2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Binary Fish School Search Applied to Feature Selection: Application to ICU Readmissions

João A. G. Sargo, Susana M. Vieira, João M. C. Sousa, and Carmelo J.A. Bastos Filho

Abstract—**This paper proposes a novel feature selection approach formulated based on the Fish School Search (FSS) optimization algorithm, intended to cope with premature convergence. In order to use this population based optimization algorithm in feature selection problems, we propose the use of a binary encoding scheme for the internal mechanisms of the fish school search, emerging the binary fish school search (BFSS). The suggested algorithm was combined with fuzzy modeling in a wrapper approach for Feature Selection (FS) and tested over three benchmark databases. This hybrid proposal was applied to an ICU (Intensive Care Unit) readmission problem. The purpose of this application was to predict the readmission of ICU patients within 24 to 72 hours after being discharged. We assessed the experimental results in terms of performance measures and the number of features selected by each used FS algorithms. We observed that our proposal can correctly select the discriminating input features.**

I. Introduction

The information age is very hard to grasp. In an average person's life nowadays, we get more information in a day than someone who lived 100 years ago would get in a lifetime. The speed at which information is increasing means that finding accurate data is becoming more important than the data itself [1]. With the urgent need for a new generation of computation techniques and tools to assist humans in extracting useful information from a fast growing volume of data, a methodology was created, called Knowledge Data Discovery (KDD), first introduced by Fayyad in 1996 [2].

The KDD process comprises a series of steps to extract knowledge from data: 1) Data Acquisition - the process of acquiring and storing data; 2) Data Preprocessing - consists of applying proper techniques that allow the improvement of the overall quality of the data, includes processing of noise/outliers, correction of missing values, and/or alignment of data sampled at different frequencies; 3) Feature Selection (FS) - consists of finding useful features to represent the data and discarding the non-relevant ones, *i.e.* the ones that contain redundant information; 4) Modeling - refers to the process of combining methods from computational intelligence and/or statistics to extract patterns in data sets. In this work, classification models (fuzzy modeling) were used; and finally 5) Interpretation - the process of evaluating the discovered knowledge with respect to its validity, usefulness, novelty, and simplicity. External expertise may be required in this step.

J.Sargo, S. Vieria, and J.Sousa are with the LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal (email: {joao.sargo, susana.vieira,jmsousa}@tecnico.ulisboa.pt).

Carmelo J. A. Bastos Filho, IEEE SM, University of Pernambuco (UPE), Recife, Brazil (email: carmelofilho@ieee.org)

This work was supported by FCT, through IDMEC, under LAETA Pest-OE/EME/LA0022 and partially supported by the project PTDC/EMS-SIS/3220/2012 - IC4U, IDMEC, and by a FCT post-doctoral grant SFRH/BPD/65215/2009, Ministério da Educação e Ciência, Portugal.

The field of feature selection has been object of extensive research in recent years [3]. This can be explained due to the potential benefits introduced by data dimensionality reduction. This can greatly improve data visualization and understanding, facilitating knowledge discovery. Furthermore, if a lower number of features is used, less information needs to be measured and stored, leading to simpler equipments, and consequently, reducing unnecessary costs. From the clinical point of view, this process may bring to light new variables that had not been previously considered as relevant for a given medical problem [4].

However, in real-world systems, the selection of a low number of features that consistently describe the problem is usually time consuming and, in many cases, impossible to achieve with a greedy approach [5]. Thus, the maximization of the model performance and the minimization of the number of used features depends on the chosen combination of features, often making the FS problem NP-Hard. Metaheuristics, such as Particle Swarm Optimization (PSO), have shown to be well suited for this type of problems, due to their randomized nature [5,6]. The main advantage of using Metaheuristics is that they are able to find good solutions, without having to try all possible combinations. However, the best Metaheuristic to solve a problem depends on the properties of the tackled problem.

In this paper, We propose a novel approach for binary variables based on the Fish School Search (FSS) optimization algorithm [7], named binary fish school search algorithm (BFSS). BFSS applied to the FS problem intends to present the capability to select discriminating input features and also to achieve high classification accuracy.

