Filter-based feature selection in the context of evolutionary neural networks in supervised machine learning

Antonio J. Tallón-Ballesteros[1] · **José C. Riquelme**[1] · **Roberto Ruiz**[2]

Abstract
This paper presents a workbench to get simple neural classification models based on product evolutionary networks via a prior data preparation at attribute level by means of filter-based feature selection. Therefore, the computation to build the classifier is shorter, compared to a full model without data pre-processing, which is of utmost importance since the evolu-tionary neural models are stochastic and different classifiers with different seeds are required to get reliable results. Feature selection is one of the most common techniques for pre-processing the data within any kind of learning task. Six filters have been tested to assess the proposal. Fourteen (binary and multi-class) difficult classification data sets from the University of California repository at Irvine have been established as the test bed. An empirical study between the evolutionary neural network models obtained with and without feature selection has been included. The results have been contrasted with non-parametric statistical tests and show that the current proposal improves the test accuracy of the previous models significantly. Moreover, the current proposal is much more efficient than the previous methodology; the time reduction percentage is above 40%, on average. Our approach has also been compared with several classifiers both with and without feature selection in order to illustrate the performance of the different filters considered. Lastly, a statistical analysis for each feature selector has been performed providing a pairwise comparison between machine learning algorithms.

Keywords Artificial neural networks · Feed-forward · Evolutionary programming · Classification · Feature selection · Filters

1 Introduction

Classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representations is often referred to as pattern recognition [52]. A possible taxonomy of classifiers [63] may distinguish among artificial neural networks, decision trees, rule-based classifiers, classifiers based on nearest neighbours and so on. There are many domains, such as medicine [22], atmospheric sciences [39], computer vision, remote sensing [27], finance, molecular biology [68] and veterinary [43], where supervised learning has been successfully applied [35].

✉ Antonio J. Tallón-Ballesteros
atallon@us.es

[1] Department of Languages and Computer Systems, University of Seville, Reina Mercedes Avenue, 41012 Seville, Spain

[2] Area of Computer Science, Pablo de Olavide University, Km. 1, Utrera Road, 41013 Seville, Spain

An ideal learning algorithm would use only attributes that lead to good generalisation so there would be no need for attribute selection methods. Unfortunately, many learning procedures have their own biases and weaknesses, and given excess attributes will often have detrimental consequences. Attribute selection procedures are one way of mitigating the inability of a learning procedure to properly deal with excess attributes [11].

Computational intelligence is the synergistic interplay of soft computing techniques such as neural networks, genetic algorithms, fuzzy logic and artificial life [21]. There is a widespread myth among most of the researchers that neural networks are capable of dealing with large amounts of noise and useless data. This is true to a certain extent, but it is also true that the cleaner and more descriptive the data is, the better the neural networks will perform [51]. The aforementioned paper which shows that feature selection does increase the accuracy is very interesting. Thus, the sentence "less is more", that gives title, in the feature selection context, to the opening chapter by Liu and Motoda of the book

[46] written by themselves, is also true in the scope of neural classification. The current paper puts forth both ideas up for approval, and these are certainly fulfilled. Additionally, feature selection favours the scalability of the problem and new features could be measured in order to a get a more accurate neural network model [47].

The purpose of feature selection is to determine a subset from all input variables which could lead to an equal or, preferably, better performance of the classifier compared with the model containing all the problem variables. Theoretically, having more features should give us more discriminating power. However, the real world provides us with many reasons as to why this is generally not the case [42]: (1) the induction algorithm complexity grows dramatically with the number of features and (2) the irrelevant and redundant features also cause problems in the classification context as they may confuse the learning algorithm by helping to obscure the distributions of the small set of truly relevant features for the task at hand.

Our goal is to improve the accuracy and the simplicity of the classification models based on product unit neural networks trained with an evolutionary approach by means of filter-based feature selection methods and to determine the more proper methods to pre-process the data sets in order to obtain more accurate and more compact models. More concretely, we introduce a workbench to get simple neural classification models via a prior data preparation at the feature level applying feature selection. We have used, among others, filters based on correlation, consistency or information measures. The training of the neural network classifier, containing product units (PUs) as hidden neurons, is performed by means of an evolutionary programming algorithm (EPA).

This paper is organised as follows: Sect. 2 reviews some concepts about Product Unit Neural Networks (PUNNs), training algorithms for neural networks, the baseline EPA to train PUNN and FS; Sect. 3 describes our proposal; Sect. 4 details the experimentation process; Sect. 5 shows and analyses the results obtained; finally, Sect. 6 states the concluding remarks.

2 Foundations

2.1 Product unit neural networks

Depending on the architecture, artificial neural networks can be divided into two types: feed-forward and recurrent neural networks [36]. Among the huge amount of the literature

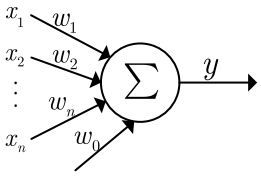

Fig. 1 Representation of an additive unit

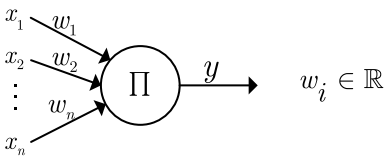

Fig. 2 Representation of a product unit

related to neural networks, 95% of publications are concerned with feed-forward ones [34]. According to the kind of unit or node, there are two feed-forward neural network models:

(a) Additive model. The network is composed of additive units. The output of each unit is function of the weighted inputs including the weights plus the activation value or bias. This model is depicted in Fig. 1.

The mathematical equation of an additive unit is given by:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i; \quad w_0, w_i \in \mathbb{R} \tag{1}$$

(b) Multiplicative model. This kind of neural net contains units that multiply their inputs instead of adding them. In this model, the more general approach is the product unit, proposed by Durbin and Rumelhart [20]. Figure 2 shows a PU.

Mathematically, a PU is expressed as follows:

$$y = x_1^{w_1} \cdot x_2^{w_2} \cdot \cdots \cdot x_n^{w_n} = \prod_{i=1}^{n} x_i^{w_i}; \quad w_i \in \mathbb{R} \tag{2}$$

The kind of network used in this paper is a feed-forward PUNN whose structure is shown in Fig. 3, where a PUNN model with a k:m:1 architecture for a bi-classification problem is sketched out; 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.

An identity function is utilised as a transfer function of each node in the hidden and output layers. Thus, the mathematical expression obtained by each of the nodes in the output layer with J classes is given by:

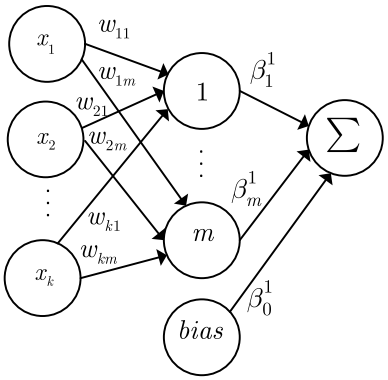

Fig. 3 Structure of a feed-forward PUNN model for a bi-classification problem. *Source*: Tallón-Ballesteros and Hervás-Martínez [66]

$$f(x_1, x_2, \ldots, x_k) = \sum_{j=1}^{m} \beta_j^l \left(\prod_{i=1}^{k} x_i^{w_{ij}} \right) + \beta_0^l$$
$$l = 1, 2, \ldots, J-1; \quad w_{ij} \in \Re \tag{3}$$

2.2 Training algorithms for neural networks

There are several algorithms to train neural networks. Among these are the methods based on:

(a) Local optimisation: inside this group, we must observe the gradient-descendent methods that use a fixed architecture and coefficient values which are calculated to try to minimise a cost function that is evaluated with the train data. The best-known example is the popular back-propagation algorithm [10, 55, 61, 70].
(b) Global optimisation: this type of optimisation treats to escape from local optima, exploring the search space with greater efficiency. Any random component is usually added to the search in order to eventually jump from a local optimum to another point. Among the global search methods, several meta-heuristic algorithms are found, which can be divided into two groups:

- Methods based on trajectories. The relevant approaches are the simulated annealing algorithm [12, 38], that has been employed, for instance, in [64] and [7], and the taboo search [29] as in [5].
- Methods based on populations. In this category the scatter search and all the techniques based on bio-inspired computation such as the algorithms based on evolutionary computing and/or Swarm intelligence should be underlined. The scatter search [28] has recently been applied in the work written by Larson and Newman [45]. The neural networks trained by methods based on evolutionary computation lead to the evolutionary neural networks that have been a key research area in the last decade of the twentieth century [62] and in the first of the twenty-first century [15], providing an improved platform to simultaneously optimise the network performance and the architecture (number of nodes in the hidden layer and the number of connections).

2.3 Methodology to train evolutionary product unit neural networks

Evolutionary computing was proposed as a good candidate in looking for the architecture space [50]. Since then, many evolutionary programming (EP) approaches have been developed to evolve neural networks, such as [26, 54, 72], whose main particularity is that the mutation and the recombination are the only two operators without any kind of crossover. We focus on feed-forward neural networks containing product units that are trained with an EPA to simultaneously learn about the architecture and the weights of the model. The baseline EPA can be found in some previous works [33, 49]. A deep explanation of the EPA including all the details can be found in Sect. 2.2 of [66], which we summarise next.

We are now going to briefly describe the applied EPA. As usual, it is based on a single population that is evolved throughout the full evolutionary cycle. This EPA is used to design the structure and learn the weights of PUNNs. The search begins with a random initial population, and, for each iteration, the population is modified using a population-update algorithm. The population is subjected to the operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving artificial networks [3, 72]. As a general remark, the EPA is substantially a hybridisation between two stochastic or probabilistic techniques such as an evolutionary algorithm and a simulated annealing procedure. Figure 4 depicts the pseudo-code of the EPA for a classification problem.

The basic foundations of the EPA are as follows:

- Error and fitness functions. We have considered a standard soft-max activation function [9], associated with the g network model, given by:

$$g_j(\mathbf{x}) = \frac{\exp f_j(\mathbf{x})}{\sum_{i=1}^{J} \exp f_i(\mathbf{x})} \quad j = 1, \ldots, J-1 \tag{4}$$

where J is the number of classes in the problem, $f_j(\mathbf{x})$ is the output of node j for pattern $\mathbf{x}$ and $g_j(\mathbf{x})$ is the probability that this pattern belongs to class j. Taking this into account, a function of cross-entropy error is used to evaluate a network

Program: Evolutionary Programming Algorithm
Data: Training set
Input parameters: gen, neu
Output: Best PUNN model
1: t ← 0
2: P(t) ← {ind$_1$, …, ind$_{10000}$} // Random initialisation of the population
3: f(P(t) {ind$_1$, …, ind$_{10000}$}) ← *fitness* (P(t) {ind$_1$, …, ind$_{10000}$}) // Calculate fitness
4: P(t) ← P(t) {ind$_1$, …, ind$_{10000}$} // Sort individuals by fitness: ind$_i$ > ind$_{i+1}$
5: P(t) ← P(t) {ind$_1$,… ind$_{1000}$} // Retain the 1000 best ones
6: **while** stop criterion not met **do** // main loop
7: P(t) {ind$_{901}$,…ind$_{1000}$} ← P(t) {ind$_1$, …, ind$_{100}$} // Best 10% replace the worst 10%
8: P(t+1) ← P(t) {ind$_1$, …, ind$_{900}$}
9: P(t+1) ← *pm* (P(t+1) {ind$_1$, …, ind$_{90}$}) // Parametric mutation (10% P(t+1))
10: P(t+1) ← *sm* (P(t+1) {ind$_{91}$, …, ind$_{900}$}) // Structural mutation (90% P(t+1))
11: f(P(t+1) {ind$_1$, .. ind$_{900}$}) ← *fitness* (P(t+1) {ind$_1$, …, ind$_{900}$}) // Evaluate
12: P(t+1) ← P(t+1) (ind$_1$, …, ind$_{900}$) ∪ P(t) {ind$_{901}$, …, ind$_{1000}$}
13: P(t+1) ← P(t+1){ind$_1$, …, ind$_{1000}$} // Sort individuals
14: last_generation ← t
15: t ← t+1
16: **end while**
17: **return** *best* (P(last_generation) {ind$_1$})

Fig. 4 EPA pseudo-code for a classification problem. Adapted from: Tallón-Ballesteros and Hervás-Martínez [66]

g with the instances of a problem, which is reflected in the following expression:

$$l(g) = -\frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{J} (y_i^j \ln(g_j(\mathbf{x}_i))) \tag{5}$$

and substituting g_j defined in (4),

$$l(g) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(- \sum_{j=1}^{J} y_i^j f_j(\mathbf{x}_i) + \ln \left(\sum_{j=1}^{J} \exp f_j(\mathbf{x}_i) \right) \right) \tag{6}$$

where y_i^j is the target value for class j with pattern $\mathbf{x}_i$ ($y_i^j = 1$ if $\mathbf{x}_i \in$ class j and $y_i^j = 0$ otherwise), $f_j(\mathbf{x}_i)$ is the output value of the neural network for the output neuron j with pattern $\mathbf{x}_i$, N the number of patterns and J the number of classes. On the one hand, we can observe that soft-max transformation produces positive estimates that sum to one, and therefore, the outputs can be interpreted as the conditional probability of class membership. On the other hand, the probability of one of the classes does not need to be estimated because of the normalisation condition. One activation function is usually set to zero; in this work, $f_J(\mathbf{x}_i) = 0$ and we reduce the number of parameters to estimate. In this way, the number of nodes in the output layer is equal to the number of classes minus one in the problem.

