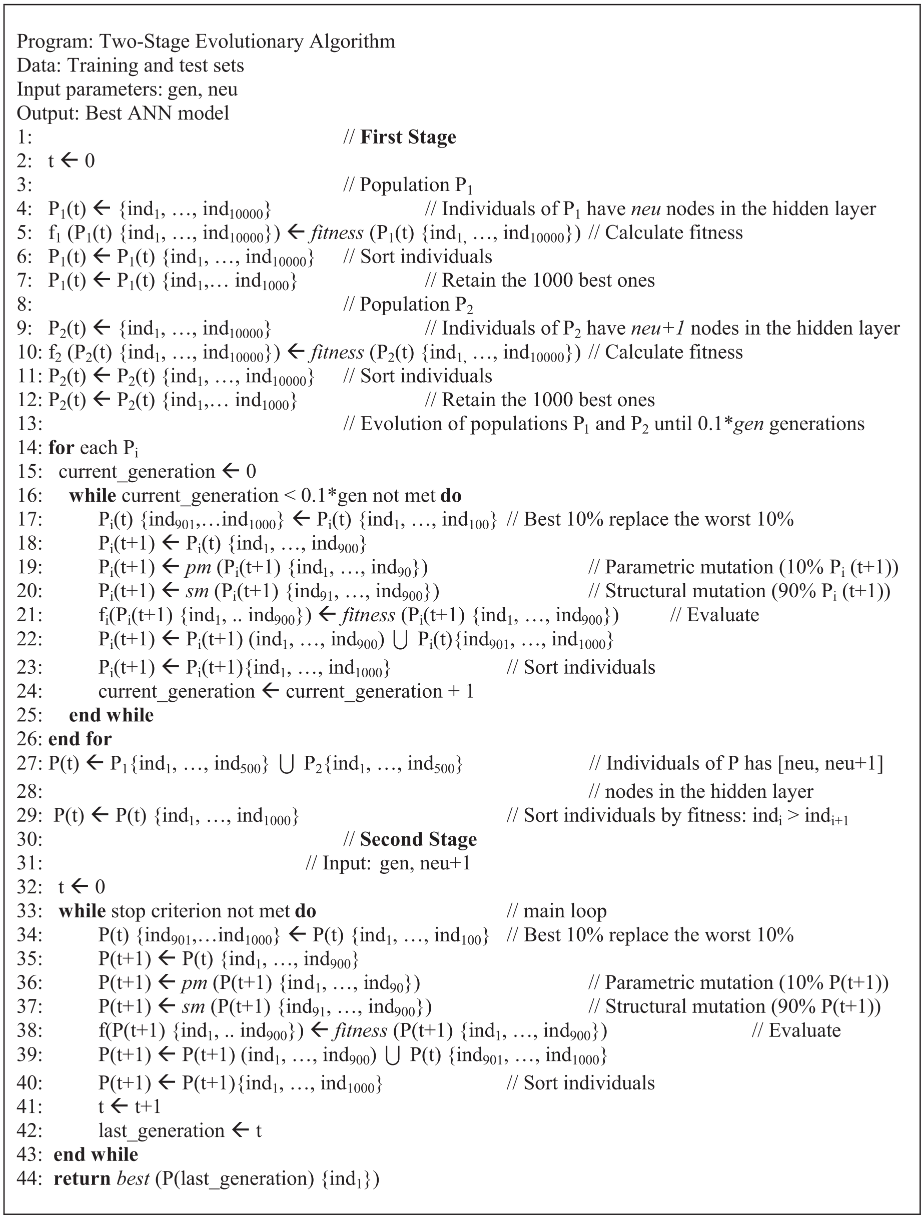
Next, we are going to describe briefly the TSEA applied. A full explanation of it and the details about common parameters can be read in Section 3 of [4]. TSEA is used to design the structure and learn the weights of PUNNs in two sequential phases. The population is subjected to the operations of replication and mutation; two types of mutations have been applied: parametric and structural ones. The TSEA pseudo-code for a classification problem appears in Fig. 2. In the first stage, TSEA evolves two populations for a small number of generations. The best half individuals of each one are merged in a new population that follows the full evolutionary cycle. The main parameters of the TSEA are the maximum number of generations (gen) and the maximum number of nodes in the hidden layer (neu). The minimum number of nodes is an unit lower than neu. The remaining parameters will be described further on. At the end of the TSEA, it returns the best PUNN model with a number of nodes between *neu* and *neu+1* in the hidden layer.

We have considered a standard soft-max activation function, associated with the *g* network model, given by:

$$g_j(\mathbf{x})=\frac{\exp f_j(\mathbf{x})}{\sum_{j=1}^{J}\exp f_j(\mathbf{x})}\qquad j=1,\ldots,J \tag{2}$$

where *J* is the number of classes in the problem, $f_j(\mathbf{x})$ is the output of node *j* for pattern **x** and $g_j(\mathbf{x})$ is the probability that this pattern belongs to class *j*.