The clinical problem addressed is the readmission problem. The motivation relies on the fact that patients readmitted to an Intensive Care Unit (ICU) during the same hospitalization period have an increased length of stay, higher costs and increased risk of death [8,9,10]. The purpose of this case study is to predict the readmission of ICU patients within 24 to 72 hours after being discharged, using real world data [4].

This paper is organized as follows: Section II introduces the formulation of the feature selection problem, where the main concepts of fuzzy modeling are also presented; Section III describes the original FSS; Section IV presents the proposed modification to the internal mechanisms of the FSS, leading to the BFSS algorithm; Section V depicts the results obtained in the benchmark test problems and the results obtained from readmission application case study; and Section VI presents the conclusions.

II. Feature Selection

Feature selection algorithms can be grouped into four categories: wrappers (used in this work), filters, hybrids and embedded [3, 9].

Wrappers use a search algorithm to search through the space of possible features and evaluate each subset by running a model on this subset. Wrappers can be computationally expensive and have a risk of over-fitting the model. Filters are similar to wrappers in the search approach, but instead of evaluating against a model, the features are selected by means of performance evaluation that does not require building a model. In embedded feature selection methods, similarly to wrapper methods, feature selection is linked to the classification stage. This link is much stronger in this case, since feature selection in embedded methods is included into the classifier construction. The main advantage of the wrapper method over embedded methods is a better coverage of the search space. And the main advantage of the wrapper method over filter methods is that the final selected subset is highly correlated with the chosen metrics to assess the performance in the wrappers. This means that the classifier or in this case the classifier. When the objective is to obtain a model as accurate as possible and the time to obtain it is not an issue, then the wrapper method is an advantageous choice.

A. Wrapper Methods

The main characteristic of wrapper methodologies is the involvement of the predictor as part of the selection procedure. In this paper, a learning machine was used as a "black box" to score the subsets according to their predictive performance [11].

Wrappers are constituted by three main components:

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

Wrapper approaches were aimed to improve the results of the specific predictors they work with. During the search, subsets were evaluated without incorporating knowledge about the specific structure of the classification [11].

B. Fuzzy Modeling

The fuzzy modeling technique was considered. This learning machine method allows approximation of nonlinear systems when there is little or no previous knowledge of the problem to be modeled [10, 4]. This tool supports the development of models around human reasoning (also referred to as approximate reasoning), and allows an element to belong to a set to a degree, indicating the certainty (or uncertainty) of its membership.

First order Takagi-Sugeno (TS) fuzzy models [12] were applied, which consist of fuzzy rules where each rule describes a local input-output relation. When first order TS fuzzy systems are used, each discriminant function consists of rules as shown in (1).

$$\begin{aligned} R_j &: if \;\; x_1 \;\; is \;\; A_{j1} \;\; and \;\; \ldots \;\; and \;\; x_N \;\; is \;\; A_{jN} \\ & then \;\; y_j = (a_j)^T x + b_j \end{aligned} \tag{1}$$

where, $j=1, \ldots, J$ corresponds to the rule number, $x=(x_1, \ldots, x_N)$ is the input vector, N is the total number of inputs (features), A_{jn} is the fuzzy set for rule R_j and n^{th} feature, and y_j is the consequent function for rule R_j. The degree of activation of the j^{th} rule is given by:

$$\beta_j = \prod_{n=1}^{N} \mu_{A_{jn}}(x), \tag{2}$$

where $\mu_{Ajn}(x){:}R \rightarrow [0,1]$. The number of rules j and the antecedent fuzzy sets A_{jn} were determined by means of fuzzy clustering in the product space of the input and output variables. In this work, the chosen clustering method was Fuzzy C-Means (FCM) [1], since it is suitable for classification.

Given a classification problem, and being a linear consequent, a threshold t is required to turn the continuous output $y \in [0,1]$ into the binary output $y \in \{0,1\}$. In this way, *if* $y < t$ *then* $y = 0$, and *if* $y \geq t$ *then* $y = 1$.

