

# Metaheuristics for feature selection: application to sepsis outcome prediction

Susana M. Vieira\*, Luis F. Mendonça<sup>†\*</sup>, Gonçalo J. Farinha\* and João M.C. Sousa\*

\*Technical University of Lisbon, Instituto Superior Técnico

Dept. of Mechanical Engineering, CIS/IDMEC LAETA, Lisbon, Portugal

Email:susana.vieira@ist.utl.pt

<sup>†</sup>Escola Superior Náutica Infante D. Henrique

Department of Marine Engineering, Lisbon, Portugal

**Abstract**—This paper proposes the application of a new binary particle swarm optimization (BPSO) method to feature selection problems. Two enhanced versions of binary particle swarm optimization, designed to cope with premature convergence of the BPSO algorithm, are proposed. These methods control the swarm variability using the velocity and the similarity between best swarm solutions. The proposed PSO methods use neural networks, fuzzy models and support vector machines in a wrapper approach, and are tested in a benchmark database. It was shown that the proposed BPSO approaches require an inferior simulation time, less selected features and increase accuracy. The best BPSO is then compared with genetic algorithms (GA) and applied to a real medical application, a sepsis patient database. The objective is to predict the outcome (survived or deceased) of the sepsis patients. It was shown that the proposed BPSO approaches are similar in terms of model accuracy when compared to GA, while requiring an inferior simulation time and less selected features.

## I. INTRODUCTION

Knowledge is only valuable when it can be used efficiently and effectively. Therefore, extensive research has been made in the urge to find new computational theories and tools, that can aid to the extraction of useful information (knowledge) from rapidly growing databases. The field of science concerned with automated knowledge discovery is called knowledge discovery in databases (KDD) [1].

The KDD process comprises a series of steps to extract knowledge from data. The first step is *selection* and it consists in acquiring the most useful target data from the available databases. The target database has to be adequately chosen so that it contains sufficient information regarding the system we want to describe. The next two steps (feature construction and feature selection), are part of the feature extraction (FE) process and are used with the purpose of extracting the most relevant features (or features) of the target data. *Feature construction* (also called data preprocessing), [2], comprehends all the methods that involve some degree of modification to the original feature, e.g. data *standardization*, *normalization* and *noise filtering*. The objective of this crucial preprocessing step is to make the underlying information in data easier to identify. In opposition, *feature selection* (FS) does not induce a transformation to the features, it simply searches for the optimal feature subset discarding the features with lowest informative potential.

There is a great number of available FS techniques, but there are three aspects that roughly differentiate them [2]:

- feature subset generation (or search strategy);
- evaluation criterion definition (e.g. relevance index or predictive performance);
- evaluation criterion estimation (or assessment method).

The first refers to the applied search strategy to evaluate the solutions in the space of possible feature combinations. The last two correspond to the evaluation criterion, i.e. the method and measures used to assess the quality of each feature subset. Based on the subset evaluation procedure we may divide FS algorithms in two classes, *wrapper methods* and *filter methods*.

The main advantage of wrapper method over the filter methods is that, in wrappers the predictive performance of the final selected subset is correlated with the chosen relevance measure. Once the objective in this work is to improve the modeling quality using the information underlying in large databases, it has been decided to use wrapper methods. Nevertheless, wrapper methods have the associated problem of having to train a classifier for each tested feature subset. This means testing all the possible combinations of features will be virtually impossible. To solve this problem several search heuristics have been proposed, e.g. genetic algorithms (GA), particle swarm (PSO), ant colony optimization (ACO). These methods are able to find fairly good solutions without searching the entire workspace.

The feature selection techniques in study are two modified versions of the binary PSO algorithm (BPSO) and genetic algorithms. They are used as a wrapper method, that is solely a feature selection method in which every candidate solution is evaluated using a learning machine, [2]. In this work, we have chosen to use the following modeling methodologies: neural networks (NN), support vector machines (SVM) and fuzzy modeling (FM). The most important characteristic of these methods are their universal function approximation properties, [3]. The objective of this paper is the application of feature selection to a publicly available septic shock patient database in order to obtain more accurate models of the disease.

We will start by introducing the main concepts of modeling in Section II. Then, the implementation of wrapper methodologies is addressed in Section III. In Section IV, the applied metaheuristics are presented, namely the proposed BPSO-

based methods. Furthermore, in Section V the studied wrapper methods are validated over multiple databases. Finally, the conclusions of this work are discussed in Section VI.

## II. MODELING

It is always difficult to choose a specific modeling technique. The sepsis problem is a very complex medical state, thus the criteria for choosing a modeling technique was the capability of effectively representing highly non-linear problems. Artificial Neural Networks (NN), Fuzzy Modeling (FM) and Support Vector Machines (SVM), were chosen due to their universal function approximation properties, [4].