Given a training set $D=(\boldsymbol{x}_i,\boldsymbol{y}_i)i=1,\ldots,N$, a function of cross-entropy error is used to evaluate a network *g* with the instances of a problem



```
Program: Two-Stage Evolutionary Algorithm
Data: Training and test sets
Input parameters: gen, neu
Output: Best ANN model
                                    // **First Stage**
t ← 0
                                    // Population $P_1$
$P_1(t) \leftarrow \{ind_1, \dots, ind_{10000}\}$          // Individuals of $P_1$ have *neu* nodes in the hidden layer
$f_1 (P_1(t) \{ind_1, \dots, ind_{10000}\}) \leftarrow$ *fitness* $(P_1(t) \{ind_1, \dots, ind_{10000}\})$ // Calculate fitness
$P_1(t) \leftarrow P_1(t) \{ind_1, \dots, ind_{10000}\}$    // Sort individuals
$P_1(t) \leftarrow P_1(t) \{ind_1, \dots ind_{1000}\}$          // Retain the 1000 best ones
                                    // Population $P_2$
$P_2(t) \leftarrow \{ind_1, \dots, ind_{10000}\}$          // Individuals of $P_2$ have *neu+1* nodes in the hidden layer
$f_2 (P_2(t) \{ind_1, \dots, ind_{10000}\}) \leftarrow$ *fitness* $(P_2(t) \{ind_1, \dots, ind_{10000}\})$ // Calculate fitness
$P_2(t) \leftarrow P_2(t) \{ind_1, \dots, ind_{10000}\}$    // Sort individuals
$P_2(t) \leftarrow P_2(t) \{ind_1, \dots ind_{1000}\}$          // Retain the 1000 best ones
                                    // Evolution of populations $P_1$ and $P_2$ until 0.1*gen generations
**for** each $P_i$
  current_generation ← 0
    **while** current_generation < 0.1*gen not met **do**
        $P_i(t) \{ind_{901}, \dots ind_{1000}\} \leftarrow P_i(t) \{ind_1, \dots, ind_{100}\}$  // Best 10% replace the worst 10%
        $P_i(t+1) \leftarrow P_i(t) \{ind_1, \dots, ind_{900}\}$
        $P_i(t+1) \leftarrow$ *pm* $(P_i(t+1) \{ind_1, \dots, ind_{90}\})$          // Parametric mutation (10% $P_i$ (t+1))
        $P_i(t+1) \leftarrow$ *sm* $(P_i(t+1) \{ind_{91}, \dots, ind_{900}\})$        // Structural mutation (90% $P_i$ (t+1))
        $f_i(P_i(t+1) \{ind_1, .. ind_{900}\}) \leftarrow$ *fitness* $(P_i(t+1) \{ind_1, \dots, ind_{900}\})$     // Evaluate
        $P_i(t+1) \leftarrow P_i(t+1) (ind_1, \dots, ind_{900}) \cup P_i(t)\{ind_{901}, \dots, ind_{1000}\}$
        $P_i(t+1) \leftarrow P_i(t+1)\{ind_1, \dots, ind_{1000}\}$            // Sort individuals
        current_generation ← current_generation + 1
    **end while**
**end for**
$P(t) \leftarrow P_1\{ind_1, \dots, ind_{500}\} \cup P_2\{ind_1, \dots, ind_{500}\}$        // Individuals of P has [neu, neu+1]
                                                  // nodes in the hidden layer
 $P(t) \leftarrow P(t) \{ind_1, \dots, ind_{1000}\}$            // Sort individuals by fitness: $ind_i > ind_{i+1}$
                                    // **Second Stage**
                                 // Input:  gen, neu+1
  t ← 0
  **while** stop criterion not met **do**                   // main loop
        $P(t) \{ind_{901}, \dots ind_{1000}\} \leftarrow P(t) \{ind_1, \dots, ind_{100}\}$   // Best 10% replace the worst 10%
        $P(t+1) \leftarrow P(t) \{ind_1, \dots, ind_{900}\}$
        $P(t+1) \leftarrow$ *pm* $(P(t+1) \{ind_1, \dots, ind_{90}\})$            // Parametric mutation (10% P(t+1))
        $P(t+1) \leftarrow$ *sm* $(P(t+1) \{ind_{91}, \dots, ind_{900}\})$          // Structural mutation (90% P(t+1))
        $f(P(t+1) \{ind_1, .. ind_{900}\}) \leftarrow$ *fitness* $(P(t+1) \{ind_1, \dots, ind_{900}\})$          // Evaluate
        $P(t+1) \leftarrow P(t+1) (ind_1, \dots, ind_{900}) \cup P(t) \{ind_{901}, \dots, ind_{1000}\}$
        $P(t+1) \leftarrow P(t+1)\{ind_1, \dots, ind_{1000}\}$            // Sort individuals
        t ← t+1
        last_generation ← t
  **end while**
  **return** *best* (P(last_generation) $\{ind_1\}$)
```

Fig. 2. Pseudo-code of the TSEA for a classification problem.

The TSEA loops are repeated until the maximum number of generations, in each case, is reached or until the best individual or the population mean fitness does not improve during gen-without-improving generations (20 in this paper).

3. Proposal description

Our attention is focused on evolutionary PUNNs for classification problems. The current paper presents TSEAFS methodology that is based on a combination between TSEA and a pre-processing stage. First of all, some feature selectors are applied independently to the training set of all data sets in order to obtain a list of attributes, for each of them, considered for training and test phases. In this way, two reduced sets (reduced training and test sets) are generated, where only most relevant features are included. It is important to point out that the FS is performed only with training data; the reduced test set has the same features as the reduced training set. These reduced sets are taken as input to TSEA. TSEAFS operates with four filters as independent feature selectors. As a result of the FS stage, a list of relevant features is obtained with each of the FS methods for each data set. Fig. 3 presents the framework of the proposed methodology. TSEAFS has two phases: (i) feature selection and (ii) classification by means of TSEA.

There are two different configurations in TSEA, named 1* and 2*. The TSEAFS features are the following: (a) PUNN have been employed, with a number of neurons in the input layer equal to the number of variables in the problem after FS; a hidden layer with a number of nodes that depends on the data set to be classified and the number of selected features; and the number of nodes in the output layer equal to the number of classes minus one because a softmax-type probabilistic approach has been used; (b) two different configurations (1*# and 2*#) are applied to subsets obtained with each of the selectors, for each data set. The parameters of each configuration are *neu#*, gen# and α_2. The first two ones take specific values depending on the data set and the last one depends on the configuration number (1*#,…). α_2 is related with the parametric mutation and acts on the coefficients of the output-layer (β_j^l). Moreover, α_2 parameter controls the diversity of the individuals in the population; an experimental study about it was performed in [4,22]. Table 1 shows the main aspects of TSEA/TSEAFS configurations.

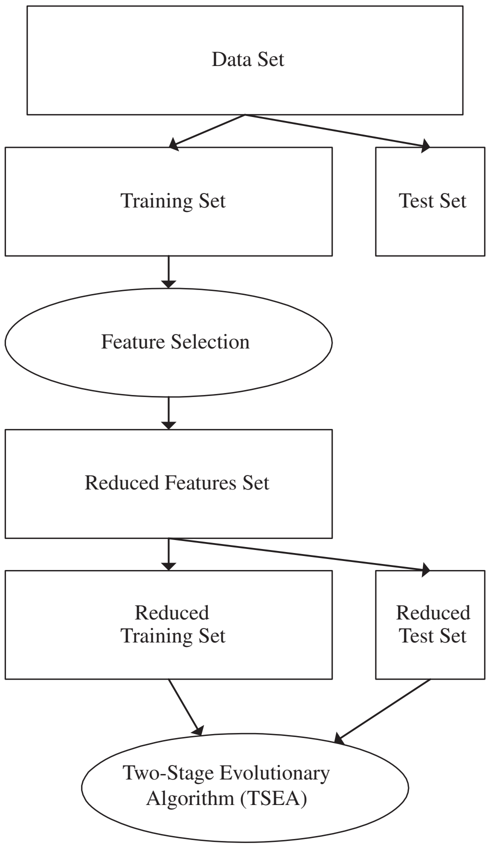


Fig. 3. TSEAFS framework.

Table 1
Description of the TSEA/TSEAFS configurations.

Methodology	Config.	Num. of neurons in each pop.	Size of each pop.	Num. gener. in each pop.	α_2
TSEA	1*	neu and neu+1	1000	0.1*gen	1
TSEA	2*	neu and neu+1	1000	0.1*gen	1.5
TSEAFS	1*#	neu# and neu#+1	1000	0.1*gen#	1
TSEAFS	2*#	neu# and neu#+1	1000	0.1*gen#	1.5

4. Experimentation

4.1. *Data sets, parameters and validation technique*

Table 2 describes the data sets employed. All of them, except the last one, are publicly available at the UCI repository [23]. The following 19 have been used: *Appendicitis, Breast* Cancer, Breast Tissue (*Breast-t*), *Cardiotocography*, Statlog (*Heart*), *Hepatitis*, Labor Relations, *Led24*, *Lymphography*, *Parkinsons*, *Pima* Indians diabetes, Steel *Plates* Faults, Molecular Biology (*Promoter* Gene Sequences), *SPECTF*, *Vowel*, *Waveform* database generator (version 2), Wine Quality (*Winequality-red*), *Yeast and Liver-transplantation, a real-world problem*.