Since the EPA objective is to minimise the chosen error function, a fitness function is used in the form $A(g) = (1 + l(g))^{-1}$.

- *Initialisation of the population and stop condition* At the beginning of the EPA, 10**N* individuals are randomly generated by means of a pseudo-random number generator, *N* being the population size; in the current paper, it is equal to 1000 in all cases. Next, all individuals are evaluated, sorted by decreasing fitness and the best *N* ones will compose the initial population. The main loop of the EPA is repeated until the maximum number of generations (gen) is reached or until the best individual or the population mean fitness does not improve during *gen-without-improving* generations. Elitism [74] is a key ingredient which means that the most fit individuals are transferred to the next generation without alteration.
- Parametric and structural mutations. Parametric mutation is accomplished for each exponent w_{ji} and coefficient β_j^l of the model with Gaussian noise, where the variance depends on the temperature:

$$w_{ij}(t+1) = w_{ij}(t) + \xi_1(t) \quad i = 1, \ldots, k \quad j = 1, \ldots, m \tag{7}$$

$$\beta_j^l(t+1) = \beta_j^l(t) + \xi_2(t) \quad j = 0, \ldots, m \quad l = 1, \ldots, J-1 \tag{8}$$

Table 1 EPA general parameters

Parameter	Value
Population size (N)	1000
gen-without-improving	20
Interval for the exponents w_{ji}/coefficients β_j^l	$[-5, 5]$
Initial value of α_1	0.5
Initial value of α_2	1
Normalisation of the input data	[1, 2]
Number of nodes in node addition and node deletion operators	{1, 2}

where $\xi_k(t) \in N(0, \alpha_q T(g))$ $q = 1, 2$, represents a one-dimensional normally distributed random variable with mean 0 and variance $\alpha_q(t) \cdot T(g)$, t is the tth generation and T is the temperature of the g network model. Rechenberg's 1/5 success rule [58] has been applied as an evolutionary mechanism to update α_1 and α_2 parameters. On the other hand, a structural mutation implies a modification in the structure of the model and allows different regions in the search space to be explored while helping to maintain the diversity of the population. There are five different structural mutations: node addition, node deletion, connection addition, connection deletion and node fusion.

- Parameters. The main parameters of the EPA are the maximum number of generations (gen) and the maximum number of nodes in the hidden layer (neu). The minimum number of nodes is a unit lower than neu. The remaining parameters will be described further on. At the end of the EPA, it returns the best PUNN model with a number of nodes equal to the value of parameter neu in the hidden layer. Table 1 describes the values of the EPA general parameters.

Our attention is focused on the evolutionary PUNNs for classification problems. The experimental design distribution (EDD), introduced in [65], is our starting point. This methodology consists of distributing some parameters, either of the network topology or of the EPA, as the maximum number of nodes in the hidden layer (neu), the maximum number of generations (gen) and the output-variance value (α_2), over some computing nodes; each set of concrete values of previous parameters is called a configuration. To do this, an initial configuration, called the base configuration, is defined and it is modified with new values in one or two parameters in each of the computing nodes. Therefore, once the changes have been made, each of the processing nodes will run the EPA with a different configuration. Our interest now lies in distributing two parameters, neu and α_2; therefore, the gen parameter is fixed for the four considered configurations and depends only on the data set. Moreover, the first configuration is taken as the base one; therefore, we only mention four different configurations which are described in Table 2. Hence, the EPA is run for each data set with four configurations, that combine two different values for each of the parameters neu and α_2. To sum up, in the current paper, EDD runs the EPA (depicted in Fig. 4) with four different configurations.

Table 2 Description of the EDD configurations

Configuration	Num. of neurons (neu)	Max. num. of generations (gen)	α_2
1	neu	gen	1
2	neu + 1	gen	1
3	neu	gen	1.5
4	neu + 1	gen	1.5

It should be noted that EDD is used to obtain classification models without applying any feature selection method. Therefore, it takes the original data set and generates models without any pre-processing related to feature selection.

2.4 Feature selection

The goal of feature selection is to select a smaller attribute subset, composed of p attributes out of N attributes from a given set ($p \leq N$) [25]. Feature selection is essentially a task to remove irrelevant and/or redundant features [46]. Irrelevant features can be removed without affecting learning performance [37]. Redundant features are a type of irrelevant feature [73]. The distinction is that a redundant feature implies the co-presence of another feature; individually, each feature is relevant, but the removal of one of them will not affect learning performance.

According to the survey published in 1997 by Dash and Liu [16], an FS method generates different candidates from the feature space and assesses them based on an evaluation criterion to find the best feature subset. Depending on the evaluation criterion, FS can be divided into filter, hybrid and wrapper methods. Filters assess the relevance of features by taking a look at intrinsic data properties only—such as distance, consistency, and correlation [16, 17, 31]—and are thus independent of any learning algorithm. Hybrid models are the intermediate approach and were presented to handle large data sets [71] without needing a huge amount of time although they require more computational effort than filters. Finally, the wrapper approach requires a predetermined data mining algorithm and their performance is used to evaluate and determine which features are selected [41]. Wrappers often select higher accuracy features; however, they have a major drawback in their high computational cost and low generality.

The selection of features can be achieved in two general ways: one is to rank features according to some criterion and select the top k features, and the other is to select a minimum subset of features without learning performance deterioration. In other words, feature subset selection (FSS) algorithms can automatically determine the number of selected features, while feature ranking (FR) algorithms need to rely on some given threshold to select features [46]. In this context, a hybrid model entitled BIRS (Best Incremental Ranked Subset), which is a mixture between FR and FSS (FR-FSS), was proposed by Ruiz et al. [60]; it operates in 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. Any evaluator may be used in any BIRS stage. There is a great deal of criteria to group the feature selection algorithms based on filters; however, the described taxonomies are the most common in the literature.

We now comment on some concrete implementations of filters. Another interesting well-known option is the algorithm fast correlation-based filter (FCBF) [73] that uses symmetrical uncertainty (SU) to obtain relevant features and to remove redundancy in two steps. The first step generates a ranking based on the SU between each feature and the class. The 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, analysing whether a feature is discarded or not, depending on the feature–feature SU correlation. Information gain (IG) [14] is a popular concept used to evaluate the relevance of an attribute. Another measure is SOAP (selection of attributes by projection) whose principle is to place the best attributes with the smallest number of label changes [59].

3 Description of the proposal

The current paper presents a workbench to get simple models based on PUNNs via filter-based feature selection. The new methodology is a further step of EDD; more concretely, it is a mixture of some FS methods with EDD, and thus, we have called EDDFS methodology. First of all, some feature selectors based on filters are applied, in an independent way, to the training set of all data sets in order to obtain a list of attributes for each filter and data set. The list obtained for each data set and filter is considered for training and test phases. In this way, two reduced sets (reduced training and test sets) are generated, where only the most relevant features are included. It is important to point out that the feature selection is performed only with training data. Both reduced sets contain the same features. The reduced training set is taken as input to EDD and used to train the neural network model. After that, the model is evaluated by means of the reduced test set. EDDFS operates with six independent filters as 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. Figure 5 introduces the scheme of the proposed EDDFS methodology. EDDFS has two phases: (1) feature selection and (2) classification via EDD.

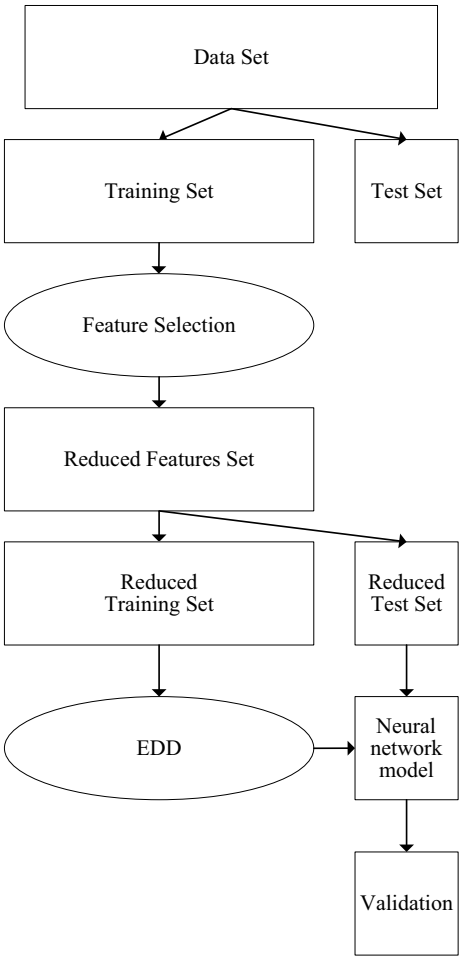

Fig. 5 EDDFS scheme

Gorunescu et al. [30] proposed a synergetic system with some similarities to our proposal in the sense that FS and evolutionary neural networks were integrated. More concretely, they proposed a tandem feature selection mechanism and evolutionary-driven neural network for a medical problem by means of a system based on both specific statistical tools and the sensitivity analysis provided by neural networks for reducing the dimension of the database.

The EDDFS properties are as follows: (a) PUNNs have been utilised, with a number of neurons in the input layer equal to the number of variables in the problem; 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 soft-max-type probabilistic approach has been used; (b) four experiments have been performed for each problem, where two different values have been used for α_2—associated with the residual of the updating expression of the output-layer weights—and the number of neurons in the hidden layer; (c) it employs similar terminology to the aforementioned EDD; (d) four different configurations (1#, 2#, 3# and 4#) 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 take specific values depending on the data set, and the last one depends on the configuration number (1#, …). Table 3 shows the main aspects of EDDFS configurations.

The proposed framework could operate with any FS method although we have tried to encompass many types of filter-based FS approaches with different options of the filter taxonomy and also bearing in mind whether the filter follows a FR and/or FSS strategy. Figure 6 provides an overview of the methods.

Additionally, a complete description of the different methods and metrics is now given. CFS and FCBF are two classic and very well recognised which have been accepted by the data mining community as reference procedures to cope with feature selection. More recently, BIRS has been established as a versatile approach, which does a fast search over a minimal part of the feature space and follows a forward search.

Table 3 Description of the EDDFS configurations

Configuration	Num. of neurons (neu#)	Max. num. of generations (gen#)	α_2
1#	neu2	gen2	1
2#	neu2 + 1	gen2	1
3#	neu2	gen2	1.5
4#	neu2 +1	gen2	1.5

CFS guides the search assessing the quality of a feature subset bearing in mind the hypothesis that good feature subsets include features highly correlated with the class label. CFS algorithm incorporates a heuristic to assess the worth or merit of a feature subset which takes into account the usefulness of individual features to predict the class label along with the level of inter-correlation among them. The hypothesis on which the heuristic is found is: *good feature subsets contain features highly correlated with the class, yet uncorrelated with each other*. It is expressed as follows:

$$\text{Merit}_S = \frac{k\overline{r_{\text{cf}}}}{\sqrt{k + k(k-1)r_{\text{ff}}}} \tag{9}$$

where Merit_S is the heuristic of a feature subset S containing k features, $\overline{r_{\text{cf}}}$ the average feature-class correlation and r_{ff} the average feature–feature inter-correlation. For discrete class problems, CFS first discretises numerical features and then uses symmetrical uncertainty to estimate the degree of association between discrete features.

CNS makes use of a consistency metric to estimate, for a feature subset at hand, the number of sources with full coincidence but their class labels; the inconsistency rate is then used to evaluate its quality. The algorithm creates a random subset S from the number of attributes in every round; whether the number of features of S is lower than the current best, the data with the attributes included in S are computed against the inconsistency criteria; and if its inconsistency rate is below a predefined one, S becomes the new current best.