### A. Decision Regions and Discriminant Functions

A classifier can be defined as any function  $f$ , that partitions the feature space into decision regions  $D_1, D_2, \dots, D_l$ , such that:

$$f : \mathbb{R}^{N_f} \rightarrow \Omega \quad (1)$$

Ideally, all feature vectors belonging to the same class are assigned to the same decision region. These regions are usually non-overlapping hypervolumes, with the boundaries between them being called *decision boundaries*. More generally, the classification decisions can be stated using a set of  $\mathbf{d} = (d^{(1)}, d^{(2)}, \dots, d^{(l)})$ , discriminant functions [5]:

$$d^{(j)} : \mathbb{R}^{N_f} \rightarrow \mathbb{R}, \quad j = 1, 2, \dots, l, \quad (2)$$

where each discriminant function  $d^{(j)}$ , yields a score for the respective class  $v_j$ . The sample  $\mathbf{x}$  is assigned to the class  $v_i$ , which has the largest discriminant value:

$$d^{(i)}(\mathbf{x}) > d^{(j)}(\mathbf{x}), \quad j = 1, 2, \dots, l, \quad j \neq i \quad (3)$$

The case of a two-class (or binary) classification problem can be treated in a different manner. Instead of two discriminant functions  $d^{(1)}(\mathbf{x})$  and  $d^{(2)}(\mathbf{x})$  applied separately, we can define a *dichotomizer* as being:

$$d(\mathbf{x}) = d^{(1)}(\mathbf{x}) - d^{(2)}(\mathbf{x}). \quad (4)$$

The class is assigned to the sample based on the *sign* of the value of  $d(\mathbf{x})$ . When  $d^{(1)}(\mathbf{x})$  is larger than  $d^{(2)}(\mathbf{x})$ , and the sample  $\mathbf{x}$  is assigned to the class 1, and vice-versa.

The machine learning techniques that will be presented use the values of the output functions as discriminant functions in order to perform classification. We will start by introducing neural networks.

### B. Artificial Neural Networks

Artificial Neural Networks [4] are adaptive systems design to emulate, in a very simplified way, the structure of the human brain. One of the main reasons why NN were developed was to capture some of the advantages that biological neural networks have over computational systems. They have the ability to learn complex relations by generalizing from a limited amount of data. This happens because of its simple parallel distributed structure, which will be following described.

The basic components of an NN are its artificial neurons (or nodes) and its weights. Their mathematical functionality is characterized by a combination and modification of the inputs. In the first step, each of the neuron inputs  $x_{ij}$  is multiplied by a weight  $w_{ij}$  (weighting process). Further, all the resulting terms are summed. This process is summarized in the following equation:

$$z = \sum_{i=1}^n w_{ij} x_{ij} = \mathbf{w}_j^T \mathbf{x}_j, \quad (5)$$

where  $j$  is the neuron index and  $i$  is the input index, with  $i = 1, \dots, n$ . The weighted sum of the neuron inputs is denoted by  $z$ . Sometimes is useful to introduce in the calculation of  $z$ , an extra weight from a constant input  $b_j$ . Adding this contribution to (5) we have:

$$z = \sum_{i=1}^n w_{ij} x_{ij} + b_j = [\mathbf{w}_j^T \ b_j] \begin{bmatrix} \mathbf{x}_j \\ 1 \end{bmatrix}, \quad (6)$$

where  $[\mathbf{x}_j \ 1]^T$  is the expanded input vector and  $[\mathbf{w}_j^T \ b_j]$  is the weights vector. The neuron output is obtained by passing the previously calculated sum through the activation function,  $\sigma_j$ :

$$\sigma_j(z) = \sigma_j \left( [\mathbf{w}_j^T \ b_j] \begin{bmatrix} \mathbf{x}_j \\ 1 \end{bmatrix} \right) \quad (7)$$

Neurons are usually organized in multiple layers. This organization defines the structure or architecture of the network. It was chosen to use in the problem of classification *feedforward networks*, which are composed of multiple neuron layers. The information flows only in one direction from layer to layer, from the inputs to the outputs.

The training or learning of these models is the process of adjusting the weights of the individual neurons in such a way that the difference between real and estimated outputs are minimized. One of the most used algorithms is *backpropagation* (BP), [4], [5].

### C. Fuzzy Modeling

Fuzzy models are systems that describe relations in terms of rules. They are flexible mathematical structures that can cope with ambiguity. More than “black-box” models, they are “grey-box” models, [5], since their rule-based nature allows for a linguistic description of knowledge. Hence, it is possible to incorporate expert knowledge of the system into the model.

Takagi-Sugeno fuzzy models [6] are used. In this type of fuzzy models the consequents are crisp functions instead of linguistic propositions (linguistic or Mamdani fuzzy models). In classification each discriminant is computed based on a group of rules. Hence, for each discriminant function,  $d^{(j)}$ , we have:

$$\begin{aligned} R_i^{(j)} : & \text{If } x_1 \text{ is } A_{i1}^{(j)} \text{ and } \dots \text{ and } x_{N_f} \text{ is } A_{iN_f}^{(j)} \\ & \text{then } d_i^{(j)}(\mathbf{x}) = f_i^{(j)}(\mathbf{x}), \quad i = 1, 2, \dots, R, \end{aligned} \quad (8)$$

where  $R_i^{(j)}$  is the  $i^{th}$  rule for class  $j$  and  $R$  is the number of rules. The function  $f_i^{(j)}$  is the consequent for rule  $R_i^{(j)}$ . Note

that  $d^{(j)}(\mathbf{x})$  can be interpreted as a score for the associated class  $j$  given the input vector  $\mathbf{x}$ . To aggregate all the rules it is first necessary to calculate the degree of fulfillment DOF of each rule.