These data sets are complex problems that report error rates in test accuracy about 20% or above with reference classifiers such as C4.5 [24] or 1-NN [25,26]. The last data set has an important problem with the distribution of the data and will be explained in their own section. Since we are using neural networks, all nominal variables have been converted to binary ones. Also, the missing values have been replaced in the case of nominal variables by the mode or, when concerning continuous variables, by the mean, taking into account the full data set. The experimental design uses the cross validation technique called stratified hold-out [27] that consists of splitting the data into two sets: training and test set, maintaining the class distribution of the samples in each set approximately equal as in the original data set. Their sizes are approximately *3N/4* and *N/4*, where *N* is the number of patterns in the problem [28]. Some data sets were prearranged in the repository, so we maintain the original distribution.

Regards to TSEA methodology, the concrete values of *neu* and *gen* parameters depend on the data set and are shown in the eighth column of Table 2. With respect to the number of generations, we have defined three kinds of values: small (150, 300), medium (500) and large (1000). We have given, in some cases, values of our choice to the two parameters depending on the complexity of the data set (number of classes, inputs, instances,…). Other times the values are based on a previous

Table 2
Summary of the 19 data sets used and parameter values for TSEA and TSEAFS methodologies.

Data set	Size	Train	Test	Features	Inputs	Classes	Neu; Gen	Neu#; Gen#
Appendicitis	106	80	26	7	7	2	4; 300	4; 100
Breast	286	215	71	9	15	2	9; 500	9; 300
Breast-t	106	81	25	9	9	6	5; 300	5; 150
Cardiotocography	2126	1594	532	23	31	3	6; 300	5; 150
Heart	270	202	68	13	13	2	6; 500	4; 20
Hepatitis	155	117	38	19	19	2	3; 300	3; 300
Labor	57	43	14	16	29	2	6; 300	5; 300
Led24	3200	200	3000	24	24	10	8; 500	8; 500
Lymphography	148	111	37	18	38	4	6; 500	6; 100
Parkinsons	195	146	49	23	22	2	6; 300	6; 300
Pima	768	576	192	8	8	2	4; 150	4; 150
Plates	1941	1457	484	27	27	7	6; 500	6; 500
Promoter	106	80	26	58	114	2	11; 500	6; 300
SPECTF	267	80	187	44	44	2	6; 500	6; 300
Vowel	990	528	462	12	11	11	6; 1000	6; 1000
Waveform	5000	3750	1250	40	40	3	3; 500	3; 500
Winequality-red	1599	1196	403	11	11	6	6; 300	4; 300
Yeast	1484	1112	372	8	8	10	11; 1000	11; 1000
Liver-transplantation	615	462	153	39	53	2	6; 500	6; 300

work [4]. In TSEAFS, again there are two parameters, neu# and gen#, whose value is defined for each data set. The last column of the Table 2 presents the values of them along with the ones of TSEA to have a general view of the differences. In TSEAFS the number of neurons is upper bounded by TSEA value. It is important to note that aforementioned values of the parameters concern to the base configuration (1*/1*#). The gen# parameter takes values similar to *gen* with the exception of Heart in whose case is very small (20) since the search converges quickly. The values of the remaining configurations are presented further on.

4.2. Filter-based feature selection methods

Table 3 depicts the methods used in the experimentation. There are four ones with and one without feature selection that belong respectively to TSEAFS (the current proposal) and TSEA methodologies. The feature selectors are filters. Last column defines an abbreviated name for each of them that is employed in next sections.

In a previous work, we proposed BIRS (Best Incremental Ranked Subset) [16] method. BIRS belongs to a hybrid category where the selection process is divided into two stages: in the first one, features are evaluated individually, providing a ranking based on a criterion; in stage two, a feature subset evaluator is applied to a certain number of features in the previous ranking following a search strategy. BIRS can use any evaluator in the two phases. In the cited work, BIRS uses as a subset evaluator CFS (Correlation-based Feature Selection) [7] and CNS (consistency based measure) [6] – that are established on correlation and consistency concepts – at the second phase, and SOAP (Selection Of Attributes by Projection) [29] measure and the own subset evaluator at the first phase as a ranking evaluator. The hybrid algorithm FCBF (Fast Correlation-Based Filter) uses symmetrical uncertainty (SU) in two steps. In the first step generate a ranking based on the SU between each feature and the class. Second step starts with a full set of features and begins eliminating some, that is, it finds the best subset using a backward selection technique with sequential search strategy, analyzing whether a feature is discarded or not depending on the feature–feature SU correlation. BestFirst [7] is a well-known search strategy use with CFS evaluation measure.

Table 3
List of methods employed in experimentation with and without feature selection.

Feature selector name	Ranking method	Subset evaluation	Methodology	Abb. name
–	None	None	TSEA	FS0
spBI _CFS	spBI	CFS	TSEAFS	FS1
cnBI _CNS	cnBI	CNS	TSEAFS	FS2
FCBF	Symmetrical Uncertainty	FCBF	TSEAFS	FS3
BestFirst_CFS	BestFirst	CFS	TSEAFS	FS4

sp stands for SOAP, BI for BIRS and cn for CNS

As previously mentioned, four FS methods implemented as filters have been applied to each data set. Table 4 illustrates for each data set the number of inputs of the original train set (see column labelled FS0) and those that have been obtained with the different feature selectors (see columns labelled FS1–4) along with the reduction percentage in the inputs of each selector compared to the original data set. Last row shows the average of the number of inputs and reduction percentage of the test bed for each experimented method on this paper.

The reduction percentage of the number of inputs is defined as:

$$\text{Reduction\_of\_Inputs}(\%) = \left(1 - \frac{\text{Inputs}(FSi)}{\text{Inputs}(FS0)}\right)100 \qquad i = 1,\ldots,4 \qquad (3)$$

where i is the FS method index and Inputs(j) represents the number of inputs of a given data set with method j.

In all cases, FS methods successfully decreased the data dimensionality by selecting, in mean, much less than the half of the original features.

5. Results

This section details the results obtained, measured in Correct Classification Ratio (CCR) in the test set or in the test subset depending on that FS has been considered or not. First of all, we present the results obtained with TSEA and TSEAFS. After that, a statistical analysis compares them to determine whether there are significant differences between applying or not FS. Next, the

Table 4
Number of inputs and reduction percentage for the 19 data sets with and without feature selection.