The operation rationale of SOAP is to count the label changes of samples projected onto each attribute. It associates each feature with the classification label. This value is calculated by projecting data set elements onto the respective axis of the attribute (ordering the samples by this attribute),

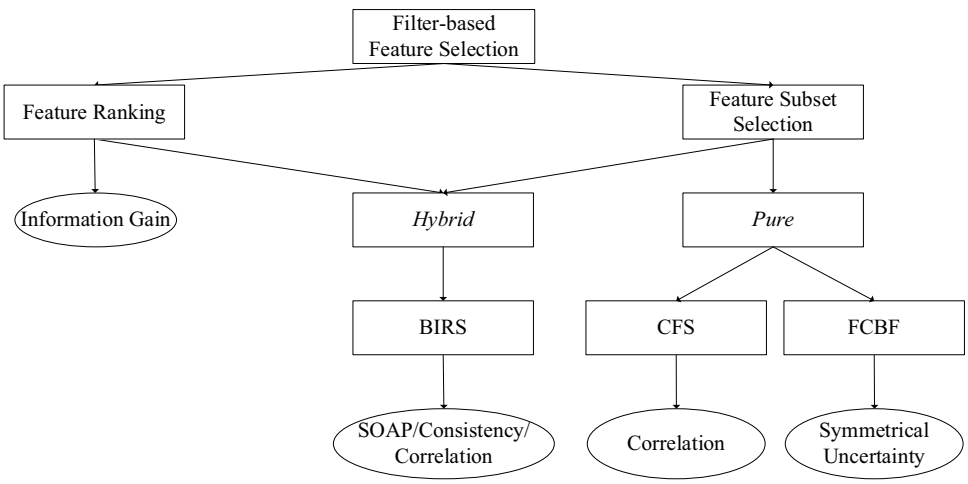

Fig. 6 Feature selection approaches within EDDFS

then crossing the axis from the beginning to the largest attribute value and counting the number of label changes (NLC) produced. BIRS copes with the incremental ranked usefulness in order to develop an approach to explicitly identify relevant attributes and to discard the redundant ones. The idea is to pick a feature from a ranked list one by one in the following way: firstly, the attributes are ranked according to some assessment measure (SOAP, CFS, CNS); and secondly, BIRS manages the list of features once, crossing the ranking from the beginning to the last ranked feature. Evaluation results are obtained using CFS or CNS with the first feature in the list, and it is marked as selected. Next, the result is again obtained with the first and second features; the second will be marked as selected depending on whether the evaluation obtained is statistically significantly better. The procedure is repeated until the last feature on the ranked list is reached. Finally, the algorithm returns the best subset found, and it can be stated that it will not contain irrelevant or redundant features.

Information measures or uncertainty measures determine the information gain from an attribute. The information gain from a feature is defined as difference between the prior uncertainty and expected posterior uncertainty using the feature. The *i*th attribute is preferred to the *j*th attribute whether the information gain from the *i*th attribute is greater than that from the *j*th attribute. The information gain of a given feature x with respect to the class label y is defined as the difference between the marginal entropy of y and the conditional entropy of y given x. The symmetrical uncertainty between the attributes x and y is defined as two times the quotient of the information gain of x provided by y divided by the sum of the individual entropies of x and y. FCBF is a feature subset approach which is based on information measures and, specifically, in symmetrical uncertainty; moreover, FCBF is a very fast feature selection method whose results according to the bibliographical review are very competitive or even better for some problems compared to CFS.

4 Experimentation

Experimentation is carried out with a good number of problems from the UCI repository [4] in order to evaluate the proposed methodology. We compare its results to the ones obtained with models without pre-processing the data sets by means of FS. After presenting the results with and without feature selection, there is a nonparametric analysis to determine whether the performance of the solutions improves quality-wise with respect to the correct classification rate (CCR), also known as accuracy, in the first case. Moreover, a computational cost comparison is performed to get an overview of the time reduction rate achieved with the new proposal. Other state-of-the-art classifiers have also been tested in order to ascertain whether there are significant differences between them applying the different FS methods. Lastly, an exhaustive statistical analysis is accomplished to determine the best classifier/s for each of the considered feature selectors.

4.1 Data sets and cross-validation

Table 4 summarises the binary and multi-class classification data sets employed. All of them are publicly available at the UCI repository [4]. The following fourteen have been used: *breast* cancer, breast tissue (*breast-t*), *cardiotocography*, statlog (*heart*), *hepatitis*, *labour* relations, lymphography, *Parkinson's*, *pima* Indians diabetes, steel *plates* faults, molecular biology (*promoter* gene sequences), *waveform* database generator (version 2), wine quality (*winequality-red*) and *yeast*.

Table 4 Summary of the binary and multi-class classification data sets used

Data set	Total patterns	Training patterns	Test patterns	Features	Inputs	Classes
Breast	286	215	71	9	15	2
Breast-t	106	81	25	9	9	6
Cardiotocography	2126	1594	532	23	31	3
Heart	270	202	68	13	13	2
Hepatitis	155	117	38	19	19	2
Labour	57	43	14	16	29	2
Lymphography	148	111	37	18	38	4
Parkinson's	195	146	49	23	22	2
Pima	768	576	192	8	8	2
Plates	1941	1457	484	27	27	7
Promoter	106	80	26	58	114	2
Waveform	5000	3750	1250	40	40	3
Winequality-red	1599	1196	403	11	11	6
Yeast	1484	1112	372	8	8	10

These data sets have in common that present error rates in test accuracy about 20% or above with reference classifiers such as C4.5 [57] or 1-NN [1, 13]. The size of the data sets ranges from over fifty to five thousand. The number of features depends on the problem and varies between eight and forty, while the number of classes is between two and ten. There are seven multi-class classification problems, and the remaining are binary ones. The column labelled Inputs represents the number of input nodes in the PUNN model. Since we are using neural networks, all nominal variables have been converted to binary ones; due to this, sometimes the number of inputs is greater than the number of features. Regarding the number of inputs, it ranges between eight and one hundred and fourteen. The missing values have also 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 hold-out, which consists of splitting the data into two sets: a training and a test set. The former is employed to train the neural network, and the latter is used to test the training process and to measure neural network generalisation capability. In our case, the size of the training set is $3n/4$ and that of the test set is approximately $n/4$, where n is the number of patterns in the problem; these percentages are similar to those used in [56]. More specifically, we have employed a stratified hold-out where the two sets are stratified [40] so that the class distribution of the samples in each set is approximately the same as in the original data set.

4.2 Feature selection methods, parameters and dimensionality reduction analysis

On the one hand, according to the filter type, there are three variations, namely FR, FSS and FR-FSS. On the other hand, several kinds of measures can be used for evaluating data properties of the ranking or the feature subset, such as those based on information, dependency or correlation, consistency, projection and any combination of some of the previous ones such as symmetrical uncertainty (based on information and correlation). Sometimes, FS is performed using fuzzy entropy measures as in the work of Luukka [48] in the context of a similarity classifier.

We have chosen six filters with the purpose of exploring the performance of some combinations of the aforementioned concepts. In a preliminary paper, we have used four out of these six methods as a previous step to the training of evolutionary PUNN by means of a two-stage evolutionary algorithm [67] which evolves two populations of individuals at the beginning of the evolutionary cycle. Now, once the feature selection is performed, the classification model is obtained using a canonical EPA exclusively with only one population as the main core of EDD. Specifically, we have considered six filters and we have carried out a detailed experimentation followed by an analysis of computational cost for the PUNN. We aim at giving a broader view of the feature selection methods and their performance with a good number of classifiers belonging to different approaches and to provide the most appropriate machine learning algorithms for each one of the filters.

For the methods based on FSS, as a subset evaluation we have used correlation-based feature selection (CFS) [31] and FCBF. As FR-FSS, we have applied BIRS using different evaluation measures such as those based on SOAP or consistency in the first phase as a ranking evaluator and for the subset evaluation in the second phase CFS or CNS (consistency based measure) [17]. The FR filter computes as ranking method the ranker that is based on an information measure.

Table 5 illustrates the general outline of the methods trialled in the experimentation containing six ones with and one without feature selection that belong to EDDFS (the current proposal) and EDD methodologies, respectively. All of them have been applied to each data set in an independent way. The feature selectors are implemented as filters. The third column defines an abbreviated name for each of them which is employed in the next sections. The details of

Table 5 General outline of the seven methods trialled in experimentation both with and without feature selection

Feature selector name	Methodology	Denomination	Ranking method	Subset evaluation	Measure kind	Filter type	References
–	EDD	FS0	None	None	–	–	[65]
spBI _CFS	EDDFS	FS1	spBI	CFS	Correlation	FR-FSS	[31, 59, 60]
BestFirst_CFS	EDDFS	FS2	BestFirst	CFS	Correlation	FSS	[31]
spBI _CNS	EDDFS	FS3	spBI	CNS	Consistency	FR-FSS	[17, 59, 60]
cnBI _CNS	EDDFS	FS4	cnBI	CNS	Consistency	FR-FSS	[17, 60]
InfoGain	EDDFS	FS5	Ranker	–	Information	FR	[14]
FCBF	EDDFS	FS6	Symmetrical uncertainty	FCBF	Information and correlation	FSS	[73]

sp and BI stand for SOAP and BIRS, respectively

each method are described from the fourth to the seventh columns. Last column provides some references of the filters, taking into account the ranking method, subset evaluation, measure kind and filter type. The framework EDD performs no feature selection, so we have indicated this situation with the symbol "–" and the letters FS0 for the feature selector name and denomination, respectively. In the rows containing FS1, FS2… and FS6, different filters have been applied. Generally speaking, these feature selection methods are parameter-less with the exception of FS5 for the one in which we have tried 0.01 and 0.05 values as threshold. We carried out a preliminary experimental design that showed a better performance of the classifiers with the 0.01 value. Thus, we have held the threshold of FS5 to 0.01 for all the experiments.

Table 6 depicts for each data set the number of inputs of the original train set (see column labelled FS0) and those which have been obtained with the different feature selectors (see columns labelled FS1–6) along with the reduction percentage in the inputs of each selector compared to the original data set. The last row shows the average of the number of inputs and the reduction percentage of the test bed for each experimented method on this paper. By rows, the maximum reduction percentage appears in bold.

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

$$\text{Reduction\_of\_Inputs}\ (\%) = \left(1 - \frac{\text{Inputs}\ (\text{FS}_i)}{\text{Inputs}\ (\text{FS}_0)}\right)100 \quad i = 1, \ldots, 6 \tag{10}$$

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. FS6 filter achieves a reduction percentage, on average, of 68.53% (from 27.43 to 5.93 features in average), which is the highest overall average value obtained. Individually, promoter data set has the highest reduction rate, ranging from 88.60 to 93.86%, depending on the method.

4.3 Choice of specific parameters depending on the data set

In relation to EDD methodology, the concrete values of neu and gen parameters for the base configuration depend on the data set and are shown in the first column of Table 7. The decision on the number of neurons in the hidden layer is a very difficult task in the scope of neural networks. The performance of the classifier might be better with other values, although determining the optimal values is a challenge. With respect to the number of generations, we have defined three kinds of values: small (100–120), medium (300) and large (500). Again, the optimal number is unknown; however, the algorithm has a stop criterion to avoid evolving up to the maximum number of generations if there is no improvement. We have performed a previous experimental design by means of a fivefold cross-validation with five repetitions on the training set for each data set in order to determine the values of both parameters. For this purpose, we have divided the data sets in two types: small and big problems. The former are those with less than one thousand of instances, and the latter have a number of instances greater than or equal to a thousand. *Promoter* data set has been submitted to the

Table 6 Number of inputs and reduction percentage for the test bed with and without feature selection

Data set	Inputs							Reduction (%)					
	FS0	FS1	FS2	FS3	FS4	FS5	FS6	FS1	FS2	FS3	FS4	FS5	FS6
Breast	15	4	4	2	2	5	3	73.33	73.33	**86.67**	**86.67**	66.67	80.00
Breast-t	9	6	6	5	6	8	4	33.33	33.33	44.44	33.33	11.11	**55.56**
Cardiotocography	31	9	7	10	21	20	8	70.97	**77.42**	67.74	32.26	35.48	74.19
Heart	13	7	7	8	9	9	6	46.15	46.15	38.46	30.77	30.77	**53.85**
Hepatitis	19	10	10	11	5	12	6	47.37	47.37	42.11	**73.68**	36.84	68.42
Labour	29	7	8	5	5	11	8	75.86	72.41	**82.76**	**82.76**	62.07	72.41
Lymphography	38	11	12	9	9	15	8	71.05	68.42	76.32	76.32	60.53	**78.95**
Parkinson's	22	5	6	7	6	21	4	77.27	72.73	68.18	72.73	4.55	**81.82**
Pima	8	3	4	4	5	6	4	**62.50**	50.00	50.00	37.50	25.00	50.00
Plates	27	16	10	19	21	27	6	40.74	62.96	29.63	22.22	0.00	**77.78**
Promoter	114	7	10	8	7	13	11	**93.86**	91.23	92.98	**93.86**	88.60	90.35
Waveform	40	14	14	15	15	19	5	65.00	65.00	62.50	62.50	52.50	**87.50**
Winequality-red	11	5	4	8	8	8	4	54.55	**63.64**	27.27	27.27	27.27	**63.64**
Yeast	8	5	7	7	7	7	6	**37.50**	12.50	12.50	12.50	12.50	25.00
Average	27.43	7.79	7.79	8.43	9.00	12.93	5.93	60.68	59.75	55.83	53.17	36.71	**68.53**

Table 7 Values of EDD/EDDFS parameters depending on the data set

Data set	EDD		EDDFS	
	Num. of neurons (neu)	Max. num. of generations (gen)	Num. of neurons (neu#)	Max. num. of generations (gen#)
Breast	9	500	7	500
Breast-t	5	300	5	150
Cardiotocography	6	300	5	150
Heart	6	500	4	25
Hepatitis	3	100	3	100
Labour	6	300	5	300
Lymphography	6	500	6	100
Parkinson's	6	500	3	500
Pima	3	120	3	120
Plates	6	500	5	500
Promoter	11	500	5	300
Waveform	3	500	3	500
Winequality-red	6	300	4	300
Yeast	11	500	11	500

experimental design associated with big problems due to the high number of inputs. Regarding small data sets, the values of the parameters are: neu= {2, 3, 4, 5, 6, 7, 8, 9, 10} and gen= {100/120, 300, 500}. For big problems, the following values have been taken: neu= {3, 4, 5, 6, 7, 8, 9, 10, 11} and gen= {300, 500}. The chosen values of the parameters are specified in Table 7. The value for neu parameters determines the topology of the neural network model.