$$\beta_i^{(j)} = \prod_{k=1}^{N_f} \mu_{A_{ik}}^{(j)}(\mathbf{x}), \quad \mu_{A_{ik}}^{(j)} : \mathbb{R}^{N_f} \rightarrow [0, 1]. \quad (9)$$

The discriminant output for each class is given by:

$$d^{(j)}(\mathbf{x}) = \frac{\sum_{i=1}^R \beta_i^{(j)} f_i^{(j)}(\mathbf{x})}{\sum_{i=1}^R \beta_i^{(j)}}, \quad j = 1, \dots, l \quad (10)$$

A discriminant function  $d^{(j)}(\mathbf{x})$  can be interpreted as a score for class  $j$ , given the input vector  $\mathbf{x}$ . Therefore, a sample is assigned to class  $j$  which has the maximum discriminant value,  $\max_j d^{(j)}(\mathbf{x})$ .

The antecedent fuzzy sets  $A_{ik}$ , and the consequent parameters  $f_i^{(j)}$  are determined in this step, using fuzzy clustering in the product space of the input and output variables, [7]. It is an unsupervised learning method that organizes and categorizes data based on the similarity of data objects. In this work, the chosen clustering method was fuzzy C-means (FCM), [8], since it demonstrated to be the most suitable for classification.

#### D. Support Vector Machines

Support vector machines (SVM) is a method for learning separating functions in two-class classification problems, [5]. The algorithm maps the non-linear inputs to a high dimension feature space. In this feature space a linear classification surface is constructed. SVM are an easy to use classification technique even though users usually get unsatisfactory results at first due to poor parameter setup. SVM are widely employed in computational biology due to their high accuracy, their ability to deal with high-dimensional and large databases, and their flexibility in modeling diverse sources of data, [9].

The first version of SVM was introduced as a linear classifier. However, most classification problems can only be separated using a non-linear classifier. The solution is to project the data in a superior *features space*  $\mathcal{F}$ :

$$\begin{aligned} \phi : \mathbb{R}^N &\rightarrow \mathcal{F}, \\ \mathbf{x} &\rightarrow \phi(\mathbf{x}), \end{aligned} \quad (11)$$

with  $\phi$  being the non-linear mapping function between  $\mathbb{R}^N$  and  $\mathcal{F}$ . After this operation one works with the new database:

$$(\phi(\mathbf{x}_1), y_1), \dots, (\phi(\mathbf{x}_n), y_n) \in \mathcal{F} \times Y, \quad (12)$$

The discriminant function will be in the form:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b, \quad (13)$$

In the feature space,  $\mathcal{F}$ , data is linearly separable. Further, when viewed in the original input space,  $f$  is a non-linear

function if  $\phi(\mathbf{x})$  is a non-linear function. The dual problem in  $\mathcal{F}$  becomes:

$$\begin{aligned} \text{maximize}_{\alpha} \quad & L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) \\ \text{subject to:} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n, \\ & \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned} \quad (14)$$

A *kernel function* is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (15)$$

As will be seen, with this function it is only necessary to compute the feature vectors  $\mathbf{x}$ , since the scalar products  $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$  are directly calculated by computing the kernel  $K(\mathbf{x}_i, \mathbf{x}_j)$ .

The chosen training algorithm, used to solve the QP problem of SVM, was *sequential minimal optimization* (SMO), [10].

### III. WRAPPER METHODS

The main characteristic of wrapper methodologies is the involvement of the predictor as part of the selection procedure. They use a learning machine as a “black box” to score the subsets according to their predictive performance [2]. Wrappers are constituted by three main components:

- 1) Search method;
- 2) Learning machine;
- 3) Feature evaluation criteria.

Wrapper approaches are aimed to improve the results of the specific predictors they work with. During the search subsets are evaluated without incorporating knowledge about the specific structure of the classification or regression function [2].

#### A. Encoding

The use of an optimization algorithm in parallel with a modeling technique presupposes that each solution will be represented in a manner, that allows information to be correctly exchanged. Further, if the optimization algorithm is searching simultaneously for the optimal feature subset and model parameters, both solution have to be properly combined.

There are various ways of encoding a problem solution, the most common and more generic are real, integer and binary encoding. The use of each of them depends on the problem in hand. Normally in feature selection, the search space organization is made, such that each state represents a feature subset [2]. In a problem with  $N_t$  variables, a state is encoded by a sequence of  $N_t$  bits, each bit indicating whether a feature is present or absent. An example of a possible state is represented by the sequence:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iN_t}) = (1, 0, \dots, 1), \quad (16)$$

The variable  $x_{ij}$  corresponds to input  $F_j$ , where  $j = 1, \dots, N_t$ . If feature  $F_j$  is to be selected then  $x_{ij} = 1$  if not  $x_{ij} = 0$ . This process is illustrated in Figure 1.

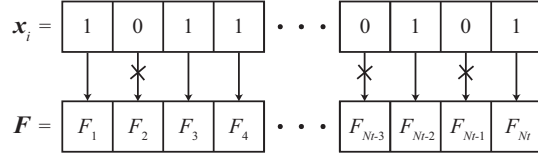


Fig. 1. Decoding process in feature selection.