Data set	Inputs					Reduction (%)			
	FS0	FS1	FS2	FS3	FS4	FS1	FS2	FS3	FS4
Appendicitis	7	4	2	2	5	42.86	71.43	71.43	28.57
Breast	15	4	2	3	4	73.33	86.67	80.00	73.33
Breast-t	9	6	6	4	6	33.33	33.33	55.56	33.33
Cardiotocography	31	9	21	8	7	70.97	32.26	74.19	77.42
Heart	13	7	9	6	7	46.15	30.77	53.85	46.15
Hepatitis	19	10	5	6	10	47.37	73.68	68.42	47.37
Labor	29	7	5	8	8	75.86	82.76	72.41	72.41
Led24	24	6	6	6	6	75.00	75.00	75.00	75.00
Lymphography	38	11	9	8	12	71.05	76.32	78.95	68.42
Parkinsons	22	5	6	4	6	77.27	72.73	81.82	72.73
Pima	8	3	5	4	4	62.50	37.50	50.00	50.00
Plates	27	16	21	6	10	40.74	22.22	77.78	62.96
Promoter	114	7	7	11	10	93.86	93.86	90.35	91.23
SPECTF	44	12	9	6	12	72.73	79.55	86.36	72.73
Vowel	11	3	9	7	3	72.73	18.18	36.36	72.73
Waveform	40	14	15	5	14	65.00	62.50	87.50	65.00
Winequality-red	11	5	8	4	4	54.55	27.27	63.64	63.64
Yeast	8	5	7	6	7	37.50	12.50	25.00	12.50
Liver-transplantation	53	13	11	7	12	75.47	79.25	86.79	77.36
Average	27.24	7.65	8.82	6.00	7.65	62.94	53.95	68.07	62.17

Table 5
Results obtained in 18 data sets applying TSEA and TSEAFS.

Data set	Method	Topology	Mean ± SD	
			Config 1*/1*#	Config 2*/2*#
Appendicitis	FS0	7:[4,5]:1	**81.66 ± 4.24**	80.51 ± 2.84
	FS1	4:[4,5]:1	**82.82 ± 2.19**	81.02 ± 2.65
	FS2	2:[4,5]:1	80.89 ± 0.70	80.76 ± 0.00
	FS3	2:[4,5]:1	80.38 ± 2.73	79.61 ± 3.21
	FS4	5:[4,5]:1	81.79 ± 2.00	81.66 ± 2.97
Breast	FS0	15:[9,10]:1	**65.96 ± 2.89**	62.76 ± 3.08
	FS1	4:[9,10]:1	**69.85 ± 1.50**	68.21 ± 1.08
	FS2	2:[9,10]:1	69.01 ± 0.00	69.01 ± 0.00
	FS3	3:[9,10]:1	68.92 ± 0.73	69.10 ± 0.36
	FS4	4:[9,10]:1	69.01 ± 0.00	69.01 ± 0.00
Breast-t	FS0	9:[5,6]:5	54.53 ± 7.89	**55.33 ± 9.16**
	FS1	6:[5,6]:5	54.40 ± 6.77	48.93 ± 8.83
	FS2	6:[5,6]:5	55.73 ± 8.72	57.87 ± 6.87
	FS3	4:[5,6]:5	**60.93 ± 4.77**	56.53 ± 7.03
	FS4	6:[5,6]:5	59.47 ± 8.90	56.00 ± 6.96
Cardiotocography	FS0	31:[6,7]:2	**81.69 ± 3.56**	81.55 ± 2.90
	FS1	9:[5,6]:2	**85.26 ± 2.27**	84.88 ± 2.11
	FS2	21:[5,6]:2	71.20 ± 2.55	76.71 ± 1.04
	FS3	8:[5,6]:2	81.55 ± 1.69	81.12 ± 1.80
	FS4	7:[5,6]:2	81.58 ± 2.48	82.38 ± 2.42
Heart	FS0	13:[6,7]:1	76.62 ± 2.33	**77.45 ± 3.09**
	FS1	7:[4,5]:1	77.45 ± 2.16	77.69 ± 2.28
	FS2	9:[4,5]:1	**78.57 ± 1.99**	77.79 ± 1.60
	FS3	7:[4,5]:1	75.24 ± 2.70	75.34 ± 2.80
	FS4	7:[4,5]:1	77.45 ± 2.16	77.69 ± 2.28
Hepatitis	FS0	19:[3,4]:1	82.10 ± 4.44	**87.01 ± 3.78**
	FS1	10: [3,4]:1	90.78 ± 1.79	89.29 ± 1.53
	FS2	5: [3,4]:1	86.14 ± 1.81	87.45 ± 1.49
	FS3	6: [3,4]:1	85.00 ± 1.56	**91.05 ± 2.55**
	FS4	10: [3,4]:1	90.78 ± 1.79	89.29 ± 1.53
Labor	FS0	29:[6,7]:1	85.24 ± 8.78	**86.90 ± 5.96**
	FS1	7:[5,6]:1	93.09 ± 4.39	**96.19 ± 4.08**
	FS2	5:[5,6]:1	87.62 ± 4.16	88.33 ± 4.39
	FS3	8:[5,6]:1	90.95 ± 5.28	90.48 ± 5.73
	FS4	8:[5,6]:1	89.76 ± 6.41	89.76 ± 5.20
Led24	FS0	24:[8,9]:9	50.29 ± 6.59	**51.03 ± 5.58**
	FS1	6:[8,9]:9	67.26 ± 1.46	**68.30 ± 0.57**
	FS2	6:[8,9]:9	67.26 ± 1.46	**68.30 ± 0.57**
	FS3	6:[8,9]:9	67.26 ± 1.46	**68.30 ± 0.57**
	FS4	6:[8,9]:9	67.26 ± 1.46	**68.30 ± 0.57**
Lymphography	FS0	38:[6,7]:3	**79.37 ± 4.73**	78.73 ± 4.79
	FS1	11:[6,7]:3	79.09 ± 5.71	78.55 ± 4.42
	FS2	9:[6,7]:3	80.18 ± 3.27	80.36 ± 4.54
	FS3	8:[6,7]:3	79.18 ± 5.17	80.61 ± 3.12
	FS4	12:[6,7]:3	78.19 ± 3.88	**80.90 ± 5.71**
Parkinsons	FS0	22:[6,7]:1	73.94 ± 2.43	**78.09 ± 3.51**
	FS1	5:[6,7]:1	78.36 ± 2.86	78.77 ± 1.66
	FS2	6:[6,7]:1	80.13 ± 2.26	80.06 ± 3.73
	FS3	4:[6,7]:1	82.52 ± 2.92	**82.79 ± 2.50**
	FS4	6:[6,7]:1	79.25 ± 2.15	76.05 ± 3.47
Pima	FS0	8:[4,5]:1	78.38 ± 1.59	**79.21 ± 1.53**
	FS1	3:[4,5]:1	79.35 ± 1.09	**79.72 ± 1.08**
	FS2	5:[4,5]:1	78.52 ± 0.80	78.54 ± 1.37
	FS3	4:[4,5]:1	78.42 ± 1.35	79.53 ± 0.98
	FS4	4:[4,5]:1	78.42 ± 1.35	79.53 ± 0.98
Plates	FS0	27:[6,7]:6	50.74 ± 4.24	**51.46 ± 3.03**
	FS1	16:[6,7]:6	53.81 ± 3.99	53.38 ± 4.17
	FS2	21:[6,7]:6	**56.93 ± 2.43**	54.40 ± 4.95
	FS3	6:[6,7]:6	50.87 ± 4.75	51.87 ± 3.06
	FS4	10:[6,7]:6	48.84 ± 4.60	48.53 ± 2.58
Promoter	FS0	114:[11,12]:1	65.76 ± 8.99	**68.20 ± 9.52**
	FS1	7:[6,7]:1	83.84 ± 3.83	**85.64 ± 4.03**
	FS2	7:[6,7]:1	80.00 ± 2.74	76.30 ± 4.10
	FS3	11:[6,7]:1	73.66 ± 6.77	75.12 ± 4.48
	FS4	10:[6,7]:1	74.74 ± 5.11	73.97 ± 3.73
SPECTF	FS0	44:[6,7]:1	60.17 ± 4.15	**61.56 ± 4.97**
	FS1	12:[6,7]:1	73.20 ± 2.18	73.85 ± 2.71
	FS2	9:[6,7]:1	72.07 ± 1.16	71.64 ± 1.56
	FS3	6:[6,7]:1	**73.99 ± 1.30**	70.60 ± 1.84
	FS4	12:[6,7]:1	72.35 ± 1.69	73.76 ± 1.02
Vowel	FS0	11:[6,7]:10	45.04 ± 2.93	**47.18 ± 4.03**
	FS1	3:[6,7]:10	48.07 ± 3.11	**54.31 ± 2.29**
	FS2	9:[6,7]:10	47.65 ± 5.01	46.80 ± 4.27