In EDDFS, again there are two parameters, neu# and gen#, whose value is defined for each data set. The two last columns of Table 7 present their values along with the ones of EDD to have a general view of the differences. In EDD, the assignment of values is not trivial, but now in EDDFS this decision is more difficult because there are six FS methods and the values are common for all of them. Kwak and Choi [44] have also considered this idea. In our opinion, this approach is applicable because the order of magnitude of the selected features, as can be seen in the next section, is approximately the same and there are not very large differences in their number in the great majority of the FS methods for each data set. Moreover, choosing different values may lead the comparison to wrong conclusions because the better or worse performance could be originated by values of the parameters and not by the method itself. The problem in finding the best architectures in neural networks that employ input feature selection remains unsolved. The values of EDDFS parameters are limited by the EDD ones. This means that we have made a previous experimental design as in EDD, although the maximum values for neu# and gen# are those determined in EDD (neu and gen). For *Heart* data set, the algorithm converges very fast (no more than 30 generations for all filters) and the stop condition is soon fulfilled; thus, in EDDFS the gen# parameter value has been rounded to 25. It is crucial to note that the aforementioned values of the parameters concern the base configuration. The values of the remaining configurations are presented further on.

4.4 Topologies

We now proceed to describe the topologies whose results have been reported throughout the paper. Table 8 shows the topology of the different configurations for each data set and method. Configurations towards FS methods contain the symbol #, unlike those where no FS method is applied. Later, we only show the configuration number in order to ease legibility.

4.5 Nonparametric statistical analysis

We perform nonparametric statistical tests following the recommendations pointed out by Demšar [18]. Average ranks provide a fair comparison of the algorithms. The purpose of a nonparametric test is to determine the statistical significance of the differences in rank observed for each method with all data sets. There are two methods, the Friedman [24] and Iman–Davenport [32] tests. The former test is equivalent to the repeated-measures ANOVA [2], and it is based in χ_F^2 statistic; the null hypothesis expresses that all algorithms perform equally, so a rejection of it implies the existence of significant differences. The latter test is a derivation of the former based of F_F which is a better statistic, derived from χ_F^2. F_F is distributed according to the F distribution with $(k-1)$ and $(k-1)\,(N-1)$ degrees of freedom with k

Table 8 Topologies in fourteen data sets applying EDD and EDDFS

Data set	Method	Configuration (topology as *number of inputs*: *number of hidden neurons*: *number of outputs*)			
Breast	FS0	1 (15:9:1)	2 (15:10:1)	3 (15:9:1)	4 (15:10:1)
	FS1	1# (4:7:1)	2# (4:8:1)	3# (4:7:1)	4# (4:8:1)
	FS2	1# (4:7:1)	2# (4:8:1)	3# (4:7:1)	4# (4:8:1)
	FS3	1# (2:7:1)	2# (2:8:1)	3# (2:7:1)	4# (2:8:1)
	FS4	1# (2:7:1)	2# (2:8:1)	3# (2:7:1)	4# (2:8:1)
	FS5	1# (5:7:1)	2# (5:8:1)	3# (5:7:1)	4# (5:8:1)
	FS6	1# (3:7:1)	2# (3:8:1)	3# (3:7:1)	4# (3:8:1)
Breast-t	FS0	1 (9:5:5)	2 (9:10:5)	3 (9:5:5)	4 (9:10:5)
	FS1	1# (6:5:5)	2# (6:6:5)	3# (6:5:5)	4# (6:6:5)
	FS2	1# (6:5:5)	2# (6:6:5)	3# (6:5:5)	4# (6:6:5)
	FS3	1# (5:5:5)	2# (5:6:5)	3# (5:5:5)	4# (5:6:5)
	FS4	1# (6:5:5)	2# (6:6:5)	3# (6:5:5)	4# (6:6:5)
	FS5	1# (8:5:5)	2# (8:6:5)	3# (8:5:5)	4# (8:6:5)
	FS6	1# (4:5:5)	2# (4:6:5)	3# (4:5:5)	4# (4:6:5)
Cardiotocography	FS0	1 (31:6:2)	2 (31:7:2)	3 (31:6:2)	4 (31:7:2)
	FS1	1# (9:5:2)	2# (9:6:2)	3# (9:5:2)	4# (9:6:2)
	FS2	1# (7:5:2)	2# (7:6:2)	3# (7:5:2)	4# (7:6:2)
	FS3	1# (10:5:2)	2# (10:6:2)	3# (10:5:2)	4# (10:6:2)
	FS4	1# (21:5:2)	2# (21:6:2)	3# (21:5:2)	4# (21:6:2)
	FS5	1# (20:5:2)	2# (20:6:2)	3# (20:5:2)	4# (20:6:2)
	FS6	1# (8:5:2)	2# (8:6:2)	3# (8:5:2)	4# (8:6:2)
Heart	FS0	1 (13:6:1)	2 (13:7:1)	3 (13:6:1)	4 (13:7:1)
	FS1	1# (7:4:1)	2# (7:5:1)	3# (7:4:1)	4# (7:5:1)
	FS2	1# (7:4:1)	2# (7:5:1)	3# (7:4:1)	4# (7:5:1)
	FS3	1# (8:4:1)	2# (8:5:1)	3# (8:4:1)	4# (8:5:1)
	FS4	1# (9:4:1)	2# (9:5:1)	3# (9:4:1)	4# (9:5:1)
	FS5	1# (9:4:1)	2# (9:5:1)	3# (9:4:1)	4# (9:5:1)
	FS6	1# (6:4:1)	2# (6:5:1)	3# (6:4:1)	4# (6:5:1)
Hepatitis	FS0	1 (19:3:1)	2 (19:4:1)	3 (19:3:1)	4 (19:4:1)
	FS1	1# (10:3:1)	2# (10:4:1)	3# (10:3:1)	4# (10:4:1)
	FS2	1# (10:3:1)	2# (10:4:1)	3# (10:3:1)	4# (10:4:1)
	FS3	1# (11:3:1)	2# (11:4:1)	3# (11:3:1)	4# (11:4:1)
	FS4	1# (5:3:1)	2# (5:4:1)	3# (5:3:1)	4# (5:4:1)
	FS5	1# (12:3:1)	2# (12:4:1)	3# (12:3:1)	4# (12:4:1)
	FS3	1# (6:3:1)	2# (6:4:1)	3# (6:3:1)	4# (6:4:1)
Labour	FS0	1 (29:6:1)	2 (29:7:1)	3 (29:6:1)	4 (29:7:1)
	FS1	1# (7:5:1)	2# (7:6:1)	3# (7:5:1)	4# (7:6:1)
	FS2	1# (8:5:1)	2# (8:6:1)	3# (8:5:1)	4# (8:6:1)
	FS3	1# (5:5:1)	2# (5:6:1)	3# (5:5:1)	4# (5:6:1)
	FS4	1# (5:5:1)	2# (5:6:1)	3# (5:5:1)	4# (5:6:1)
	FS5	1# (11:5:1)	2# (11:6:1)	3# (11:5:1)	4# (11:6:1)
	FS6	1# (8:5:1)	2# (8:6:1)	3# (8:5:1)	4# (8:6:1)
Lymphography	FS0	1 (38:6:3)	2 (38:7:3)	3 (38:6:3)	4 (38:7:3)
	FS1	1# (11:6:3)	2# (11:7:3)	3# (11:6:3)	4# (11:7:3)
	FS2	1# (12:6:3)	2# (12:7:3)	3# (12:6:3)	4# (12:7:3)
	FS3	1# (9:6:3)	2# (9:7:3)	3# (9:6:3)	4# (9:7:3)
	FS4	1# (9:6:3)	2# (9:7:3)	3# (9:6:3)	4# (9:7:3)
	FS5	1# (15:6:3)	2# (15:7:3)	3# (15:6:3)	4# (15:7:3)
	FS3	1# (8:6:3)	2# (8:7:3)	3# (8:6:3)	4# (8:7:3)

Table 8 (continued)

Data set	Method	Configuration (topology as *number of inputs*: *number of hidden neurons*: *number of outputs*)			
Parkinson's	FS0	1 (22:6:1)	2 (22:7:1)	3 (22:6:1)	4 (22:7:1)
	FS1	1# (5:3:1)	2# (5:4:1)	3# (5:3:1)	4# (5:4:1)
	FS2	1# (6:3:1)	2# (6:4:1)	3# (6:3:1)	4# (6:4:1)
	FS3	1# (7:3:1)	2# (7:4:1)	3# (7:3:1)	4# (7:4:1)
	FS4	1# (6:3:1)	2# (6:4:1)	3# (6:3:1)	4# (6:4:1)
	FS5	1# (21:3:1)	2# (21:4:1)	3# (21:3:1)	4# (21:4:1)
	FS6	1# (4:3:1)	2# (4:4:1)	3# (4:3:1)	4# (4:4:1)
Pima	FS0	1 (8:3:1)	2 (8:4:1)	3 (8:3:1)	4 (8:4:1)
	FS1	1# (3:3:1)	2# (3:4:1)	3# (3:3:1)	4# (3:4:1)
	FS2	1# (4:3:1)	2# (4:4:1)	3# (4:3:1)	4# (4:4:1)
	FS3	1# (4:3:1)	2# (4:4:1)	3# (4:3:1)	4# (4:4:1)
	FS4	1# (5:3:1)	2# (5:4:1)	3# (5:3:1)	4# (5:4:1)
	FS5	1# (6:3:1)	2# (6:4:1)	3# (6:3:1)	4# (6:4:1)
	FS6	1# (4:3:1)	2# (4:4:1)	3# (4:3:1)	4# (4:4:1)
Plates	FS0	1 (27:6:6)	2 (27:7:6)	3 (27:6:6)	4 (27:7:6)
	FS1	1# (16:5:6)	2# (16:6:6)	3# (16:5:6)	4# (16:6:6)
	FS2	1# (10:5:6)	2# (10:6:6)	3# (10:5:6)	4# (10:6:6)
	FS3	1# (19:5:6)	2# (19:6:6)	3# (19:5:6)	4# (19:6:6)
	FS4	1# (21:5:6)	2# (21:6:6)	3# (21:5:6)	4# (21:6:6)
	FS5	1# (27:5:6)	2# (27:6:6)	3# (27:5:6)	4# (27:6:6)
	FS6	1# (6:5:6)	2# (6:6:6)	3# (6:5:6)	4# (6:6:6)
Promoter	FS0	1 (114:11:1)	2 (114:12:1)	3 (114:11:1)	4 (114:12:1)
	FS1	1# (7:5:1)	2# (7:6:1)	3# (7:5:1)	4# (7:6:1)
	FS2	1# (10:5:1)	2# (10:6:1)	3# (10:5:1)	4# (10:6:1)
	FS3	1# (8:5:1)	2# (8:6:1)	3# (8:5:1)	4# (8:6:1)
	FS4	1# (7:5:1)	2# (7:6:1)	3# (7:5:1)	4# (7:6:1)
	FS5	1# (13:5:1)	2# (13:6:1)	3# (13:5:1)	4# (13:6:1)
	FS6	1# (11:5:1)	2# (11:6:1)	3# (11:5:1)	4# (11:6:1)
Waveform	FS0	1 (40:3:2)	2 (40:4:2)	3 (40:3:2)	4 (40:4:2)
	FS1	1# (14:3:2)	2# (14:4:2)	3# (14:3:2)	4# (14:4:2)
	FS2	1# (14:3:2)	2# (14:4:2)	3# (14:3:2)	4# (14:4:2)
	FS3	1# (15:3:2)	2# (15:4:2)	3# (15:3:2)	4# (15:4:2)
	FS4	1# (15:3:2)	2# (15:4:2)	3# (15:3:2)	4# (15:4:2)
	FS5	1# (19:3:2)	2# (19:4:2)	3# (19:3:2)	4# (19:4:2)
	FS6	1# (5:3:2)	2# (5:4:2)	3# (5:3:2)	4# (5:4:2)
Winequality-red	FS0	1 (11:6:5)	2 (11:7:5)	3 (11:6:5)	4 (11:7:5)
	FS1	1# (5:4:5)	2# (5:5:5)	3# (5:4:5)	4# (5:5:5)
	FS2	1# (4:4:5)	2# (4:5:5)	3# (4:4:5)	4# (4:5:5)
	FS3	1# (8:4:5)	2# (8:5:5)	3# (8:4:5)	4# (8:5:5)
	FS4	1# (8:4:5)	2# (8:5:5)	3# (8:4:5)	4# (8:5:5)
	FS5	1# (8:4:5)	2# (8:5:5)	3# (8:4:5)	4# (8:5:5)
	FS6	1# (4:4:5)	2# (4:5:5)	3# (4:4:5)	4# (4:5:5)
Yeast	FS0	1 (8:11:9)	2 (8:12:9)	3 (8:11:9)	4 (8:12:9)
	FS1	1# (5:11:9)	2# (5:12:9)	3# (5:11:9)	4# (5:12:9)
	FS2	1# (7:11:9)	2# (7:12:9)	3# (7:11:9)	4# (7:12:9)
	FS3	1# (7:11:9)	2# (7:12:9)	3# (7:11:9)	4# (7:12:9)
	FS4	1# (7:11:9)	2# (7:12:9)	3# (7:11:9)	4# (7:12:9)
	FS5	1# (7:11:9)	2# (7:12:9)	3# (7:11:9)	4# (7:12:9)
	FS6	1# (6:11:9)	2# (6:12:9)	3# (6:11:9)	4# (6:12:9)

algorithms and N data sets. If the null hypothesis is rejected, we can proceed with a post hoc test. Bonferroni–Dunn [19] and Nemenyi [53] tests have both been carried out. The former compares some methods with a control method. The latter is used when all classifiers are compared to each other. The critical difference (CD) for each of them can be computed from critical values—which 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.