One of the major problems when using wrapper methods is the process of selecting a proper set of parameters, specially in the case of SVM. These problems can be surpassed by automatically optimizing model parameters in parallel with the FS process, which is possible due to the use of binary encoding. A part of the binary sequence refers to the feature selection (whether to accept or discard a feature) and the rest to the model parameters. Nevertheless, the parameters are usually real valued, thus it is necessary to decode the binary strings in to floating point values. For each parameter we have a specific part of the binary string that is decoded independently to a selected range. To convert the binary code to a floating point number (17) is used, note that  $p_{min}$  and  $p_{max}$  can take different values.

$$p = p_{min} + \frac{(p_{max} - p_{min})}{1 - 2^{-u}} \times \times (2^{-1}g_1 + \dots + 2^{-i}g_i + \dots + 2^{-u}g_u), \quad (17)$$

where  $p$  is the parameter value,  $g_i$  is the value of the bit  $i$  and  $u$  the number of bits used to encode the parameter value. If this value belongs to  $\mathbb{Z}$ , then  $p$  is rounded to the closest integer.

### B. Objective Function

Through this work we have used feature selection with the purposes of improving modeling quality. The problem in study can be of two different types, classification or function modeling. However, the goals are roughly the same, maximizing model exactitude and minimizing the number of used features. Traditionally, accuracy has been used to evaluate classifier performance. This measure is defined as the total number of good classifications over the total number of available examples. Usually most of the classification problems have two classes, positive and negative cases. Thus, the classified test points can be divided in four categories: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

Given the accuracy formula can be written as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (18)$$

This criteria is limited, specially in medical applications, for various reasons. If one of the classes is more underrepresented than the others, misclassifications in this class will not have a great impact in the accuracy value. Also, a good classification of a class might be more important than classifying other classes, and this can not be assessed with accuracy. To take this matter into account, there have been introduced two performance measures:

$$Sensitivity = \frac{TP}{TP + FN}, \quad (19)$$

$$Specificity = \frac{TN}{FP + TN}. \quad (20)$$

Sensitivity is the equal to the  $TP$  rate, i.e. the ratio of true positives that were identified in all the positive class samples. Analogously, specificity is the rate of the  $TN$ .

Recall that there are two main objectives in the FS problem: maximize the model accuracy and minimize the size of the feature subset. The objective function will be defined as a fitness function (as in EA), being our goal its maximization. The most suitable representation to our task was found to be:

$$f(\mathbf{x}_i) = \alpha \left(1 - \frac{N_e}{N_s}\right) + \gamma \left(1 - \frac{N_f}{N_t}\right), \quad (21)$$

where  $N_e$  is the number of misclassifications,  $N_s$  is the total number of tested samples, and  $N_f$  is the size of the feature subset. The term on the left side of the equation accounts for the overall accuracy and the term on the right for the percentage of used features. Note that both the terms in the objective function are normalized. Constants  $\alpha, \gamma \in [0, 1]$  are the weights of the related goal; accuracy and subset size, respectively.

The problem of finding an optimal feature subset using the above objective function (fitness function) is highly complex. Thus, various metaheuristics are used to search the space of feature combinations.

## IV. METAHEURISTICS IN FEATURE SELECTION

Metaheuristics are general upper level (*meta*) algorithmic techniques, that can be used as guiding strategies in the design of heuristics to solve specific optimization problems. Those techniques are capable of finding acceptable solutions, in reasonable time, by using experience-based techniques or through guided search, but do not guarantee that the optimum is found.

### A. Genetic algorithms

Genetic algorithms, [11], belong to the class of evolutionary algorithms (EA), which are computational techniques based on processes similar to those happening during biological evolution. GA performs a series of iterative computations in order to evolve a population of individuals (possible solutions), using the principle of *survival of the fittest* [12]. These steps are the following:

- Step 1: Initialize the population;
- Step 2: Evaluate each chromosome;
- Step 3: Selection according to the fitness;
- Step 4: Recombination using crossover;
- Step 5: Random mutations;
- Step 6: Replacement of the old population by the new;
- Step 7: Repeat from Step 2 until the termination criteria is met.

1) *Selection*: Selection is a fitness based method, used with the purpose of choosing the most suited chromosomes in the population to form new individuals.

2) *Crossover*: Two individuals are selected from the population (applying two times the chosen selection operator) and are then recombined with a probability  $p_c$ , creating two new individuals. This is done by generating a random number  $r \in [0, 1]$ . If  $r \leq p_c$ , the two individuals are combined through crossover.

3) *Mutation*: For the case when a chromosome is represented by a bit string, each allele has a value  $\in \{0, 1\}$ . Thus, the mutation operator is simply a bit flip of the current value of each bit.

4) *Replacement and Elitism*: Replacement is the process by which the new individuals, created with the above operators, are introduced in the population. There are cases when the fittest individual in the population is replaced by an individual with lower fitness. Therefore, the elitism reintroduces  $n_c$  copies of the best individual into the population.

### B. Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic population-based metaheuristic, inspired in swarming behavior of some biological species (e.g. fish schools, bird flocks). Essentially, in PSO, each particle (corresponding to individuals in EA) is a candidate solution of the optimization problem. A particle has associated a position and a velocity in the search space, where the method for determining the changes in velocity depends on the particle itself and the other particles. The iterative process in search of the optimum is:

- Step 1: Evaluate each particle in the swarm;
- Step 2: Find the swarm and particle best values;
- Step 3: Update velocities;
- Step 4: Update positions of the particles;
- Step 5: Go to Step 1 if not finished/stop criteria.