III. Fish School Search

The novel Binary Fish School Search algorithm (BFSS), formulated and presented in this paper, was created based on the optimization search algorithm Fish School Search (FSS), first proposed by Bastos-Filho and Lima Neto in 2008 [7].

Several oceanic fish species, as well as other animals, present social behavior. The main reason for this is to increase mutual survivability and may be viewed in two ways: (i) for mutual protection and (ii) for synergistic achievement of other collective tasks. Here, protection means reducing the chances of being caught by predators; and synergy, refers to an active mean of achieving collective goals, such as finding food. Some of the main characteristics derived from real fish schools inspired the mechanisms that were incorporated into the core of the approach soundly, and are grouped into two observable categories of behaviors. The operators are grouped accordingly as follows:

Feeding – food is a metaphor for indicating to the fish the regions of the aquarium that are likely to be good spots for the search process. Feeding operators consider variations in the fitness function and return variations on the weight of the fish;

Swimming – a collection of operators that are responsible for guiding the search effort globally towards subspaces of the aquarium that were collectively sensed by all individual fish as more promising with regard to the search process.

The concept of food was considered as related to the function to be optimized in the process, the fitness function. As an example, consider a minimization problem, the amount of food in a region would be inversely proportional to the fitness function evaluation in this region. The "aquarium" is defined by the delimited region in the search space where the fish can be positioned. The position of each fish represents a possible solution for the tackled problem. The original operators of the FSS are described as follows.

A. The Feeding Operator

As in real situations, the artificial fish of FSS are attracted to food scattered in the aquarium in various concentrations. In order to find greater amounts of food, each fish in the school can move independently (see individual movement in the next section). As a result, each fish is allowed to increase or decrease in weight, depending on its success or failure in obtaining food. The weight of the fish represents the mechanism to store the success of fish during the search process. The authors, in [7], proposed that fish's weight should vary proportionally to the normalized difference between the values of the fitness function at the previous and current fish position with regards to the food concentration on these spots:

$$W_i(t+1) = W_i(t) + \frac{f[\vec{x}_i(t+1) - \vec{x}_i(t)]}{\max\{|f[\vec{x}_i(t+1) - \vec{x}_i(t)]|\}, \forall i} \qquad (3)$$

where $W_i(t)$ is the weight of the fish i in the iteration t, $x_i(t)$ the position of the fish i in the iteration t and $f[\vec{x}_i(t)]$ evaluated the fitness function (*i.e.* amount of food) in $\vec{x}_i\ (t)$. A few additional measures were included to ensure rapid convergence toward rich areas of the aquarium, namely:

- Fish weight variation is evaluated once at every FSS cycle;
- An additional parameter, named weight scale (W_{scale}) was created to limit the weight of a fish. The fish weight may vary between 1 and W_{scale} .

All the fish are born with weight equal to $W_{scale}/2$.

B. The Swimming Operators

For fish, swimming is directly related to all the important individual and collective behaviours such as feeding, breeding, escaping from predators, moving to more liveable regions of the aquarium or, simply being gregarious. This panoply of motivations to swim away inspired the authors of [7] to group causes of swimming into three classes: a) individual, b) collective-instinct and c) collective volition.

Below further explanations on how computations are performed on each of them are provided.

C. Individual Movement

Individual movement occurs for each fish in the aquarium at every cycle of the FSS algorithm. The swim direction is randomly chosen. Provided the candidate destination point lies within the aquarium boundaries, the fish assess whether the food density there seems to be better than at its current location. If not, or if the step-size would be considered not possible (*i.e.* lying outside the aquarium or blocked by, say, reefs), the individual movement of the fish does not occur. Soon after each individual movement, feeding would occur, as detailed in the Section III.*A*.