5 Results

This section depicts the CCR results obtained in the test set or in the test subset depending on that feature selection has been considered or not.

Firstly, we jointly present the results obtained with EDD and EDDFS. Then, a statistical analysis compares EDD versus EDDFS to determine whether there are significant differences between applying or not a feature selection with evolutionary PUNN. Later, the computational cost is reported in order to provide an efficiency measure.

Afterwards, a second experiment compares for each feature selector, the best mean values obtained with the current proposal to other classifiers using the same reduced data sets. Hence, in regard to EDD, we deal with the associated topology for each FS method that we report in this paper; the results have been extracted from next subsection for FS1–6 methods.

Lastly, a statistical comparison for each of the FS methods is presented in order to establish which classifier performs better with the considered test bed.

5.1 Results applying EDD and EDDFS

The results obtained with EDD [65] and EDDFS methodologies are shown together. There were eight configurations in the original proposal of EDD, denoted in the following way: 1, 2, … 8. This paper only deals with the first four configurations. In EDDFS, the four existing configurations are 1#… 4#.

Table 9 reports the mean and standard deviation (SD) of the test accuracies for each data set for a total of 30 runs. The rows containing FS0 refer to the results using the original data set performing no feature selection. The letters FS1, FS2, … FS6 mean that a filter has been applied. The best results without and with FS appear in bold for each data set. The last rows show the average values obtained for all data sets with each filter and configuration.

From the descriptive analysis of the data, it can be noted that the EDDFS methodology obtains the best results for almost all data sets. The SD reduction with EDDFS is often clear, and it expresses more homogeneous results compared to EDD. The accuracy average value increases from 70.83 to 75.27 in the best case; hence, FS is very valuable in order to improve the performance of the classifier based on evolutionary PUNN. All filters provide better average results that those obtained without FS.

5.1.1 Statistical analysis

In this section, we compare EDD and EDDFS methodologies by means of nonparametric statistical tests. To determine whether there are significant differences, we apply an Iman–Davenport test. It compares the average ranks of the algorithms, where a low rank value indicates a good algorithm performance and a high value a bad algorithm performance. The average ranks of all methods, without (FS0) and with FS (FS1–6), are shown in Table 10. The Iman–Davenport test results are presented in Table 11. According to them, since the F_F statistic is higher than the critical value 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 from at least the CD which can be calculated from the Bonferroni–Dunn test critical values, k and N. In our case, we make a comparison of methods that employ FS (FS1–6) versus the control method (FS0) that does not use FS. Table 12 displays 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 have been indicated for more clarity.

Next, the Bonferroni–Dunn test results are analysed and these enable us to ascertain that: there are significant differences between EDD applying each of the FS methods and EDD without FS. The statistical tests point out that evolutionary PUNN performance improves significantly preprocessing the data set with any of the FS methods employed in this paper. The best method is FS2 followed by FS1.

5.1.2 Analysis of computational cost

The comparison between EDD and EDDFS methodologies is completed by means of a computational cost analysis. Table 13 reports the time results concerning the average computational cost per evaluation measured in milliseconds (ms). Experiments have been run in a desktop computer with an Intel Core 2 Quad processor at 2.4 GHz and 2 GB RAM of physical memory. We now explain the table's contents. The first column specifies the data set name. Columns two to eight show the average elapsed time of an evaluation regarding to EDD (FS0) and EDDFS (FS1–6) methodologies. Columns nine to fourteen depict the reduction percentage of

Table 9 Test results obtained in fourteen data sets applying EDD and EDDFS

Data set	Method	Mean ± SD			
		Configuration			
		1/1#	2/2#	3/3#	4/4#
Breast	FS0	63.85 ± 3.81	63.00 ± 3.24	**64.27 ± 3.89**	63.43 ± 3.80
	FS1	70.84 ± 1.92	**70.93 ± 1.59**	70.18 ± 1.77	70.00 ± 1.92
	FS2	70.84 ± 1.92	**70.93 ± 1.59**	70.18 ± 1.77	70.00 ± 1.92
	FS3	69.20 ± 0.48	69.10 ± 0.35	69.06 ± 0.25	69.06 ± 0.25
	FS4	69.20 ± 0.48	69.10 ± 0.35	69.06 ± 0.25	69.06 ± 0.25
	FS5	68.49 ± 1.83	68.73 ± 2.96	68.26 ± 3.02	67.88 ± 2.67
	FS6	68.26 ± 1.03	68.17 ± 1.20	68.54 ± 0.77	68.64 ± 0.90
Breast-t	FS0	52.80 ± 6.42	54.00 ± 8.83	**54.53 ± 8.03**	53.46 ± 10.37
	FS1	56.00 ± 9.15	54.66 ± 7.52	53.33 ± 8.15	55.33 ± 7.20
	FS2	54.93 ± 7.27	57.07 ± 7.50	54.27 ± 8.51	55.73 ± 7.20
	FS3	54.93 ± 7.64	54.26 ± 7.90	59.86 ± 4.98	57.73 ± 8.44
	FS4	**61.06 ± 8.19**	57.46 ± 8.88	58.13 ± 8.64	58.80 ± 7.80
	FS5	55.73 ± 7.49	55.60 ± 7.07	57.20 ± 9.28	54.26 ± 9.43
	FS6	55.20 ± 6.51	56.00 ± 5.66	57.87 ± 6.79	57.33 ± 6.91
Cardiotocography	FS0	80.93 ± 2.76	79.94 ± 3.70	**81.25 ± 3.17**	81.04 ± 3.79
	FS1	**85.45 ± 1.61**	84.16 ± 2.27	84.34 ± 2.31	83.93 ± 2.85
	FS2	81.57 ± 1.86	81.79 ± 2.34	81.30 ± 1.91	81.35 ± 1.80
	FS3	83.17 ± 2.42	82.78 ± 3.63	82.30 ± 3.51	81.04 ± 3.35
	FS4	77.27 ± 1.41	76.44 ± 2.14	76.89 ± 1.73	76.52 ± 2.40
	FS5	81.37 ± 2.76	80.35 ± 3.08	81.14 ± 3.20	81.71 ± 2.62
	FS6	81.09 ± 2.36	81.81 ± 1.87	81.22 ± 2.37	81.62 ± 2.47
Heart	FS0	75.93 ± 2.40	75.83 ± 3.27	**76.23 ± 2.48**	76.03 ± 3.50
	FS1	76.23 ± 1.86	76.47 ± 2.12	75.93 ± 2.33	77.50 ± 2.01
	FS2	76.23 ± 1.86	76.47 ± 2.12	75.93 ± 2.33	77.50 ± 2.01
	FS3	76.08 ± 2.50	75.98 ± 2.30	76.47 ± 2.01	75.59 ± 2.37
	FS4	77.40 ± 2.10	76.76 ± 2.09	77.89 ± 2.49	**77.99 ± 1.79**
	FS5	77.40 ± 2.10	76.76 ± 2.09	77.89 ± 2.49	**77.99 ± 1.79**
	FS6	76.08 ± 3.27	76.32 ± 2.68	75.58 ± 3.10	76.32 ± 3.04
Hepatitis	FS0	84.47 ± 4.49	**85.52 ± 4.67**	84.47 ± 4.55	84.29 ± 5.33
	FS1	88.77 ± 2.49	88.77 ± 2.93	88.95 ± 2.80	**89.91 ± 2.59**
	FS2	88.77 ± 2.49	88.77 ± 2.93	88.95 ± 2.80	**89.91 ± 2.59**
	FS3	89.04 ± 2.40	89.30 ± 2.49	89.56 ± 2.13	89.47 ± 2.85
	FS4	86.67 ± 1.53	86.67 ± 1.82	86.40 ± 1.40	86.32 ± 1.45
	FS5	87.01 ± 3.01	87.63 ± 2.20	87.89 ± 2.72	88.07 ± 2.74
	FS6	88.42 ± 2.81	88.68 ± 2.68	88.24 ± 2.82	88.24 ± 2.26
Labour	FS0	84.04 ± 11.04	83.57 ± 10.29	**84.28 ± 10.67**	82.85 ± 13.98
	FS1	92.85 ± 4.96	92.61 ± 5.13	90.95 ± 4.93	91.42 ± 4.74
	FS2	95.71 ± 4.44	93.57 ± 5.42	**96.19 ± 4.08**	94.29 ± 4.75
	FS3	91.19 ± 3.07	90.95 ± 3.21	89.76 ± 4.05	89.76 ± 3.60
	FS4	91.19 ± 3.07	90.95 ± 3.21	89.76 ± 4.05	89.76 ± 3.60
	FS5	92.61 ± 5.13	95.00 ± 5.97	92.61 ± 5.77	95.00 ± 5.01
	FS6	91.42 ± 3.93	91.42 ± 4.35	91.66 ± 4.22	92.38 ± 4.56
Lymphography	FS0	75.49 ± 6.98	76.12 ± 6.95	76.48 ± 6.70	**76.85 ± 7.56**
	FS1	78.28 ± 5.91	77.47 ± 4.83	78.01 ± 6.05	75.04 ± 6.49
	FS2	78.46 ± 6.08	81.08 ± 4.76	**81.17 ± 5.56**	79.54 ± 4.69
	FS3	77.02 ± 4.90	76.93 ± 5.15	77.20 ± 4.79	75.31 ± 5.48
	FS4	76.93 ± 4.74	79.18 ± 4.60	78.46 ± 4.83	79.18 ± 4.97
	FS5	78.64 ± 5.46	78.79 ± 5.38	78.19 ± 4.48	79.27 ± 5.37

Table 9 (continued)