There are two steps that are crucial in the way the algorithm operates, the update of velocities and update of particle positions.

1) *Update velocities*: Velocity directs the movement in the search space taking into account the performance of the own particle and of the swarm, and it is update with the following equation:

$$v_{ij} \leftarrow wv_{ij} + c_1q(x_{ij}^{pb} - x_{ij}) + c_2r(x_j^{sb} - x_{ij}) \quad (22)$$

$i = 1, \dots, N, j = 1, \dots, N_t.$

The term involving constant  $c_1$  is called the *cognitive component* and the term involving  $c_2$  is the *social component*.  $q$  and  $r$  are uniform random numbers  $\in [0, 1]$ . Once velocities have been update, the restriction  $|v_{ij}| < v_{max}$  is applied; this is a crucial step for the swarm to maintain coherence.

2) *Update particle position*: The logistic function of the velocity is used as the probability distribution for the position, [13]:

$$\sigma(v_{ij}) = \frac{1}{1 + e^{-v_{ij}}} \quad (23)$$

Thus the particle position is calculated for each variable by:

$$x_{ij} \leftarrow \begin{cases} 0, & \text{if } r > \sigma(v_{ij}) \\ 1, & \text{otherwise} \end{cases} \quad i = 1, \dots, N, j = 1, \dots, N_t \quad (24)$$

### C. Mechanisms to avoid premature convergence in PSO

Medical data characteristically has a high number of variables (high dimension). Thus, for this type of data, it is complex to perform feature selection efficiently. The binary version of particle swarm optimization (BPSO) has the tendency to prematurely converge as noted by [13], [14], [15], especially in more challenging optimization tasks. There are various approaches to cope with this problem: using the mutation operator from EA, [14], reset the swarm best if the fitness stagnates, [15] or using perturbation mechanisms, [13]. In this novel approach we associate the benefits of local search (mutations) with the resetting the swarm best mechanism. These mechanisms have individual characteristics that contribute to an improvement of the solutions when there is premature convergence of the binary PSO. The first two mechanisms were proposed in [16]

1) *Mutations*: After updating the particle position with (23) and (24), each of the bits of the position vector is mutated with a probability  $pr_{mut}$ . Normal values for the mutation probability are  $r_{mut} = 1/N_t$ , which means that at least one of the bits in the position vector will be flipped. This method introduces diversity in the swarm, but in problems with higher dimensionality, it may not be enough to prevent the algorithm from stagnating. In that cases we apply the following method.

2) *Reset swarm best*: Each particle adjusts its position according to two values, its own best solution so far,  $x^{pb}$ , and the swarm best solution  $x^{sb}$ . The particle best is a local search value, whereas the swarm best constitutes a global search value. If the  $x^{sb}$  value is itself trapped in a local optimum, a search of each particle will be limited, thereby preventing superior results of classification. By resetting  $x^{sb}$  we can avoid the binary PSO to get trapped in a local optimum, and superior classification result can be achieved by searching for a new  $x^{sb}$  value in a region with a lower number of features. This process is done by assigning to  $x^{sb}$  a vector of length  $N_t$  with all bits equal to 0 except one of them, that is set to 1 (chosen randomly).

The above mechanism were combined in a novel PSO (NPSO) approach presented in [16]. However, it was noticed that some of its aspects could be improved so another approach which consists in an evolution of the NPSO was presented. This simplified novel PSO (SNPSO) combines the reset swarm best mechanism with the two following operators:

3) *Local search*: This mechanism consists of displacing the  $x_i^{pb}$  values when resetting the swarm best:

$$\begin{cases} x_{ij}^{pb} = \neg x_{ij}^{pb} & \text{if } r \leq dr \\ x_{ij}^{pb} = x_{ij}^{pb} & \text{otherwise} \end{cases}, \quad i = 1, \dots, P, \quad j = 1, \dots, N_t, \quad (25)$$

where  $dr$  is the displacement rate, i.e. the probability of each bit in the  $x_i^{pb}$  being flipped and  $r$  a random number  $\in [0, 1]$ .

4) *Variability in the swarm*: The mutations mechanism introduces small “mistakes” during position update, avoiding premature convergence. However, this can be alternatively made by controlling the value of  $v_{max}$ . This simplifies the

algorithm and is more effective when particle position stagnates due to the saturation of the velocity. To demonstrate the improvements achieved by this novel PSO approach, FS was performed in the various dataset from [17].

## V. RESULTS

The main objective in this section is to evaluate the applicability of the proposed wrapper methods to large databases. These methods combine the machine learning algorithms introduced in Section II, with the search algorithms presented in Section IV, using the approach in Section III. The chosen model evaluation measure was accuracy based on the holdout method (Section III-B). The used data division was 65% of data for training and 35% for testing, since the generated models shown the best value of test accuracy.

To verify the informative potential of each final subset of features, found by the search method (optimization method), two measures were used. The first was the accuracy obtained from 10-fold cross validation, using the best subset of features found by the algorithm. The other is a statistical measure called p-value, which is obtained from a paired t-test, [18]. If the p-value is less than 0.05, it means that the compared results are different with a confidence level greater than 95%.