For this movement, a parameter was defined to determine the fish displacement in the aquarium called individual step ($step_{ind}$). Each fish moves $step_{ind}$ if the new position has more food than the previous position. Actually, to include more randomness in the search process the individual $step_{ind}$ is multiplied by a random number generated by a uniform distribution in the interval [-1,1], represented as u in (4). In general, the individual step was decreased linearly in order to provide exploitation abilities in later iterations:

$$\begin{aligned} x_{ij}(t+1) &= x_{ij}(t) + u(-1,1)S_{ind}(t), \\ S_{ind}(t) &= step_{ind,initial} - (step_{ind,initial} - step_{ind,final})\frac{g_{current}}{g_{final}} \end{aligned} \qquad (4)$$

where $g_{current}$ is the number of the current iteration and g_{final} is the total number of iterations, for the fish i and dimension j.

D. Collective-Instinctive Movement

After the individual movement, a weighted average of individual movement based on the instantaneous success of all fish of the school is computed. This means that fish that had successful individual movements influence the resulting direction of movement more than the unsuccessful ones. When the overall direction is computed, each fish is repositioned. This movement is based on the fitness evaluation enhancement achieved, as shown in (5):

$$\begin{aligned} x_i(t+1) &= x_i(t) + I(t), \\ I(t) &= \frac{\sum_{i=1}^{N} \Delta x_{indi}\{f[x_i(t+1)]\} - f[x_i(t)]\}}{\sum_{i=1}^{N}\{f[x_i(t+1)]\} - f[x_i(t)]\}} \end{aligned} \qquad (5)$$

where Δx_{indi} is the displacement of the fish i due to the individual movement within the current FSS cycle.

E. Collective-Volitive Movement

After individual and collective-instinctive movements are performed, one additional positional adjustment is still necessary for all fish in the school: the collective-volitive movement. This movement is devised as an overall success/failure evaluation based on the incremental weight variation of the whole fish school. In other words, this last movement is based on the overall performance of the fish school during the current iteration. The rationale is as follows: if the fish school is putting on weight (meaning the search has been successful), the radius of the school should contract; if not, it should dilate. This operator is deemed to help greatly in enhancing the exploration abilities in FSS. This phenomenon might also occur in real swarms, but the reasons are as yet unknown. The fish-school dilation or contraction is applied as a small step drift to every fish position with regard to the school's barycenter. The fish-school's barycenter is obtained by considering all fish positions and their weights, as shown in (6):

$$Bari(t) = \frac{\sum_{i=1}^{N} x_i(t)W_i(t)}{\sum_{i=1}^{N} W_i(t)} \qquad (6)$$

For this movement, a parameter called volitive step ($step_{vol}$) was defined as well. The new position is evaluated as in (7) if the overall weight of the school increases in the FSS cycle; if the overall weight decreases, (8) should be used.

$$x(t+1) = x_i(t) - step_{vol}.u[x_i(t) - Bari(t)] \qquad (7)$$

$$x(t+1) = x_i(t) + step_{vol}.u[x_i(t) - Bari(t)] \qquad (8)$$

where u is a random number uniformly generated in the interval [0,1]. We also decreased the linear $step_{vol}$ along the iterations.

The FSS algorithm starts by randomly generating a fish school according to parameters that control fish sizes and their initial positions.

Regarding dynamics, the central idea of FSS is that all bio-inspired operators perform independently from each other. The FSS search process is enclosed in a loop, where invocations of the previously presented operators will occur until at least one stop condition is met. Stop conditions conceived for FSS are as follows: limitation of the number of cycles, time limit, maximum school radius and maximum school weight. Examples of the use of FSS algorithm can be visualized in [7].

IV. BINARY FISH SCHOOL SEARCH

Although there are numerous ways to encode the FSS algorithm in order to solve feature selection problems, after some initial tests, we decided to modify the internal mechanisms of the FSS algorithm to manipulate binary inputs.

The following sections describe the modifications to the original fish school search algorithm, emerging the binary fish school search (BFSS).

A. Encoding

There are various ways of encoding a problem solution. The encoding scheme presented here is inspired in [7]. An example of a possible state (position of a fish *i*) with a total of N_t features to be selected can be represented by the sequence:

$$x_i = (x_{i1}, x_{i2}, ..., x_{iN_t}) = (1,0,...,1) \,. \tag{9}$$

The state is encoded by a sequence of N_t bits, in which each bit indicates whether a feature is present or absent.