Table 5 (*continued*)

Data set	Method	Topology	Mean ± SD	
			Config 1*/1*#	Config 2*/2*#
	FS3	7:[6,7]:10	48.12 ± 3.40	49.45 ± 2.54
	FS4	3:[6,7]:10	48.07 ± 3.11	**54.31 ± 2.29**
Waveform	FS0	40:[3,4]:2	**84.46 ± 0.92**	82.01 ± 1.48
	FS1	14:[3,4]:2	86.35 ± 0.85	**86.89 ± 0.71**
	FS2	15:[3,4]:2	86.02 ± 2.16	85.67 ± 0.96
	FS3	5:[3,4]:2	79.96 ± 0.47	80.67 ± 0.37
	FS4	14:[3,4]:2	86.35 ± 0.85	**86.89 ± 0.71**
Winequality-red	FS0	11:[6,7]:5	60.95 ± 1.58	**61.11 ± 1.02**
	FS1	5:[4,5]:5	61.63 ± 1.09	61.25 ± 1.62
	FS2	8:[4,5]:5	61.47 ± 0.95	60.87 ± 1.29
	FS3	4:[4,5]:5	**61.65 ± 0.95**	60.95 ± 0.91
	FS4	5:[4,5]:5	61.63 ± 1.09	61.25 ± 1.62
Yeast	FS0	8:[11,12]:9	60.05 ± 1.21	**60.16 ± 1.10**
	FS1	5:[11,12]:9	59.25 ± 1.44	60.06 ± 1.09
	FS2	7:[11,12]:9	**60.78 ± 1.29**	59.43 ± 1.29
	FS3	6:[11,12]:9	58.29 ± 1.18	57.91 ± 1.32
	FS4	7:[11,12]:9	**60.78 ± 1.29**	59.43 ± 1.29

proposal is evaluated in a real-world problem related to liver-transplantation in Spain.

5.1. *Results applying TSEA and TSEAFS*

The results obtained by applying TSEA methodology [4] are presented, along with those obtained with TSEAFS. Table 5 shows the mean and standard deviation (SD) of the test accuracies for each data set for a total of 30 runs. The best results without and with FS appear in boldface for each data set. From the analysis of the data, it can be concluded, from a purely descriptive point of view, that the TSEAFS methodology obtains best results for all data sets. In most of cases, the SD reduction with TSEAFS is clear and it expresses more homogeneous results compared to TSEA.

5.1.1. *Statistical analysis*

We follow the recommendations pointed out by J. Demšar [30] to perform non-parametric statistical tests. To determine the statistical significance of the differences in rank observed for each method with all data sets, a non-parametric test might be used. There are two methods, Friedman [31] and Iman-Davenport [32] tests. The former test is equivalent to the repeated-measures ANOVA and is based on χ_F^2 statistic; the null hypothesis states that all algorithms perform equal, so a rejection of it implies the existence of significant differences. The latter test is a derivation of the former based on F_F which is a better statistic, derived from χ_F^2, and is not undesirably conservative. F_F is distributed according to the F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom with k algorithms and N data sets. If the null-hypothesis is rejected, we can proceed with a post-hoc test. Bonferroni–Dunn [33] has been performed. It compares some methods with a control method. The critical difference (CD) can be computed from critical values – that can be found in any statistical book –, k and N. The considered significance levels have been 0.05 for Iman-Davenport test, and 0.05 and 0.10 for the post-hoc methods.

The average ranks of all methods without (FS0) and with FS (FS1–4), taking into account the best average between the two configurations, are respectively 4.33, 2.28, 3.14, 2.72 and 2.53. According to Iman-Davenport test results, since the $F_F = 6.03$ statistic is higher than the critical value at $\alpha = 0.05$ ($F(4,68) = 2.51$) the null-hypothesis is rejected. Therefore, we apply a post-hoc Bonferroni–Dunn test that compares a number of methods with a control method, by determining whether the average ranks differ by at least the CD. In our case, we make a comparison of the methods that employ FS (FS1–4) versus the control method (FS0) that does not use FS. Table 6 shows the Bonferroni–Dunn test results where the ranking difference, the CD (at $\alpha = 0.05$ and $\alpha = 0.10$) and the detected significant difference level. Next, the Bonferroni–Dunn test results are analysed and these enable us to ascertain the following. There are significant differences between TSEA applying each of the FS methods and without FS. The statistical tests points out that PUNN performance improves significantly pre-processing the data set with any of the FS methods employed in this paper. However, FS1, FS3 and FS4 are better regarding to statistical significance level.

Table 6
Critical difference values and ranking differences of TSEA and TSEAFS by means of a Bonferroni–Dunn test (FS0 is the control method).