### A. Results for the BPSO algorithms

To compare the several PSO approaches the sonar database was used. This is a benchmark database from the UCI repository [17], it is a binary classification problem, it has 60 features and 208 samples. In this work, two novel techniques, NPSO and SNPSO, have been introduced. To assess the quality of these algorithms, in the problem of FS, a comparison was made with previously suggested techniques inspired in PSO, the original BPSO, the BPSO+Mutations approach, and the IBPSO.

TABLE I  
PARAMETERS FOR THE DIFFERENT BPSO-BASED APPROACHES.

	BPSO	BPSO+Mutations	IBPSO	NPSO	SNPSO
$v_{max}$	6	4	6	6	4-5
$p_{mut}$	-	$1/N_f$	-	$1/N_f$	-
$r_{mut}$	-	-	-	-	$0.5/N_f$
Reset $x_{sb}$	-	-	3	3	4

The various BPSO inspired algorithms were combined with three modeling techniques, NN, FM and SVM in a wrapper approach, and evaluated over the sonar database.

The selection of the appropriate measure is one of the most important steps in the comparison between methodologies. The objective in this section is to compare the search capabilities of the multiple BPSO-based approaches. Thus, the most fair measure to use is the fitness function defined in (21), since it is the function that the algorithms have to optimize during search. For this analysis, the selected weights of the fitness function were  $\alpha = 0.7$  and  $\beta = 0.3$ , which result from a sensitivity analysis.

Looking at the comparison in Fig. 2, it can be seen that there are great differences between algorithm performances.

The most noticeable, is that the method using the *reset swarm best* operator (IBPSO, NPSO, SNPSO) find considerably better results than the rest. The explanation to this is simple, the algorithms with the reset swarm best operator are biased towards the selection of less features. Thus, they are able to find comparable in terms of CV accuracy using less features. Furthermore, NPSO and SNPSO approaches are always better than the IBPSO, this is due to their mechanisms that introduce variability in the swarm. Nevertheless, these differences are not as pronounced when comparing the IBPSO-Fuzzy with the SNPSO-Fuzzy. It may be due to a poor adjustment of the  $dr$ , since this parameter has to be well adjusted in order to compensate an increase in the reset  $x_{sb}$  iterations.

### B. Results for Sepsis Outcome Prediction

For a question of simplicity only one modeling technique will be chosen to continue the analysis. The case study of this work consists in the application of the presented FS techniques to a medical problem. In such case the most important issue is to have a classifier with the best accuracy possible. Therefore, it has been decided to proceed using the SVM-based wrapper methods, once it is the modeling technique which has better accuracy for the Sonar database. The models generated by SVM will be optimized through feature selection using SNPSO which was the PSO with better performance in the previous study. Thus, in this section this method will be validated against GA, in order to prove its multiple advantages.

This paper uses data from the MEDAN, [19], multi-center study of 71 ICUs in Germany. For outcome prediction the data of 382 patients was analyzed by using most of the commonly documented vital parameters and doses of medicine (metric variables). Data was collected in German hospitals from 1998 to 2001.

1) *Data Preprocessing*: We follow the preprocessing procedures done in [20]. It focus exclusively in physiological parameters, which include a total of 103 different variables (features). From the initial set containing a total of 103 features, two different subsets of features were chosen. One contains the 12 most frequently measured variables, present only in the data of 121 patients. However in [20] the previous subset of features was considered too narrow, and a second data subset was defined including a total of 28 variables. These were found to be present in a total of 89 patients. These two sets of features are described in Table II.

TABLE II  
SEPSIS DATABASES

Features	Number of Patients	Samples	Classes	References
12	89	2878	2	[21]
28	121	2055	2	[20]

2) *Results using 12 features*: The first sepsis database to be studied was the database with 12 features. The benefits of FS over this database were evaluated with the chosen GA-SVM and SNPSO approaches. The results are summarized in

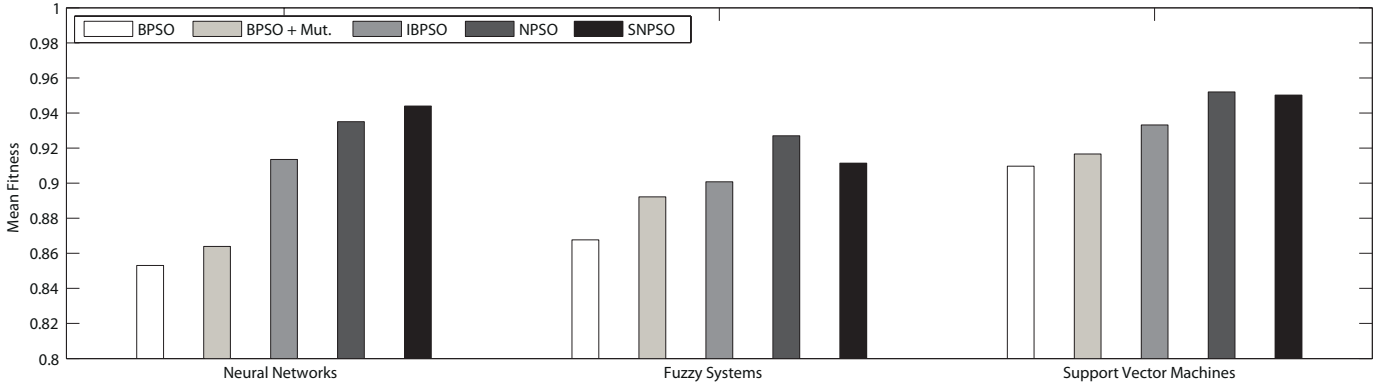


Fig. 2. Comparison of various BPSO approaches using the sonar database.