This binary scheme, offers a straightforward representation of a feature subset, allowing the algorithm to search through the workspace, adding or removing features, simple by flipping bits in the sequence. While the FSS algorithm was not originally developed in the context of binary encoding, it appeared to be possible to modify the real to a binary encoding, keeping the following principles:

- to follow the internal mechanisms of the original algorithm, without losing the meaning of each bio-inspired operator;
- to add few additional parameters;
- to ensure the convergence of the algorithm;
- to keep simplicity and understanding.

In the next sections, the modifications made to each of FSS internal mechanisms are presented.

B. Initialization of each fish

For each fish *i*, the initial position was initialized randomly by doing:

$$x_{ij} \leftarrow \begin{matrix} 1, & if \;\; u > 0.5 \\ 0, & otherwise \end{matrix} \quad , i = 1,..., N \quad j = 1,..., N_t \tag{10}$$

where u is a random number uniformly generated in the interval [0,1], N the number of fish and N_t the total number of features to be selected.

By doing this, the algorithm starts with a population of fish completely randomly positioned within the search space, being the number of features of each fish selected at the beginning of the algorithm around $N_t/2$. We have assumed this because the algorithm might not converge freely along iterations if the initial average number of features is too small.

C. Individual Movement

The Individual movement occurs once in every cycle of the BFSS. For each fish *i*, and for each feature *j*, if a random number u (uniform distribution in the interval [0,1]), is smaller than $S_{ind}(t)$ the bit will flip, otherwise it will not change:

$$x_{ij} \leftarrow \begin{matrix} \overline{x_{ij}}, if \;\; u < S_{ind\,(t)} \\ x_{ij}, otherwise \end{matrix} \quad , i = 1,..., N \quad j = 1,..., N_t \tag{11}$$

The Parameter S_{ind}, in the same way as the FSS, will decrease linearly along the iterations depending on the first value and the last value of $step_{ind}$. This allows a soft convergence through the iterations. A fish will move if the new position has more food than the previous position, *i.e.* if the fitness function of new set of features selected (new position) presents a better performance than the previous one. By doing this, the random exploration of each individual fish is preserved.

D. Collective-Instinctive Movement

After the individual movement, the weighted average of the individual movements, based on fish that had moved, is calculated. This process was executed in the same way as the FSS, equation (3).

In order to guide all fish to the direction provided by the successful individual movements some adaptations were made to the original FSS algorithm. When dealing with positions with bits, equation (5) loses its meaning. The displacement of the fish, Δx_{indi} in equation (5), can no longer be quantified correctly using the discrete flipping of a bit.

For that reason, equation (12) was used to describe the resultant position of the overall successful of the individual movement:

$$I(t) = \frac{\sum_{i=1}^{N} x_{indi} \{ f[x_i(t+1)] \} - f[x_i(t)] \}}{\sum_{i=1}^{N} \{ f[x_i(t+1)] \} - f[x_i(t)] \}} \,. \tag{12}$$

In (12), Δx_{indi} in (5) was replaced by x_{indi}. In this approach, the use of the current position of fish that had success in the individual movement is seen as being more descriptive than the flipping of bits.

The resulting vector $\vec{I}$ has the same dimension as the positions of the fishes, but with values varying between 0 and 1. As an illustrative example, (13) represents a possible configuration of $\vec{I}$:

$$I = [0.1 \quad 0.5 \quad 0 \quad 0.3 \quad ... \quad 0.7] \tag{13}$$

The goal of the Collective-Instinctive Movement operator is to attract each fish to the resultant direction of the individual movement operator. In the BFSS, each fish must follow $\vec{I}$. Therefore, it is necessary to have bit format to allow this. After some preliminary tests, we deployed an adaptive threshold per iteration. For a given $\vec{I}(t)$, a threshold was used, multiplying the parameter *thres_c* by the max value of $\vec{I}(t)$. The resultant value of this multiplication would then be used as a threshold in the current iteration for this operator: if the values of the bits of $\vec{I}(t)$ were below the threshold, they would be considered 0, otherwise 1.