FS0 vs.	**Ranking difference** (control method–compared method)	**Significant for compared method**
FS1	2.05	[a]
FS2	1.19	[b]
FS3	1.61	[a]
FS4	1.80	[a]
	$CD(\alpha = 0.05) = 1.32; CD(\alpha = 0.10) = 1.18$	

[a] Statistically significant difference with $\alpha = 0.05$.
[b] Statistically significant difference with $\alpha = 0.10$.

5.1.2. *Results obtained with a variety of classifiers*

Now, a comparison is performed between TSEA and other machine learning algorithms. These methods are C4.5, k-nearest neighbours (k-NN), -where k is 1-, SVM [34], PART [35], the MLP model [3] with a learning Back-Propagation method (BP) and the RBF model [36]. Since, C4.5, 1-NN, SVM, PART, MLP and RBF are implemented in WEKA tool [37], we have used the same cross-validation, thus the same instances in each of the partitions, whose results were shown in Table 5. Regarding the parameters, for BP were the following: learning rate $\eta = 0.3$ and momentum $\alpha = 0.2$. The remaining algorithms have been run with the WEKA default values. The number of runs for MLP and RBF was 30, thus the results are averaged. We have reported in Table 7 the results without and with FS for each data set and algorithm. For each filter, the best average appears in boldface and the second best one in italics.

Table 7
Results obtained in 18 data sets for several classifiers with and without feature selection.

Data set	Method	C4.5	1-NN	SVM	PART	MLP	RBF	TSEAFS
Appendicitis	FS0	73.08	69.23	84.62	73.08	76.92	74.67	81.66
	FS1	80.77	69.23	76.92	80.77	78.85	80.00	82.82
	FS2	76.92	57.69	76.92	76.92	77.95	77.05	80.89
	FS3	80.77	80.77	80.77	80.77	80.77	74.49	80.38
	FS4	80.77	65.38	76.92	80.77	79.23	79.36	81.79
Breast	FS0	70.42	64.79	64.79	69.01	60.80	68.78	65.96
	FS1	69.01	70.42	66.20	71.83	69.01	67.46	69.85
	FS2	69.01	70.42	64.79	69.01	69.01	69.01	69.01
	FS3	69.01	70.42	64.79	69.01	69.53	67.65	69.10
	FS4	69.01	70.42	66.20	71.83	69.01	67.46	69.01
Breast-t	FS0	52.00	60.00	52.00	44.00	63.20	61.20	55.33
	FS1	56.00	52.00	60.00	44.00	65.33	58.67	54.40
	FS2	52.00	52.00	64.00	52.00	67.20	61.20	57.87
	FS3	48.00	48.00	56.00	48.00	65.60	60.40	60.93
	FS4	68.00	56.00	60.00	56.00	65.47	60.67	59.47
Cardiotocography	FS0	82.71	76.32	83.65	82.52	80.75	81.80	81.69
	FS1	77.07	81.77	81.20	82.52	81.94	83.40	85.26
	FS2	75.19	63.91	75.19	75.00	68.29	65.91	76.71
	FS3	77.82	81.20	81.20	77.26	80.13	80.50	81.55
	FS4	78.38	80.45	81.39	81.20	80.86	84.12	82.38
Heart	FS0	70.59	73.53	76.47	73.53	74.85	78.53	77.45
	FS1	73.53	73.53	76.47	77.94	72.50	78.24	77.69
	FS2	72.06	75.00	76.47	75.00	74.85	77.60	78.57
	FS3	73.53	70.59	77.94	75.00	74.90	76.37	75.34
	FS4	73.53	73.53	76.47	77.94	72.50	78.53	77.69
Hepatitis	FS0	84.21	86.84	89.47	81.58	84.73	89.30	87.01
	FS1	84.21	89.47	86.84	84.21	87.28	89.30	90.78
	FS2	89.47	84.21	89.47	84.21	84.21	88.42	87.45
	FS3	89.47	84.21	89.47	86.84	87.72	90.79	91.05
	FS4	84.21	89.47	86.84	84.21	87.28	89.30	90.78
Labor	FS0	85.71	71.43	78.57	85.71	69.52	71.67	86.90
	FS1	85.71	71.43	78.57	85.71	64.29	71.43	96.19
	FS2	85.71	64.28	78.57	78.57	78.57	64.29	88.33
	FS3	85.71	78.57	71.43	78.57	71.43	64.29	90.95
	FS4	85.71	64.29	71.43	85.71	57.62	71.43	89.76
Led24	FS0	65.67	39.43	58.97	55.80	57.48	55.14	51.03
	FS1	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	FS2	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	FS3	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	FS4	68.10	67.90	67.93	68.50	68.44	67.42	68.30
Lymphography	FS0	75.68	83.78	91.89	75.68	86.58	70.99	79.37
	FS1	88.29	78.38	83.78	70.27	73.24	68.92	79.09
	FS2	75.68	70.27	78.38	64.86	71.89	75.77	80.36
	FS3	81.08	75.68	81.08	70.27	74.50	69.64	80.61
	FS4	81.08	81.08	81.08	64.86	80.45	69.16	80.90
Parkinsons	FS0	71.43	77.55	75.51	75.51	77.62	70.27	78.09
	FS1	75.51	79.59	75.51	77.55	81.56	77.75	78.77
	FS2	79.59	79.59	75.51	81.63	75.65	73.47	80.13
	FS3	81.63	73.47	79.59	77.55	84.83	80.27	82.79
	FS4	73.47	81.63	75.51	79.59	83.13	77.55	79.25
Pima	FS0	74.48	73.96	78.13	74.48	75.94	77.34	79.21
	FS1	76.04	74.48	77.60	76.04	78.18	79.17	79.72
	FS2	74.48	67.19	78.65	74.48	76.89	75.64	78.54
	FS3	76.04	67.71	79.17	76.04	79.01	80.28	79.53
	FS4	76.04	67.71	79.17	76.04	78.73	80.28	79.53
Plates	FS0	39.05	49.17	57.02	46.69	53.50	59.94	51.46
	FS1	40.50	51.24	51.03	46.90	56.71	64.08	53.81
	FS2	38.22	50.62	55.17	44.63	55.24	62.17	56.93
	FS3	44.63	43.18	45.04	49.79	52.85	55.88	51.87
	FS4	54.75	47.31	51.65	51.65	57.33	59.88	48.84
Promoter	FS0	69.23	65.38	88.46	53.85	86.03	79.36	68.20
	FS1	73.08	57.69	84.62	80.77	84.49	83.46	85.64
	FS2	80.77	57.69	84.62	76.92	75.64	85.00	80.00
	FS3	73.08	76.92	73.08	80.77	78.21	79.74	75.12
	FS4	73.08	69.23	73.08	80.77	76.28	80.00	74.74
SPECTF	FS0	67.91	61.50	72.19	70.59	71.28	76.19	61.56
	FS1	66.84	59.36	72.19	72.19	73.67	76.24	73.85
	FS2	65.78	60.96	70.05	65.78	70.02	74.60	72.07
	FS3	67.91	59.36	65.24	64.71	69.57	74.58	73.99
	FS4	66.84	57.75	73.26	70.05	72.26	74.63	73.76
Vowel	FS0	39.39	48.48	45.45	38.53	45.87	47.25	47.18
	FS1	45.24	46.54	54.33	44.59	52.79	43.12	54.31
	FS2	38.53	51.52	48.48	40.04	52.05	44.73	47.65
	FS3	41.56	46.97	41.34	36.58	44.97	46.95	49.45
	FS4	45.24	46.54	54.33	44.59	52.79	43.12	54.31