Data set	Method	Mean ± SD			
		Configuration			
		1/1#	2/2#	3/3#	4/4#
	FS6	80.00 ± 4.79	79.91 ± 4.64	79.64 ± 4.95	77.48 ± 3.71
Parkinson's	FS0	**79.66 ± 5.01**	78.16 ± 4.77	78.98 ± 4.05	79.32 ± 4.76
	FS1	79.66 ± 2.37	79.32 ± 2.32	79.93 ± 2.15	79.05 ± 2.83
	FS2	80.88 ± 1.96	**82.24 ± 2.21**	81.49 ± 1.92	81.29 ± 2.73
	FS3	80.27 ± 4.23	81.84 ± 4.20	80.61 ± 3.07	80.14 ± 3.90
	FS4	78.03 ± 1.16	80.07 ± 2.82	78.78 ± 1.98	79.66 ± 3.24
	FS5	80.20 ± 4.45	79.65 ± 3.14	80.34 ± 3.65	78.50 ± 3.24
	FS6	81.42 ± 1.54	82.17 ± 2.33	81.83 ± 1.88	80.95 ± 3.14
Pima	FS0	77.33 ± 2.36	**78.61 ± 1.88**	76.96 ± 1.67	77.69 ± 1.79
	FS1	79.54 ± 0.90	79.49 ± 0.79	79.60 ± 0.87	79.89 ± 0.92
	FS2	80.53 ± 0.87	80.65 ± 1.18	80.65 ± 1.14	**80.83 ± 0.96**
	FS3	75.48 ± 1.42	75.19 ± 1.26	75.00 ± 1.17	75.31 ± 1.51
	FS4	78.42 ± 1.09	78.76 ± 1.13	78.73 ± 1.06	78.71 ± 1.47
	FS5	78.94 ± 1.34	79.09 ± 1.40	79.02 ± 1.50	79.35 ± 1.48
	FS6	80.73 ± 0.87	80.80 ± 1.14	80.69 ± 1.00	80.78 ± 0.82
Plates	FS0	52.61 ± 5.09	51.18 ± 5.35	52.01 ± 3.37	**52.82 ± 4.03**
	FS1	55.25 ± 3.66	52.36 ± 2.94	50.83 ± 5.95	54.14 ± 3.89
	FS2	49.94 ± 3.72	49.35 ± 5.14	52.00 ± 6.12	52.74 ± 5.91
	FS3	54.51 ± 4.92	55.07 ± 2.81	54.24 ± 2.32	**57.30 ± 4.88**
	FS4	53.08 ± 4.96	54.91 ± 4.44	54.82 ± 5.62	55.06 ± 4.12
	FS5	52.61 ± 5.09	51.18 ± 5.35	52.01 ± 3.37	52.82 ± 4.03
	FS6	52.21 ± 4.40	53.25 ± 4.22	52.96 ± 4.40	53.28 ± 3.72
Promoter	FS0	59.74 ± 9.30	58.21 ± 9.67	**60.51 ± 10.00**	55.51 ± 10.03
	FS1	84.48 ± 3.97	**84.62 ± 3.78**	83.20 ± 3.97	82.94 ± 4.12
	FS2	77.69 ± 5.84	76.15 ± 6.50	77.05 ± 5.93	77.95 ± 6.69
	FS3	67.43 ± 5.77	67.05 ± 5.11	66.15 ± 5.56	67.94 ± 5.74
	FS4	75.89 ± 4.39	75.51 ± 4.45	76.41 ± 3.74	76.53 ± 4.55
	FS5	77.06 ± 6.61	77.43 ± 5.41	75.25 ± 6.36	76.79 ± 6.26
	FS6	80.64 ± 6.50	79.69 ± 5.71	78.46 ± 7.46	81.69 ± 5.59
Waveform	FS0	81.43 ± 2.10	82.78 ± 0.64	82.05 ± 1.64	**84.32 ± 1.73**
	FS1	84.97 ± 1.13	86.54 ± 0.48	84.92 ± 0.98	86.30 ± 0.95
	FS2	84.97 ± 1.13	86.54 ± 0.48	84.92 ± 0.98	86.30 ± 0.95
	FS3	85.39 ± 1.41	85.78 ± 0.74	85.20 ± 1.14	86.37 ± 0.84
	FS4	84.87 ± 0.93	**86.75 ± 0.57**	85.55 ± 1.21	85.66 ± 0.80
	FS5	85.58 ± 1.15	86.35 ± 0.95	85.22 ± 1.30	85.99 ± 1.20
	FS6	79.95 ± 0.61	80.06 ± 0.54	80.00 ± 0.49	80.01 ± 0.53
Winequality-red	FS0	61.03 ± 1.30	60.80 ± 1.25	**61.16 ± 1.20**	60.98 ± 1.42
	FS1	61.67 ± 1.10	61.49 ± 0.99	61.21 ± 1.11	61.15 ± 1.30
	FS2	61.67 ± 1.10	61.49 ± 0.99	61.21 ± 1.11	61.15 ± 1.30
	FS3	61.70 ± 1.06	61.54 ± 1.10	**61.75 ± 1.01**	61.66 ± 1.05
	FS4	61.70 ± 1.06	61.54 ± 1.10	**61.75 ± 1.01**	61.66 ± 1.05
	FS5	61.70 ± 1.06	61.54 ± 1.10	**61.75 ± 1.01**	61.66 ± 1.05
	FS6	61.39 ± 0.88	61.51 ± 0.95	61.29 ± 1.04	61.67 ± 0.84
Yeast	FS0	59.18 ± 1.17	**59.62 ± 1.27**	58.50 ± 1.74	59.18 ± 1.83
	FS1	59.82 ± 1.22	59.53 ± 1.36	58.95 ± 1.14	59.72 ± 1.46
	FS2	60.10 ± 1.41	**60.36 ± 1.16**	59.93 ± 1.85	60.20 ± 1.46
	FS3	60.10 ± 1.41	**60.36 ± 1.16**	59.93 ± 1.85	60.20 ± 1.46
	FS4	60.10 ± 1.41	**60.36 ± 1.16**	59.93 ± 1.85	60.20 ± 1.46

Table 9 (continued)

Data set	Method	Mean ± SD			
		Configuration			
		1/1#	2/2#	3/3#	4/4#
	FS5	60.10 ± 1.41	**60.36 ± 1.16**	59.93 ± 1.85	60.20 ± 1.46
	FS6	58.23 ± 1.27	58.19 ± 1.41	57.91 ± 1.02	58.05 ± 1.22
Average	FS0	70.61	70.53	**70.83**	70.56
	FS1	**75.27**	74.89	74.31	74.74
	FS2	74.45	74.94	74.71	74.91
	FS3	73.25	73.30	73.36	73.35
	FS4	73.70	73.89	73.75	73.94
	FS5	74.10	74.18	74.05	74.25
	FS6	73.93	74.14	74.02	74.17

Table 10 Average ranks of EDD and EDDFS

Method	Average rank
FS0	6.61
FS1	3.29
FS2	2.82
FS3	3.75
FS4	3.71
FS5	3.75
FS6	4.07

Table 11 Statistics and critical value for Iman–Davenport test of EDD and EDDFS

Statistics		Critical value for $\alpha = 0.05$
χ_F^2	F_F	$F(6, 78)$
26.60	6.07	2.22

Table 12 Critical difference values and ranking differences of EDD and EDDFS by means of a Bonferroni–Dunn test (FS0 is the control method)

FS0 versus	Ranking difference (control method − compared method)	Significant for compared method
FS1	3.32	*
FS2	3.79	*
FS3	2.86	*
FS4	2.90	*
FS5	2.86	*
FS6	2.54	*
	$CD_{(\alpha=0.05)} = 2.15$; $CD_{(\alpha=0.10)} = 1.95$	

*Statistically significant difference with $\alpha = 0.05$

each method of EDDFS versus EDD (FS0). The time values shown are the average of the four configurations for each method (FS0, FS1, FS2, FS3, FS4, FS5 and FS6). The last row contains the average of the values in the column; the best and the second best values appear in bold and italics, respectively.

The time reduction percentage is given by the following expression:

$$\text{Time\_Reduction}\ (\%) = \left(1 - \frac{\text{FS}_i}{\text{FS}_0}\right)100 \quad i = 1, \ldots, 6 \quad (11)$$

where i is the FS method index.

Looking at the previous table, we conclude that the reduction percentage, on average, is higher than 40% in all cases and greater than a 60% in the best approach. There are also extreme cases for single data sets. For instance, in Promoter and Parkinson's data sets, the higher reduction rates are 96.13% (from 16.02 to 0.62 ms) and 87.58% (from 5.52 to 0.69 ms). However, in data sets with times above 100 ms such as Plates, Waveform and Yeast the maximum reduction is, respectively, 66.03% (from 147.58 to 50.13 ms) 80.34% (from 142.96 to 28.11 ms) and 23.11% (from 144.88 to 111.40 ms). In absolute terms, the time per evaluation has been reduced to 17.65 ms, in the fastest option, from 46.01 ms, considering the averages for the test bed.

The empirical times give notice that the proposed methodology, EDDFS, is much more efficient than the previous methodology, EDD. The average time with FS ranges from 17.65 to 31.27 ms versus 46.01 ms without FS.

5.2 Comparison of EDDFS with a variety of classifiers

Now, a comparison is performed between EDDFS and other state-of-the-art machine learning classifiers with and without FS. These algorithms are C4.5, k-nearest neighbours (k-NN),—concretely 1-NN—, SVM [69], PART [23] and the MLP model [6] with a learning back-propagation method (BP). Since all of them are implemented in the Waikato environment for knowledge

Table 13 Average computational cost and reduction percentage for the test bed applying EDD and EDDFS

Data set	Average computational cost (ms)							Reduction (%)					
	FS0	FS1	FS2	FS3	FS4	FS5	FS6	FS1	FS2	FS3	FS4	FS5	FS6
Breast	5.89	1.81	1.81	1.18	1.18	1.99	1.72	69.21	69.21	80.03	80.03	66.16	70.81
Breast-t	3.00	2.09	2.01	1.79	2.07	2.53	1.90	30.26	32.85	40.48	31.11	15.55	36.73
Cardiotocography	90.89	24.60	22.22	30.43	53.15	45.97	23.01	72.93	75.55	66.53	41.52	49.42	74.69
Heart	4.10	0.96	0.96	1.02	1.15	1.15	1.00	76.57	76.57	75.07	71.82	71.82	75.62
Hepatitis	0.95	0.63	0.63	0.37	0.62	0.88	0.57	33.66	33.66	60.37	34.83	7.00	39.34
Labour	0.97	0.36	0.42	0.30	0.30	0.50	0.42	63.46	56.44	68.77	68.77	48.76	57.21
Lymphography	8.60	3.07	3.41	2.27	2.53	3.98	1.94	64.29	60.34	73.57	70.56	53.78	77.44
Parkinson's	5.52	0.82	0.90	0.96	0.97	2.47	0.69	85.21	83.66	82.64	82.45	55.31	87.58
Pima	2.99	2.53	2.81	2.48	2.91	2.95	2.84	15.30	6.04	17.21	2.55	1.35	5.14
Plates	147.58	96.39	66.98	109.44	99.31	147.58	50.13	34.68	54.62	25.85	32.71	0.00	66.03
Promoter	16.02	0.70	0.81	0.62	0.62	0.86	0.75	95.62	94.94	96.13	96.13	94.62	95.30
Waveform	142.96	64.28	64.28	71.85	74.15	87.50	28.11	55.03	55.03	49.74	48.13	38.79	80.34
Winequality-red	69.82	27.49	27.49	36.55	36.55	27.49	21.56	60.62	60.62	47.65	47.65	60.62	69.12
Yeast	144.88	111.40	111.40	133.20	133.20	111.40	112.54	23.11	23.11	8.06	8.06	23.11	22.32
Average	46.01	24.08	*21.87*	28.03	29.19	31.27	**17.65**	55.71	55.90	*56.58*	51.17	41.88	**61.26**

analysis (WEKA) workbench [8] the same cross-validation, hence the same instances, in each of the partitions, that in the first experiment have been used. BP parameter values were the following: learning rate $\eta = 0.3$ and momentum $\alpha = 0.2$. The remaining algorithms have been run with WEKA default values that are those recommended by the own algorithm authors. The number of runs for MLP was 30.

We have reported in Table 14 the results both without (FS0 method) and with FS (FS1… and FS6) for each data set and algorithm. In each row, the best result appears in bold and the second best one in italics.

From a purely descriptive analysis of the results, we can assert the following.

Taking into account the data sets without any FS application, SVM classifier achieves the best result in seven out of fourteen data sets; the EDD method, C4.5 and MLP obtain twice the highest accuracies. On average, SVM has the best accuracy (74.12%), MLP the second best one (72.48%) and the remaining algorithms range from 67.68% (PART) to 71.40% (EDD).

Focusing on FS, we can conclude that the EDD method obtains the best result for seven out of fourteen data sets; C4.5 and 1-NN algorithms yield the highest performance for two data sets, and the remaining classifiers once. Furthermore, EDD reports the highest mean accuracy (75.62%) followed by SVM (73.04%).

Both statements confirm the good behaviour of the evolutionary product units. The important achievement of the FS combined with EDD lets the proposed methodology, EDDFS, to substantially improve the performance. However, we need to consider another type of information, not only the quantitative one.

5.2.1 Statistical analysis applying FS

This section involves a comparison in the performance between the aforementioned five algorithms, considering the results applying FS. As in the first experiment, non-parametric statistical tests have been employed. To determine whether there are significant differences, we apply an Iman–Davenport test. This method compares the average ranks of the algorithms, where a low rank value indicates a good algorithm performance and a high value a bad algorithm performance. The average ranks of all methods applying FS are depicted in Table 15. The Iman–Davenport test results for each feature selector are presented in Table 16. According to the results, since the F_F statistic is higher than the critical value in all cases the null hypothesis is rejected. So, we proceed with a post hoc Nemenyi test to compare all classifiers with each other for detecting significant differences. The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the CD, whose value can be computed from Nemenyi test critical values, k and N.

Tables 17, 18, 19, 20, 21 and 22 show the Nemenyi test results, each one focusing on FS1, FS2, FS3, FS4, FS5 and FS6, respectively, where the ranking difference between each different pair and the detected significant difference level have been indicated for more clarity.