Table III. Comparing the results of the CV accuracy before and after FS, no significant differences were observed. However, the value of sensitivity increased after FS, without noticeable differences in the specificity. This may be caused by the reduction in the feature subset size from 12 (initial) to 3 (after FS) on average. The run time using the 12 features sepsis database was substantial when compared with the average value over the benchmark databases. It might be a consequence of the elevated number of available samples, which slows down the training process of SVM. Further, the SNPSO is 3 times faster than GA, and it was even faster than the No-FS case (parameter selection using grid-search).

TABLE III  
COMPARISON OF FS TECHNIQUES OVER THE 12 FEATURES SEPSIS DATABASE.

Method	Accuracy	Time (s)	NF	Sensitivity	Specificity
No-FS	72,6 $\pm$ 2,6	10908	12 $\pm$ 0	47,6 $\pm$ 5,6	88,1 $\pm$ 2,8
GA	75,0 $\pm$ 3,8	25419	3 $\pm$ 1	52,2 $\pm$ 6,2	89,1 $\pm$ 3,5
SNPSO	76,5 $\pm$ 1,8	7817	2 $\pm$ 1	60,4 $\pm$ 4,1	86,5 $\pm$ 1,9

3) *Results using 28 features:* The informative potential of the 12 feature database was found to be very insufficient [20]. Thus, a second database including a total of 28 variables was constructed. The result was a considerable increase in the predictive performance of the generated classifier, as is depicted in Table IV.

Regarding Table IV, it can be seen that the CV accuracy over the 28 features database was considerably increased after FS (around 6%). Also, an initial sensitivity of 76,10% was improved to an average around 92%, while maintaining the good specificity. This can probably be justified by the reduction of the number of redundant and noise feature used to generate SVM surface, which in this case, correspond to 3/4 of the initial features. The run time of the FS methods over this database is large, like when using the 12 features database, since it has a high number of samples.

4) *General discussion:* To better assess the advantages of each database (12 and 28 features), the results of Table III and Table IV were summarized in Table V. Looking at Table V, one can see that the CV accuracy is substantially better when

TABLE IV  
COMPARISON OF FS TECHNIQUES OVER THE 28 FEATURES SEPSIS DATABASE.

Method	Accuracy	Time (s)	NF	Sensitivity	Specificity
No-FS	89,0 $\pm$ 1,7	5047	28 $\pm$ 0	76,1 $\pm$ 5,4	95,6 $\pm$ 1,7
GA	95,7 $\pm$ 1,4	7385	7 $\pm$ 1	94,3 $\pm$ 1,2	96,5 $\pm$ 2,1
SNPSO	94,4 $\pm$ 1,2	4610	6 $\pm$ 1	90,2 $\pm$ 5,1	96,5 $\pm$ 1,9

using 28 features. However, the average number of selected features more than doubles from when using the 12 and the 28 features.

The last comparison was made regarding the performance of the GA-SVM and SNPSO in the improvement of sepsis outcome prediction. As depicted in Table VI, both a GA-SVM and SNPSO-SVM have considerably better CV validation accuracies than the initial set of features. The set of initial features is in both cases reduced, on average, to 1/4 of the initial size. Furthermore, it is interesting to see that the SNPSO is faster than the No-FS case (parameter selection using grid-search) and approximately 2 times faster than the GA.

The best performance of all wrapper methods is achieved over the 28 feature database. Therefore, a comparison was made between the studied wrapper methods and the state of the art methods for FS in the problem of sepsis outcome prediction, using the 28 feature database. These wrapper approaches consist in bottom-up (BU) using fuzzy modeling, [20] and ant feature selection (AFS) using also fuzzy modeling, [22]. The results are depicted in Table VII.

The differences between the proposed approaches and the other two methods are significant. The GA-SVM and SNPSO-SVM have superior performances in all the measures that evaluate the generalization capabilities of the generated model (e.g. accuracy, sensitivity, specificity). This comparison is solely illustrative, since the methods use different modeling techniques. However, this might be an indicator that SVM have better generalization properties over this database than fuzzy models.

## VI. CONCLUSIONS

The algorithms inspired in binary PSO were tested over the sonar database, with the proposed algorithms showing a



TABLE V  
COMPARISON AVERAGE VALUES OVER THE 12 AND 28 FEATURES SEPSIS DATABASES.

Database	Accuracy	Time (s)	NF	p-value
MEDAN-12	74,7	14715	3	0.0049
MEDAN-28	93,0	5681	7	

TABLE VI  
COMPARISON OF AVERAGE VALUES OF THE USED SEARCH METHODS OVER THE MEDAN DATABASES.