Table 7 (*continued*)

Data set	Method	C4.5	1-NN	SVM	PART	MLP	RBF	TSEAFS
Waveform	FS0	74.80	68.96	86.24	76.88	84.85	87.29	84.46
	FS1	74.40	75.36	86.88	77.04	83.21	82.24	86.89
	FS2	74.40	76.64	87.12	79.68	86.27	82.22	86.02
	FS3	74.72	69.12	78.80	74.00	77.57	76.88	80.67
	FS4	74.40	75.36	86.88	77.04	83.21	82.24	86.89
Winequality-red	FS0	53.85	49.88	59.55	51.36	56.35	57.11	61.11
	FS1	50.87	48.88	59.80	52.11	59.36	59.00	61.63
	FS2	50.12	49.63	58.81	52.85	57.04	59.19	61.47
	FS3	51.36	50.37	59.31	49.13	59.64	59.17	61.65
	FS4	50.87	48.88	59.80	52.11	59.36	59.00	61.63
Yeast	FS0	54.84	48.39	55.91	56.72	59.94	58.31	60.16
	FS1	53.49	48.92	54.03	54.84	60.20	58.48	60.06
	FS2	54.03	49.46	54.84	54.30	60.20	58.91	60.78
	FS3	52.69	48.12	51.61	52.96	58.96	58.78	58.29
	FS4	54.03	49.46	54.84	54.30	60.20	58.91	60.78
Average	FS0	66.95	64.92	72.16	65.86	70.34	70.29	69.88
	FS1	68.81	66.46	*71.88*	69.32	71.73	71.58	**74.39**
	FS2	67.78	63.83	*71.39*	67.47	70.52	70.14	**72.84**
	FS3	68.73	66.25	69.10	67.54	*71.03*	70.23	**72.87**
	FS4	69.86	66.25	70.93	69.84	*71.34*	71.28	**73.32**

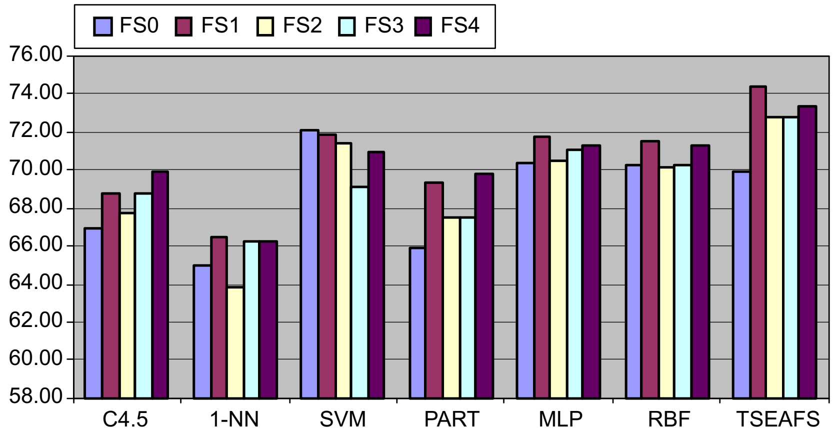


Fig. 4. Overall results in 18 data sets for each classifier with and without feature selection.

From a purely descriptive analysis of the results, we can assert the following. Focusing on FS, it can be concluded that the TSEA method obtains the best result for 7 out of 18 data sets. Furthermore, TSEA reports the highest mean accuracy (74.39%) followed by SVM (71.88%) that indicates the excellent performance of the product units. The important achievement of the FS combined with TSEA lets the proposed methodology, TSEAFS, to improve the accuracy very much. Fig. 4 shows a bar plot of the overall results obtained for each classifier and filter.

5.2. *Application of TSEAFS to a real-world liver-transplantation problem*

Liver-transplantation is strongly limited by the availability of proper liver donors. The imbalance between demand and supply is unfortunately followed by the terrible scenario of waiting list deaths. Several efforts have been made for successful donor pool expansion and the prioritization of recipients on waiting lists. Donor and graft acceptance – considering organ shortage and pool expansion –, prioritization of candidates – including waiting list mortality – and allocation policy – combining equity, utility and efficiency principles – depict a complex scenario that is not easy to model. More than 100 variables can be considered in a particular clinical decision for donor and organ acceptance, allocation and donor-recipient “best matching”.

A multicentered retrospective analysis from 11 Spanish units of liver-transplantation was conducted, including all the consecutive liver transplants performed between January 1, 2007 and December 31, 2008. All transplant recipients aged 18 years or older were included. Recipient and donor characteristics were reported at the time of transplant. 19 recipient characteristics, 20 donor characteristics and 3 operative factors were reported for each donor-recipient pair, D–R. The end-point variable for classification was 3-month graft mortality. A total of 1031 liver transplants were initially included. The follow-up period was fulfilled in 1003 liver transplants. 28 cases were excluded because of the absence of graft survival data.

The acceptance model consists of a neural network based on product units to predict the probability of graft survival 3 months following liver-transplantation. This model tries to maximize the probability that a D–R pair has belonged to the “graft survival” class.

The data set contains 615 instances, which are imbalanced with a 1:8 ratio. Thus, this data set can generate distorted models for many learning algorithms for which (i) the impact of some factors can be hidden and (ii) the prediction accuracy can be misleading. This is due to the fact that most data mining

Table 8
Results obtained in liver-transplantation problem with and without feature selection.

Classifier	Measures	Method				
		FS0	FS1	FS2	FS3	FS4
C4.5	CCR	89.54	89.54	89.54	89.54	89.54
	MS	0.00	0.00	0.00	0.00	0.00
1-NN	CCR	73.78	85.62	82.35	32.06	86.92
	MS	0.00	18.75	**31.25**	68.75	25.00
SVM	CCR	89.54	89.54	89.54	89.54	89.54
	MS	0.00	0.00	0.00	0.00	0.00
PART	CCR	80.39	87.58	88.89	89.54	84.96
	MS	0.00	0.00	12.50	0.00	6.25
MLP	CCR	83.70	85.86	87.67	89.54	87.32
	MS	4.16	4.38	10.83	0.00	4.38
RBF	CCR	89.54	88.89	89.39	89.52	89.37
	MS	0.00	0.00	1.25	0.00	0.80
TSEA	CCR	88.69	87.27	**89.71**	89.54	89.25
	MS	2.91	9.37	11.46	9.58	9.75

algorithms assume balanced data sets. When dealing with imbalanced data sets, there are two alternatives, either (i) sampling or balancing techniques: over–sampling algorithms aimed at balancing the class distribution increasing the minority class, or under–sampling algorithms that balance the class removing instances from the majority classes; and (ii) to apply algorithms that are robust to this problem.