Table 14 Test results obtained in fourteen data sets for several classifiers both with and without FS

Data set	Method	Classifier					
		C4.5	1-NN	SVM	PART	MLP	EDD
Breast	FS0	**70.42**	64.79	64.79	*69.01*	60.80	64.27
	FS1	69.01	70.42	66.20	**71.83**	69.01	*70.93*
	FS2	69.01	70.42	66.20	**71.83**	69.01	*70.93*
	FS3	69.01	**70.42**	64.79	69.01	69.01	*69.20*
	FS4	69.01	**70.42**	64.79	69.01	69.01	*69.20*
	FS5	64.79	**70.42**	64.79	**70.42**	*69.30*	68.73
	FS6	69.01	**70.42**	64.79	69.01	*69.53*	68.64
Breast-t	FS0	52.00	*60.00*	52.00	44.00	**63.20**	54.53
	FS1	56.00	52.00	*60.00*	44.00	**65.33**	56.00
	FS2	**68.00**	56.00	60.00	56.00	*65.47*	57.07
	FS3	52.00	48.00	*60.00*	52.00	**66.40**	59.86
	FS4	52.00	52.00	*64.00*	52.00	**67.20**	61.06
	FS5	56.00	56.00	*60.00*	48.00	**64.67**	57.20
	FS6	48.00	48.00	56.00	48.00	**65.60**	*57.87*
Cardiotocography	FS0	*82.71*	76.32	**83.65**	82.52	80.75	81.25
	FS1	77.07	81.77	81.20	*82.52*	81.94	**85.45**
	FS2	78.38	80.45	*81.39*	81.20	80.86	**81.79**
	FS3	*84.21*	79.70	79.14	**85.15**	80.91	83.17
	FS4	*75.19*	63.91	*75.19*	75.00	68.29	**77.27**
	FS5	*83.08*	76.88	**84.21**	81.58	81.87	81.71
	FS6	77.82	*81.20*	*81.20*	77.26	80.13	**81.81**
Heart	FS0	70.59	73.53	**76.47**	73.53	74.85	*76.23*
	FS1	73.53	73.53	76.47	**77.94**	72.50	*77.50*
	FS2	73.53	73.53	76.47	**77.94**	72.50	*77.50*
	FS3	73.53	*75.00*	**76.47**	*75.00*	73.09	**76.47**
	FS4	72.06	75.00	*76.47*	75.00	74.85	**77.99**
	FS5	72.06	75.00	*76.47*	75.00	74.85	**77.99**
	FS6	73.53	70.59	**77.94**	75.00	74.90	*76.32*
Hepatitis	FS0	84.21	*86.84*	**89.47**	81.58	84.73	85.52
	FS1	84.21	*89.47*	86.84	84.21	87.28	**89.91**
	FS2	84.21	*89.47*	86.84	84.21	87.28	**89.91**
	FS3	89.47	**92.11**	89.47	86.84	86.84	*89.56*
	FS4	**89.47**	84.21	**89.47**	84.21	84.21	*86.67*
	FS5	**89.47**	86.84	86.84	84.21	87.89	*88.07*
	FS6	**89.47**	84.21	**89.47**	86.84	87.72	*88.68*
Labour	FS0	**85.71**	71.43	78.57	**85.71**	69.52	*84.28*
	FS1	*85.71*	71.43	78.57	*85.71*	64.29	**92.85**
	FS2	*85.71*	64.29	71.43	*85.71*	57.62	**96.19**
	FS3	*85.71*	64.28	78.57	78.57	78.57	**91.19**
	FS4	*85.71*	64.28	78.57	78.57	78.57	**91.19**
	FS5	*85.71*	78.57	71.43	*85.71*	71.43	**95.00**
	FS6	*85.71*	78.57	71.43	78.57	71.43	**92.38**
Lymphography	FS0	75.68	83.78	**91.89**	75.68	*86.58*	76.85
	FS1	**88.29**	78.38	*83.78*	70.27	73.24	78.28
	FS2	*81.08*	*81.08*	*81.08*	64.86	80.45	**81.17**
	FS3	75.68	70.27	**78.38**	64.86	71.89	*77.20*
	FS4	75.68	70.27	*78.38*	64.86	71.89	**79.18**
	FS5	81.08	81.08	**86.49**	64.86	*83.42*	79.27
	FS6	**81.08**	75.68	**81.08**	70.27	74.50	*80.00*

Table 14 (continued)

Data set	Method	Classifier					
		C4.5	1-NN	SVM	PART	MLP	EDD
Parkinson's	FS0	71.43	77.55	75.51	75.51	*77.62*	**79.66**
	FS1	75.51	79.59	75.51	77.55	**81.56**	*79.93*
	FS2	73.47	81.63	75.51	79.59	**83.13**	*82.24*
	FS3	75.51	*81.63*	75.51	75.51	75.71	**81.84**
	FS4	79.59	79.59	75.51	**81.63**	75.65	*80.07*
	FS5	71.43	**85.71**	75.51	75.51	79.05	*80.34*
	FS6	81.63	73.47	79.59	77.55	**84.83**	*82.17*
Pima	FS0	74.48	73.96	*78.13*	74.48	75.94	**78.61**
	FS1	76.04	74.48	77.60	76.04	*78.18*	**79.89**
	FS2	76.04	67.71	*79.17*	76.04	78.73	**80.83**
	FS3	69.79	71.88	73.96	72.92	*74.25*	**75.48**
	FS4	74.48	67.19	*78.65*	74.48	76.89	**78.76**
	FS5	74.48	69.27	*77.08*	74.48	76.13	**79.35**
	FS6	76.04	67.71	*79.17*	76.04	79.01	**80.80**
Plates	FS0	39.05	49.17	**57.02**	46.69	*53.50*	52.82
	FS1	40.50	51.24	51.03	46.90	**56.71**	*55.25*
	FS2	*54.75*	47.31	51.65	51.65	**57.33**	52.74
	FS3	39.67	51.24	**57.44**	46.90	55.14	*57.30*
	FS4	38.22	50.62	*55.17*	44.63	**55.24**	55.06
	FS5	39.05	49.17	**57.02**	46.69	*53.50*	52.82
	FS6	44.63	43.18	45.04	49.79	*52.85*	**53.28**
Promoter	FS0	69.23	65.38	**88.46**	53.85	*86.03*	60.51
	FS1	73.08	57.69	**84.62**	80.77	*84.49*	**84.62**
	FS2	73.08	69.23	73.08	**80.77**	76.28	*77.95*
	FS3	*76.92*	61.54	*76.92*	**80.77**	65.38	67.94
	FS4	*80.77*	57.69	**84.62**	76.92	75.64	76.53
	FS5	73.08	69.23	76.92	**80.77**	72.95	*77.43*
	FS6	73.08	76.92	73.08	*80.77*	78.21	**81.69**
Waveform	FS0	74.80	68.96	**86.24**	76.88	*84.85*	84.32
	FS1	74.40	75.36	**86.88**	77.04	83.21	*86.54*
	FS2	74.40	75.36	**86.88**	77.04	83.21	*86.54*
	FS3	74.88	74.88	**87.12**	79.92	83.41	*86.37*
	FS4	74.40	76.64	**87.12**	79.68	86.27	*86.75*
	FS5	76.32	73.76	**87.12**	76.64	82.96	*86.35*
	FS6	74.72	69.12	*78.80*	74.00	77.57	**80.06**
Winequality-red	FS0	53.85	49.88	*59.55*	51.36	56.35	**61.16**
	FS1	50.87	48.88	*59.80*	52.11	59.36	**61.67**
	FS2	50.87	48.88	*59.80*	52.11	59.36	**61.67**
	FS3	50.12	49.63	*58.81*	52.85	57.04	**61.75**
	FS4	50.12	49.63	*58.81*	52.85	57.04	**61.75**
	FS5	50.87	48.88	*59.80*	52.11	59.36	**61.75**
	FS6	51.36	50.37	59.31	49.13	*59.64*	**61.67**
Yeast	FS0	54.84	48.39	55.91	56.72	**59.94**	*59.62*
	FS1	53.49	48.92	54.03	54.84	**60.20**	*59.82*
	FS2	54.03	49.46	54.84	54.30	*60.20*	**60.36**
	FS3	54.03	49.46	54.84	54.30	*60.20*	**60.36**
	FS4	54.03	49.46	54.84	54.30	*60.20*	**60.36**
	FS5	54.03	49.46	54.84	54.30	*60.20*	**60.36**
	FS6	52.69	48.12	51.61	52.96	**58.96**	*58.23*

Table 14 (continued)

Data set	Method	Classifier					
		C4.5	1-NN	SVM	PART	MLP	EDD
Average	FS0	68.50	67.86	**74.12**	67.68	*72.48*	71.40
	FS1	69.84	68.08	*73.04*	70.12	72.66	**75.62**
	FS2	71.18	68.20	71.74	70.95	*72.25*	**75.49**
	FS3	69.32	67.15	*72.24*	69.62	71.28	**74.12**
	FS4	69.34	65.07	*72.97*	68.80	71.50	**74.42**
	FS5	69.39	69.31	*72.75*	69.31	72.68	**74.74**
	FS6	69.91	66.97	70.61	68.94	*72.49*	**74.54**

Table 15 Average ranks of FS methods with different classifiers

Method	Classifier					
	C4.5	1-NN	SVM	PART	MLP	EDDFS
FS1	4.57	4.39	3.32	3.75	3.07	1.89
FS2	4.29	4.75	3.29	3.61	3.36	1.71
FS3	4.25	4.43	2.89	4.04	3.46	1.93
FS4	4.04	4.86	2.61	4.18	3.46	1.86
FS5	4.32	4.46	2.68	4.11	3.14	2.29
FS6	3.82	4.79	3.29	4.32	2.93	1.86

Table 16 Statistics and critical value for Iman–Davenport test of FS methods with different classifiers

Method	Statistics		Critical value for $\alpha = 0.05$ $F(5, 65)$
	χ_F^2	F_F	
FS1	19.22	4.92	2.36
FS2	21.79	5.87	
FS3	18.20	4.57	
FS4	24.35	6.93	
FS5	17.00	4.17	
FS6	22.01	5.96	

An analysis based upon the Nemenyi test results allows us to state the following.

First, with respect to the FS1 method, there are significant differences between EDDFS and all the methods except for SVM and MLP. Excluding EDDFS, there are no differences between the remaining algorithms.

Second, in relation to the FS2 feature selector, there are significant differences between EDDFS and C4.5, 1-NN (at a significance level of 0.05) and PART (at a significance level of 0.10) classifiers. The rank differences of SVM and MLP versus EDDFS are 1.58 and 1.65, respectively. This justifies that EDDFS with FS2 shows an excellent performance.

Third, with regards to the FS3 method, there are significant differences between EDDFS and C4.5, 1-NN and PART. EDDFS and SVM rankings are very close and differ in 0.96 in favour of the first method; SVM does not present significant differences versus the remaining algorithms.

Fourth, related to FS4 method, there are significant differences between EDDFS and C4.5, 1-NN and PART. EDDFS and SVM rankings differ now in 0.75 for the first

Table 17 Pairwise comparisons of the classifiers with FS1 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		0.18	1.25	0.82	1.50	2.68*
1-NN			1.04	0.64	1.32	2.50*
SVM				− 0.43	0.25	1.43
PART					0.68	1.86°
MLP						1.18
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

Table 18 Pairwise comparisons of the classifiers with FS2 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		− 0.46	1.00	0.68	0.93	2.58*
1-NN			1.46	1.14	1.39	3.04*
SVM				− 0.32	− 0.07	1.58
PART					0.25	1.90°
MLP						1.65
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

Table 19 Pairwise comparisons of the classifiers with FS3 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		− 0.18	1.36	0.21	0.79	2.32*
1-NN			1.54	0.39	0.97	2.50*
SVM				− 1.15	− 0.57	0.96
PART					0.58	2.11*
MLP						1.53
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

Table 20 Pairwise comparisons of the classifiers with FS4 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		− 0.82	1.43	− 0.14	0.58	2.18*
1-NN			2.25*	0.68	1.40	3.00*
SVM				− 1.57	− 0.85	0.75
PART					0.72	2.32*
MLP						1.60
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

Table 21 Pairwise comparisons of the classifiers with FS5 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		− 0.14	1.64	0.21	1.18	2.03*
1-NN			1.78	0.35	1.32	2.17*
SVM				− 1.43	− 0.46	0.39
PART					0.97	1.82
MLP						0.85
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

classifier. In the current case, SVM wins to 1-NN with significant differences.

Fifth, towards FS5, EDDFS obtains significant differences only versus C4.5 and 1-NN. The rank difference between PART and EDDFS is very close to CD (at a significance level of 0.10) in favour of EDDFS.

Lastly and taking into concern the FS6 method, EDDFS wins in a significant way to 1-NN, PART and C4.5 (at 0.10

Table 22 Pairwise comparisons of the classifiers with FS6 by means of a Nemenyi test

	C4.5	1-NN	SVM	PART	MLP	EDDFS
C4.5		− 0.97	0.53	− 0.50	0.89	1.96°
1-NN			1.50	0.47	1.86°	2.93*
SVM				− 1.03	0.36	1.43
PART					1.39	2.46*
MLP						1.07
$CD_{(\alpha=0.05)} = 2.02$; $CD_{(\alpha=0.10)} = 1.83$						

Each filled cell contains the ranking difference between the methods in the row and the column. Also, it is specified if the latter method outperforms the former one at a significance level of 0.05 (*) or 0.10 (°)

level). MLP gets significant differences with 0.10 confidence level versus 1-NN.

6 Conclusions

A methodology to achieve simple classification models combining an initial filter-based feature selection with a later step to build a classifier based on product unit neural networks trained with an evolutionary programming algorithm in classification problems using EDD workbench has been introduced. A blend of EDD framework and FS, called EDDFS, has certainly been proposed. The pre-processing stage is performed by means of FS methods implemented as filters. A review of the state-of-the-art methods has been done, and six filters from different feature selection approaches, such as FR-FSS, FSS and FR, have been tried out. According to the experimental results, the models obtained with the proposal are more accurate as well as simpler, in relation to the number of inputs and/or nodes in the hidden layer. Besides, the current contribution is much more efficient, ranging from 40 to 60%, than the previous one.