Method	Accuracy	Time (s)	NF	p-value
No-FS	80,8	7978	20	—
GA	85,4	16402	5	$1,8 \times 10^{-5}$
SNPSO	85,4	6214	4	$9,3 \times 10^{-5}$

better performance than the state of the art methods for PSO. Some limitations to the proposed PSO inspired algorithms will arise if there is a poor adjustment in the parameters, specially in the case of simplified novel PSO. Further, the novel PSO and the simplified novel PSO have two more adjustable parameters than the standard version of binary PSO. Nevertheless, they have only one more parameter than genetic algorithm which has a inferior overall performance. Despite all the mechanisms for premature convergence, the proposed PSO inspired algorithms were able to correctly estimate the parameters of support vector machines, in parallel to feature selection.

The generation of models for evaluation of the mortality risk, in patients with sepsis, is still a very important topic in medicine. The application of SVM-based feature selection over the case study of sepsis outcome prediction, shown to be superior than all the previously used methods [20], [23], [21]. However, in this application example, one of the major problems of SVM became clear, the run time for training severely increased for problems with numerous samples.

#### ACKNOWLEDGMENT

This work is supported by the Portuguese Government under the programs: Programa de financiamento Plurianual das Unidades de I&D da FCT (POCTI-SFA-10-46-IDMEC), project PTDC/SEM-ENR/100063/2008, and by a FCT grant SFRH/BPD/65215/2009, Fundação para a Ciência e a Tecnologia (FCT), Ministério do Ensino Superior, da Ciência e da Tecnologia, Portugal.

#### REFERENCES

- [1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, pp. 37–54, 1996.
- [2] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, Eds., *Feature Extraction: Foundations and Applications*. Springer, 2006.
- [3] J. M. C. Sousa and U. Kaymak, *Fuzzy Decision Making in Modeling and Control*. World Scientific Publishing Company, 2002, vol. 27.
- [4] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, 1997, a Computational Approach to Learning and Machine Intelligence.
- [5] V. Kecman, *Learning and Soft Computing Support Vector Machines, Neural Networks, Fuzzy Logic Systems*. MIT Press, 2001.

TABLE VII  
COMPARISON OF VARIOUS WRAPPER METHODS OVER THE 28 FEATURES MEDAN DATABASE.

Method	Accuracy		NF	Sensitivity		Specificity	
	Mean	Std		Mean	Std	Mean	Std
BU-Fuzzy	82,4	1,6	2-7	82,3	1,6	83,3	2,6
AFS-Fuzzy	78,6	1,4	3-9	79,2	0,04	78,2	0,03
GA-SVM	95,7	1,4	6-8	94,3	1,2	96,5	2,1
SNPSO-SVM	94,4	1,2	5-7	90,2	5,1	96,5	1,9

- [6] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, p. 7, February 1993.
- [7] R. Babuska, *Fuzzy Modeling for Control*, 1st ed., ser. International Series in Intelligent Technologies. Norwell, MA, USA: Kluwer Academic Publishers, April 1998.
- [8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.
- [9] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Rtsch, "Support vector machines and kernels for computational biology," *PLoS Comput Biol*, vol. 4, no. 10, p. e1000173, 10 2008.
- [10] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, Tech. Rep., 1998.
- [11] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1998.
- [12] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231–240, 2006.
- [13] L. Yuan and Z.-D. Zhao, "A modified binary particle swarm optimization algorithm for permutation flow shop problem," in *Proc. of the International Conference on Machine Learning and Cybernetics*, vol. 2, August 2007, pp. 902–907.
- [14] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms," in *Proc. of the IEEE Congress on Evolutionary Computation*, September 2007, pp. 284–290.
- [15] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.
- [16] G. J. Farinha, S. M. Vieira, L. F. Mendonça, and J. M. Sousa, "Optimization of fuzzy models using a novel pso algorithm : Application to medical databases," in *Proc. of the 18th World Congress of the International Federation of Automatic control*, 2011, p. 6.
- [17] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: [www.archive.ics.uci.edu/ml](http://www.archive.ics.uci.edu/ml)
- [18] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, December 2006.
- [19] E. Hanisch, R. Brause, B. Arlt, J. Paetz, and K. Holzer, "The medan database," 2003. [Online]. Available: [www.medan.de](http://www.medan.de)
- [20] A. S. Fialho, F. Cismonti, S. M. Vieira, J. M. da Costa Sousa, S. R. Reti, M. D. Howell, and S. N. Finkelstein, "Predicting outcomes of septic shock patients using feature selection based on soft computing techniques," in *IPMU(2)*, ser. Communications in Computer and Information Science, E. Hllermeier, R. Kruse, and F. Hoffmann, Eds., vol. 81. Springer, 2010, pp. 65–74.
- [21] J. Paetz, B. Arlt, K. Erz, K. Holzer, R. Brause, and E. Hanisch, "Data quality aspects of a database for abdominal septic shock patients," *Computer Methods and Programs in Biomedicine*, vol. 75, no. 1, pp. 23–30, 2004.
- [22] S. M. Vieira, J. a. M. C. Sousa, and T. A. Runkler, "Two cooperative ant colonies for feature selection using fuzzy models," *Expert Syst. Appl.*, vol. 37, pp. 2714–2723, April 2010.
- [23] J. Paetz, "Knowledge-based approach to septic shock patient data using a neural network with trapezoidal activation functions," *Artificial Intelligence in Medicine*, vol. 28, no. 2, pp. 207–230, 2003.