For the example (13), and if the parameter $thres\_c$ was 0.4, the threshold used in this iteration would be 0.4*x*0.7=0.28. Considering that 0.7 is the maximum value depicted in (13), the resulting *I* is:

$$I = [0 \quad 1 \quad 0 \quad 1 \quad \ldots \quad 1]. \tag{14}$$

Therefore, for each iteration *t*, the threshold to compute $\vec{I}(t)$ binary vector was calculated using the maximum value of $\vec{I}(t)$. This allows the algorithm to select at least 1 feature, then avoiding to loose its exploration ability, when compared the case in which a constant threshold is used through all iterations.

After the computation of $\vec{I}(t)$ in bit format, all fish position may now tend to $\vec{I}(t)$. In order to allow this, the position of each fish is compared to $\vec{I}(t)$. Then, one dimension of the fish that does not have the same value of $\vec{I}(t)$ is randomly chosen and is flipped. This process diminishes the hamming distance between the position of each fish and $\vec{I}(t)$. In comparison with the original algorithm, $\vec{I}(t)$ no longer represents the direction but the position resultant of the successfully individual movements. By flipping only one bit per fish, a soft and steady convergence of the algorithm is expected.

An illustrative example can be presented as follows. Suppose the values for the position and the vector *I*:

$$\begin{aligned} x(t) &= [\underline{0}\ \underline{1}\ \underline{0}\ 1\ 1] \rightarrow I = [\underline{0}\ \underline{1}\ \underline{0}\ 0\ 0] \\ x(t+1) &= [\underline{0}\ \underline{1}\ \underline{0}\ \underline{0}\ 1] \end{aligned} \tag{15}$$

In (15), the fish *x(t)* moved in the direction of $\vec{I}(t)$. One must observe that the bits with the same values are underlined. The resultant position, *x(t+1)*, is achieved by flipping one bit that was randomly chosen among the ones that present has different values. The total number of bits underlined in the new position is greater than the one in the position before the collective-instinctive movement, making the new position of the fish to be closer to $\vec{I}(t)$.

E. Collective-Volitive Movement

Similarly to the Collective-Instinctive Movement operator, the Collective-volitive operator underwent some changes. The main goal of this operator is, depending of the success of the individual movement, to contract or dilate the fish position to or from the barycentre.

The barycentre was computed in the same way as in the FSS algorithm (6). Analogously to the computation of the vector $\vec{I}(t)$, after the evaluation provided by (6), the barycentre is not obtained in a bit format. Thereby, the parameter $thres\_v$ was introduced. In the same way as in the collective-instinctive movement, an adaptative threshold was used: multiplying $thres\_v$ with the max value of barycentre.

If the overall individual movement is a success (overall weights improved in the iteration) each fish shall approximate to the barycentre. Similarly to the process in the collective-instinctive movement operator, every bit per fish is compared to the barycentre. One bit (chosen randomly among the different bits) is then flipped. By making only one flip per fish per movement, the algorithm enables a soft directing towards the barycentre. An illustrative example generated by an improvement of the overall weight (contraction) is shown in (16).

$$\begin{aligned} x(t) &= [\underline{0}\ \underline{1}\ \underline{0}\ 1\ 1] \rightarrow Bari = [\underline{0}\ \underline{1}\ \underline{0}\ 0\ 0] \\ x(t+1) &= [\underline{0}\ \underline{1}\ \underline{0}\ \underline{0}\ 1] \end{aligned} \tag{16}$$

In (16), fish changed *x(t)* randomly choosing one of the different bits when compared to *Bari* (bits not underlined in (16)) from barycentre (*bari*). This allowed the fish to approximate to the barycentre.