Moreover, in order to measure the performance of the classifier, we also consider the minimum sensitivity (MS) of the test set (or subset). It is very important to evaluate successfully instances of the class with the lowest number of instances. A good classifier must reach a high CCR and classify correctly as many as possible of the minority class. For instance, a classifier can identify all instances of the majority class and none of the other class, thus the CCR would very high.

A technique called SMOTE [38] is applied in the training set in order to try to balance the subset. After that, the feature selection is applied over the new training set. Once we have the list of selected features, we use only these features of the original train and test sets.

In this second experiment, we have used the same filters of the previous experiment. The number of selected features is depicted in the last part of Table 4. Next, we present in Table 8 the results with different classifiers and filters. For TSEA and TSEAFS we only report the best result of the two configurations.

TSEA obtains the best CCR with filter FS2, and their MS is 11.46; it means that some instances of the minority class are well detected. With filter FS2, 1-NN gets the highest value, 31.25%, for MS; however, the CCR is 82.35, so a hit in minority class needs more than three errors in majority class. Other classifiers do not classify properly any instance of the minority class. FS helps to TSEA to classify some instances while maintaining the CCR.

6. Conclusions

This paper presented a methodology to enhance a classifier based on two-stage evolutionary algorithm in product unit neural networks in low performance problems. Specifically, a mixture of our previous TSEA methodology and FS, called TSEAFS, has been introduced. FS is performed by means of filters. The models obtained with the proposal have the advantages that are more accurate and less complex, taking into consideration the number of inputs and/or the number of nodes in the hidden-layer. Also, the current proposal is much more efficient, the reduction of the input size is about 55 %.

An empirical study on 18 UCI classification problems, that present test error rates about 20 % or above with C4.5 or 1-NN classifiers, has been performed to compare TSEAFS and TSEA methodologies, both of them based on evolutionary artificial product unit neural networks. The average accuracy has reached about 74% starting close to 70%. The statistical analysis reveals that differences are significant in favour for any considered filter.

Also other state-of-the-art classifiers have been tested with the 19 (18 from UCI repository and a real-world liver-transplantation problem) data sets in order to get an overall outlook.

Nonparametric statistical tests have been applied and the main conclusions achieved are as follows. The considered FS methods help to improve significantly the accuracy of the models with product units in all cases. The filters with the best average ranks are, in this order, FS1 (spBI_CFS), FS4 (BestFirst_CFS), and FS3 (FCBF).

In regard to the comparison with other classifiers, TSEAFS gets the best average results in 7 out of the 18 data sets. The liver-transplantation problem throws new issues like the sturdiness of the classifier based on product units.

Acknowledgements

This work has been partially subsidized by TIN2007-68084-C02-02, TIN2008-06681-C06-03 and TIN2011-28956-C02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745, TIC-7528 and TIC-7508 projects of the "Junta de Andalucía" (Spain).

References

[1] R.O. Duda, P.E. Hart, D. Stork, Pattern Classification, second ed., Wiley, 2001.
[2] P. Bonissone, J.M. Cadenas, M.C. Garrido, R.A. Díaz-Valladares, A fuzzy random forest, Int. J. Approx. Reason 51 (7) (2010) 729–747.
[3] C.M. Bishop, Neural networks for pattern recognition, Oxford University Press, New York, 1995.
[4] A.J. Tallón-Ballesteros, C. Hervás-Martínez, A two-stage algorithm in evolutionary product unit neural networks for classification, Expert Syst. Appl. 38 (1) (2011) 743–754.
[5] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Anal. 1 (3) (1997) 131–156.
[6] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1–2) (2003) 155–176.
[7] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Morgan Kaufmann, San Francisco, CA, 2000, pp. 359–366.
[8] R. Kohavi, G. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[9] E. Xing, M. Jordan, R. Karp, Feature selection for high-dimensional genomic microarray data, in: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2001, pp. 601–608.
[10] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[11] Y. Saeys, T. Abeel, Y.V. de Peer, Robust feature selection using ensemble feature selection techniques, in: ECML/PKDD, vol. 2, 2008, pp. 313–325.
[12] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1–2) (1997) 245–271.
[13] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[14] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[15] M. Dash, H. Liu, H. Motoda, Consistency based feature selection, in: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 98–109.
[16] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recognition 39 (12) (2006) 2383–2392.
[17] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.
[18] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, IEEE Comput. Soc. Bioinformatics (2003) 523–529.

[19] R. Durbin, D. Rumelhart, Products units: a computationally powerful and biologically plausible extension to back-propagation networks, Neural Comput. 1 (1) (1989) 133–142.

[20] F.J. Martínez-Estudillo, C. Hervás-Martínez, P.A. Gutiérrez, A.C. Martínez-Estudillo, Evolutionary product-unit neural networks classifiers, Neurocomputing 72 (1–3) (2008) 548–561.

[21] A.C. Martínez-Estudillo, F.J. Martínez-Estudillo, C. Hervás-Martínez, N. García-Pedrajas, Evolutionary product unit based neural networks for regression, Neural Networks 19 (2006) 477–486.

[22] A.J. Tallón-Ballesteros, P.A. Gutiérrez-Peña, C. Hervás-Martínez, Distribution of the search of evolutionary product unit neural networks for classification, in: Proceedings of the IADIS International Conference on Applied Computing (AC 2007), IADIS, Salamanca, Spain, 2007, pp. 266–273.

[23] A. Frank, A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], University of California, School of Information and Computer Science Irvine, CA, 2010

[24] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[25] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[26] D. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.

[27] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 1995), Vol. 2, Morgan Kaufmann, Montreal, Quebec, Canada, 1995, pp. 1137–1145.

[28] L. Prechelt, Proben1—A Set of Neural Network Benchmark Problems and Benchmarking Rules, Technical Report 21/94, Fakultat für Informatik. University of Karlsruhe, Karlsruhe, Germany, 1994.

[29] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Projection-based measure for efficient feature selection, J. Intell. Fuzzy Syst. 12 (3-4) (2002) 175–183.

[30] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[31] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Statist. Assoc. 32 (200) (1937) 675–701.

[32] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, Commun. Statist. A9 (6) (1980) 571–595.

[33] O.J. Dunn, Multiple comparisons among means, J. Am. Statist. Assoc. 56 (293) (1961) 52–64.

[34] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

[35] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Morgan Kaufmann, Madison, Wisconsin, USA, 1998, pp. 144–151.

[36] R.J. Howlett, L.C. Jain, Radial Basis Function Networks 1: Recent Developments in Theory and Applications, Springer, Heidelberg, Germany, 2001.

[37] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (2009) 10–18.

[38] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Int. Res. 16 (1) (2002) 321–357.