The empirical study to compare EDDFS and EDD methodologies, both of them based on Evolutionary PUNN, has been performed on fourteen classification problems (seven binary and seven multiple class data sets) from the very well-known machine learning repository from the University of California at Irvine, which present test error rates measured in accuracy about a 20% or above with C4.5 or 1-NN classifiers. A great deal of configurations, over four hundred, has been trialled. The average test accuracy has been improved in more than four points (more or less from 71 to 75%). The statistical analysis reveals that differences are significant in favour for any considered filter.

We have also undergone other state-of-the-art classifiers with and without FS using the fourteen data sets in order to get an overall outlook. We investigated further the competitiveness of each FS method by comparing their performance applying some reference-supervised learning algorithms to the reduced data sets, in order to determine for each feature selector which is the best machine learning algorithm of the ones considered.

Nonparametric statistical tests were conducted, and the most important conclusions reached are as follows. On the one hand, all the filter-based FS methods helped to improve significantly the accuracy of the neural network models with product units. As stated by the average ranks the best filters are, in this order, FS2 (BestFirst_CFS), FS1 (spBI_CFS) and FS3 (spBI_CNS). This means that correlation-based FS methods are the most appropriate for product units, followed by those based on consistency. At a lower level, BestFirst search and SOAP measure in conjunction with Best Incremental Ranked Subset are powerful for feature selection with evolutionary neural networks containing kinds of multiplicative neurons in the hidden layer such as product units.

On the other hand, the comparison with other classifiers reported in this paper sheds light on that EDDFS has the best average rank versus the remaining algorithms for all FS methods. Comparing the different filters, the results of the tests allow us to state the following. EDDFS achieves significant differences versus three classifiers, such as C4.5, 1-NN and PART, with FS1 (spBI_CFS), FS2 (BestFirst_CFS), FS3 (spBI_CNS), FS4 (cnBI_CNS) and FS6 (FCBF) filter. With FS5 (InfoGain), the difference is significant when compared with the two classifiers. For each filter, EDDFS has a better ranking than SVM and MLP applying the Friedman test, although the difference is not significant. FS2 (BestFirst_CFS) provides greater rank differences than the remaining filters to EDDFS versus SVM and MLP. EDDFS obtains the best results, on average, with FS1 (75.62), and the results closer to this are obtained by SVM (73.04) and MLP (72.66) classifiers.

According to the above results, our new learning methodology of neural networks based on product units, EDDFS, is seen to significantly improve accuracy in all cases with respect to the previous approach, EDD. Lastly, we mention some efficiency issues. The current proposal, EDDFS, is much more efficient than EDD. The time reduction percentage ranges approximately from a 41 to 61%, on average, for the test bed.

Acknowledgements This work has been partially subsidised by TIN2011-28956-C02-02 and TIN2014-55894-C2-R projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) and FEDER funds.

References

1. Aha D, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6:37–66
2. Anderson TW (2003) An introduction to multivariate statistical analysis. Wiley, New York
3. Angeline PJ, Saunders GM, Pollack JB (1994) An evolutionary algorithm that construct recurrent neural networks. IEEE Trans Neural Netw 5(1):54–65
4. Bache K, Lichman M (2013) UCI machine learning repository. School of Information and Computer Science, University of California, Irvine
5. Battiti R, Tecchiolli G (1995) Training neural nets with the reactive tabu search. IEEE Trans Neural Netw 6(5):1185–1200
6. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, New York
7. Boese KD, Kahng AB (1993) Simulated annealing of neural networks: the cooling strategy reconsidered. In: Proceedings of the IEEE international symposium on circuits and systems (ISCAS 1993), vol 4. IEEE, Chicago, Illinois, USA, pp 2572–2575
8. Bouckaert RR, Frank E, Hall MA, Holmes G, Pfahringer B, Reutemann P, Witten IH (2010) Weka—experiences with a java open-source project. J Mach Learn Res 11(1):2533–2541
9. Bridle JS (1990) Probabilistic Interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Fogelman Soulie F, Herault J (eds) Neurocomputing: algorithms, architectures and applications. Springer, Berlin, pp 227–236
10. Bryson AE, Yu-Chi H (1969) Applied optimal control: Optimization, estimation, and control. Blaisdell Publishing Company, Waltham
11. Caruana R, Freitag D (1994) Greedy attribute selection. In: Proceedings of the eleventh international conference on machine learning (ICML 1994). Morgan Kaufmann, New Brunswick, NJ, USA, pp 28–36
12. Cerný V (1985) Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Optim Theory Appl 45(1):41–51
13. Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27
14. Cover T, Thomas J (1991) Elements of information theory. Wiley, New York
15. Curran D, O'Riordan C (2002) Applying evolutionary computation to designing neural networks: a study of the state of the art. Technical report NUIG-IT-111002, National University of Ireland, Galway, Department of Information Technology
16. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(3):131–156
17. Dash M, Liu H (2003) Consistency-based search in feature selection. Artif Intell 151(1):155–176
18. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
19. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64
20. Durbin R, Rumelhart DE (1989) Product units: a computationally powerful and biologically plausible extension to backpropagation networks. Neural Comput 1(1):133–142
21. Embrechts MJ (2001) Computational intelligence for data mining. In: Proceedings of IEEE international conference on systems, man, and cybernetics (SMC 2001), vol 3. IEEE, Los Alamitos, pp 1484–1484
22. Ferreira CBR, Borges DL (2003) Analysis of mammogram classification using a wavelet transform decomposition. Pattern Recognit Lett 24(7):973–982
23. Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: Proceedings of the fifteenth international conference on machine learning (ICML 1998). Morgan Kaufmann, Madison, Wisconsin, USA, pp 144–151
24. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
25. Fu KS, Min PJ, Li TJ (1970) Feature selection in pattern recognition. IEEE Trans Syst Sci Cybern 6(1):33–39
26. García-Pedrajas N, Hervás-Martínez C, Muñoz-Pérez J (2002) Multiobjetive cooperative coevolution of artificial neural networks. Neural Netw 15(10):1255–1274
27. Gislason PO, Benediktsson JA, Sveinsson JR (2006) Random Forests for land cover classification. Pattern Recognit Lett 27(4):294–300
28. Glover F (1977) Heuristics for integer programming using surrogate constraints. Decis Sci 8(1):156–166
29. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549
30. Gorunescu F, Belciug S, Gorunescu M, Badea R (2012) Intelligent decision-making for liver fibrosis stadialization based on tandem feature selection and evolutionary-driven neural network. Expert Syst Appl 39(17):12824–12832
31. Hall MA, Smith LA (1997) Feature subset selection: a correlation based filter approach. In: Proceedings of the 1997 international conference on neural information processing and intelligent information systems. Springer, New Zealand, pp 855–858
32. Iman RL, Davenport JM (1980) Approximations of the critical region of the Friedman statistic. Commun Stat Theory Methods 9(6):571–595
33. Hervás-Martínez C, Martínez-Estudillo FJ, Gutiérrez PA (2006) Classification by means of evolutionary product-unit neural networks. In: Proceedings of the international joint conference on neural networks (IJCNN 2006). IEEE, Vancouver, BC, Canada, pp 2834–2842
34. Jaeger H (2002) Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD report 159, German National Research Center for Information Technology
35. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
36. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. Computer 29(3):31–44
37. John GH, Kohavi R, Pfleger K (1994) Irrelevant feature and the subset selection problem. In: Proceedings of the eleventh international conference on machine learning (ICML 1994). Morgan Kaufmann, New Brunswick, NJ, USA, pp 121–129
38. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
39. Krasnopolsky VM, Fox-Rabinovitz MS (2006) Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 19:122–134
40. Kohavi R (1995) 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, Montréal, Québec, Canada, pp 1137–1145
41. Kohavi R, John G (1997) Wrappers for feature subset selection. Artif Intell 97(1):273–324

42. Koller D, Sahami M (1996) Toward optimal feature selection. In: Proceedings of the thirteenth international conference on machine learning (ICML 1996). Morgan Kaufmann, Bari, Italy, pp 284–292
43. Kuncheva LI, del Rio Vilas VJ, Rodríguez JJ (2007) Diagnosing scrapie in sheep: a classification experiment. Comput Biol Med 37(8):1194–1202
44. Kwak N, Choi CH (2002) Input feature selection for classification problems. IEEE Trans Neural Netw 13(1):143–159
45. Larson J, Newman F (2011) An implementation of scatter search to train neural networks for brain lesion recognition. Involve J Math 4(3):203–211
46. Liu H, Motoda H (2008) Computational methods of feature selection. Chapman & Hall/CRC, Boca Raton
47. Liu H, Setiono R (1998) Some issues on scalable feature selection. Expert Syst Appl 15(3–4):333–339
48. Luukka P (2011) Feature selection using fuzzy entropy measures with similarity classifier. Expert Syst Appl 38(4):4600–4607
49. Martínez-Estudillo FJ, Hervás-Martínez C, Gutiérrez-Peña PA, Martínez-Estudillo AC, Ventura-Soto S (2006) Evolutionary product-unit neural networks for classification. In: Proceedings of the seventh international conference on intelligent data engineering and automated learning (IDEAL 2006). Springer, Burgos, Spain, pp 1320–1328
50. Miller GF, Todd PM, Hegde SU (1989) Designing neural networks using genetic algorithms. In: Proceedings of the 3rd international conference on genetic algorithms (ICGA 1989). Morgan Kaufmann, George Mason University, Fairfax, Virginia, USA, pp 379–384
51. Milne L (1995) Feature selection using neural networks with contribution measures. In: Proceedings of the eighth Australian joint conference on artificial intelligence (AI 95). Canberra, Australia, pp 215–221
52. Murty MN, Devi VS (2011) Pattern recognition: An algorithmic approach. Springer, New York
53. Nemenyi PB (1963) Distribution-free multiple comparisons. PhD, Princeton University
54. Ohkura K, Yasuda T, Kawamatsu Y, Matsumura Y, Ueda K (2007) MBEANN: mutation-based evolving artificial neural networks. In: Advances in artificial life, proceedings of the 9th European conference (ECAL 2007). Springer, Lisbon, Portugal, pp 936–945
55. Parker DB (1985) Learning logic. Technical report TR-47, MIT Center for Research in Computational Economics and Management Science, Cambridge, MA
56. Prechelt L (1994) Proben1—a set of neural network benchmark problems and benchmarking rules. Technical report 21/94, Fakultat für Informatik, Univ. Karlsruhe, Karlsruhe, Germany
57. Quinlan J (1993) C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco
58. Rechenberg I (1989) Evolution strategy: Nature's way of optimization. In: Bergmann HW (ed) Optimization: Methods and applications, possibilities and limitations. Springer, Bonn, pp 106–126
59. Ruiz R, Riquelme JC, Aguilar-Ruiz JS (2003) Fast feature ranking algorithm. In: Proceedings of the seventh international conference on knowledge-based intelligent information and engineering systems (KES 2003). Springer, Oxford, UK, pp 325–331
60. Ruiz R, Riquelme JC, Aguilar-Ruiz JS (2006) Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognit 39(12):2383–2392
61. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, the PDP Research Group (eds) Parallel distributed processing: explorations in the microstructure of cognition (volume 1: foundations). MIT Press, Cambridge, MA, pp 318–362
62. Schaffer JD, Whitley D, Eshelman LJ (1992) Combinations of genetic algorithms and neural networks: a survey of the state of the art. In: Proceedings of the international workshop on combinations of genetic algorithms and neural networks (COGANN 1992). IEEE Society Press, Los Alamitos, CA, pp 1–37
63. Sethi IK, Jain AK (2014) Artificial neural networks and statistical pattern recognition: Old and new connections. Machine intelligence and pattern recognition series, vol 11. Elsevier, Amsterdam
64. Sexton R, Dorsey R, Johnson J (1999) Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. Eur J Oper Res 114(3):589–601
65. Tallón-Ballesteros AJ, Gutiérrez-Peña PA, Hervás-Martínez C (2007) 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, pp 266–273
66. Tallón-Ballesteros AJ, Hervás-Martínez C (2011) A two-stage algorithm in evolutionary product unit neural networks for classification. Expert Syst Appl 38(1):743–754
67. Tallón-Ballesteros AJ, Hervás-Martínez C, Riquelme JC, Ruiz R (2013) Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems. Neurocomputing 114:107–117
68. Towell GG, Shavlik JW (1994) Knowledge-based artificial neural networks. Artif Intell 70(1–2):119–165
69. Vapnik VN (1995) The nature of statistical learning theory. Springer, Heidelberg
70. Werbos PJ (1974) Beyond regression: new tools for prediction and analysis in the behavioural sciences. PhD thesis, Harvard University, Boston
71. Xing EP, Jordan MI, Karp RM (2001) Feature selection for high-dimensional genomic microarray data. In: Proceedings of the international conference on machine learning (ICML 2001). Morgan Kaufmann, San Francisco, CA, pp 601–608
72. Yao X, Liu Y (1997) A new evolutionary system for evolving artificial neural networks. IEEE Trans Neural Netw 8(3):694–713
73. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224
74. Zhen S, Jianlin C, Di T, Zhou YCT (2004) Comparison of steady state and elitist selection genetic algorithms. In: Proceedings of international conference on intelligent mechatronics and automation (ICMA 2004). IEEE, pp 495–499