If the overall weights had not improved, each fish has to move to the opposite direction of the barycentre. In order to allow, we introduce here the concept of anti-barycentre. The anti-barycentre consists of a vector in which all bits are flipped when compared to the barycentre. In this situation, the process is the same as described above for the case of contraction to the barycentre, but using the anti-barycentre for the dilatation. In (17) one can observe an example of dilatation, *i.e.* the case in which the overall weights did not improve.

$$\begin{aligned} x(t) &= [0\ 1\ 0\ \underline{1}\ \underline{1}] \rightarrow Antibari = [0\ 1\ 0\ \underline{1}\ \underline{1}] \\ x(t+1) &= [\underline{1}\ 1\ 0\ \underline{1}\ \underline{1}] \end{aligned} \tag{17}$$

In (17), the fish new position *x(t+1)* is obtained by comparing each bit of the anti-barycentre of the barycentre presented in (16). One of the bits with different values (not underlined in (17)), was flipped, making the new position of the fish to be closer to the *Antibari* and consequently further to *bari* in (17). After the collective-volitive movement, a new cycle begins.

The stop criterion used was the number of iterations, and the best solution per iteration was the fish with best performance in the fitness function for that iteration.

F. Objective function

Although some of the parameters of the BFSS algorithm influence the final number of features selected (thresholds), the process of developing an objective function is critical, since it serves as guidance in search of the optimum. The objective function in (18) was used to evaluate the fitness of a certain position of the fish, based on [5]:

$$f = \alpha P + (1-\alpha)(1 - \frac{N_f}{N_t}), \tag{18}$$

where N_f represents the number of features selected, N_t the total number of features and *P* accounts for the performance of the created model. The parameter $\alpha \in [0,1]$ is the weight of the related goal: accuracy or subset size. One must remember that the two main objectives in the FS problem are: maximizing the model accuracy and minimizing the size of the feature subset.

V. RESULTS

A. Benchmark tests

To demonstrate the potential of the proposed approach, feature selection was performed in three benchmark datasets. The selected benchmark databases are listed in Table I, which are available in the UCI repository. This repository has been widely used by researchers as a primary source of machine learning databases. Furthermore, the databases in this repository have a good balance between classes and a large diversity in feature number and sample size.

TABLE I
DATABASES USED FOR VALIDATION THE BFSS ALGORITHM

Databases	Samples	Features	Classes
German (credit card)	1000	24	2
Sonar	208	60	2
WDBC	569	30	2

Three feature selection methods were applied for the sake of comparison:

- Sequential Forward Selection (SFS) search algorithm, incremental algorithm reported in [14];
- Binary Particle Swarm Optimization (PSO), a metaheuristic that can be consulted in [5];
- Binary Fish School Search (BFSS), presented in this paper.

For each database the FS was performed using the three different methods, and using the same partitions of the data in order to make a fair comparison of the results. After FS, 10-fold cross validation was applied to the best subset in each algorithm. The results of the 10-fold cross validation, for each method and with no feature selection are presented in Table II.

As expected, the results presented in Table II shows that the use of a metaheuristic (BPSO and BFSS) in the process of FS clearly increases the overall accuracy of the models and selects a lower number of features.

In what concerns the comparison of the proposed BFSS to the BPSO algorithm, Table II shows that these two algorithms have slightly better results, both in the accuracy and in the number of features selected.

The convergence of the approach presented in this paper can be seen in Fig. 1. It shows the graphical evolution of the fish with the best solution per iteration, for the FS process in the sonar database.

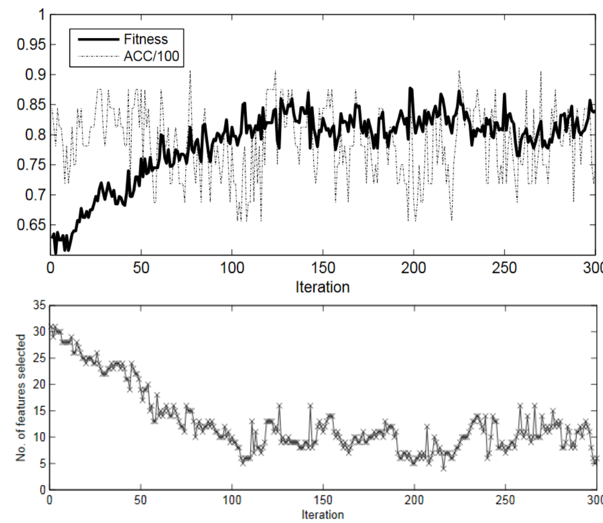

Fig. 1. Graphical evolution of the fish with the best solution per iteration using the sonar database. Fitness and ACC (above), and respective number of features selected (below).

B. Application to readmission prediction

Patient readmissions to intensive care units (ICUs) are associated with increased mortality, morbidity and costs. Current models for predicting ICU readmissions have moderate predictive value, and can utilize up to twelve variables that may be assessed at various points of the ICU inpatient stay [8,9,10].

In this paper, we used the database created in [4], derived from the MIMIC II database [14]. This database contains the time series of 22 physiologic variables that described each patient stay in an ICU.

Previous studies [4], used the arithmetic mean, the maximum, the minimum and the standard deviation of each physiologic variables in order to absorb the information and reducing the data to a constant dimension input that describes each patient stay. In this paper, in addition to these statistical measures, the Shannon entropy and the weighted average were also used, giving the possibility to withdraw more information.

TABLE II
COMPARISON OF DIFERENT METHODS FOR FEATURE SELECTION

Database	Method	Accuracy (%)			No. Features	Sensitivity			Specificity		
		Mean		Std		Mean		Std	Mean		Std
German	NO-FS	77.43	±	3.37	24	0.48	±	0.06	0.90	±	0.04
	SFS	73.72	±	4.37	7	0.45	±	0.09	0.86	±	0.04
	BPSO	74.95	±	2.84	7	0.51	±	0.11	0.85	±	0.05
	BFSS	75.74	±	3.16	6	0.52	±	0.10	0.86	±	0.03
Sonar	NO-FS	75.50	±	9.34	60	0.81	±	0.16	0.71	±	0.18
	SFS	74.91	±	7.01	11	0.71	±	0.09	0.78	±	0.14
	BPSO	78.87	±	6.49	8	0.79	±	0.08	0.79	±	0.11
	BFSS	78.88	±	8.94	9	0.80	±	0.11	0.78	±	0.12
WDBC	NO-FS	97.18	±	2.65	30	0.95	±	0.04	0.99	±	0.03
	SFS	96.30	±	2.40	5	0.93	±	0.06	0.98	±	0.03
	BPSO	96.82	±	2.32	3	0.97	±	0.05	0.97	±	0.03
	BFSS	96.51	±	3.54	3	0.98	±	0.03	0.96	±	0.05

Four different gradients for the linear distribution of the weights for the weighted mean were considered: 0.1, 0.4, 0.6 and 0.9, giving more relevance to data before the discharge of the patient. Thus, four different datasets were considered, each one with the arithmetic mean, the maximum, the minimum and the standard deviation, the Shannon entropy and the weighted average (each dataset with different gradient) of the 22 physiologic variables that describe each patient. The dimension of each data set is presented in Table III:

TABLE III
DIMENSION OF THE READMISSON DATASETS

Database	Samples	Features	Classes
Readmission	726	132	2

Each sample corresponds to a patient, and each class contains the information about the readmission or not of the patient.

Traditionally, accuracy has been used to evaluate the classifier performance. However this criterion is limited, especially for medical applications, due to several 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. Besides, a good classification of a class might be more important than classifying other classes and this cannot be assessed with accuracy. Thus, we used the sensitivity and specificity to evaluate the AUC, which was used as the main performance metrics. This choice is based on the fact that the percentage of the readmitted patients is only 12.3% against 87.7% of not readmitted patients. In this case, one of the classes is underrepresented and if the accuracy had been used the results would not be realistic.

After collecting the results of all the FS methods for the four readmission datasets, the use of different gradients for the distribution of the weights for the weighted mean proved to be not relevant. All the algorithms show almost no sensibility to the presence of different gradients of the weighted mean for the 22 physiologic variables, presenting similar results for the four datasets using the same algorithm. Thus, the results for only one dataset are presented. Table IV shows the 10-fold cross validation results after the FS process for the three FS algorithms and with no feature selection for the readmission dataset with gradient of 0.9.

The results show that BFSS both obtained a smaller number of features and introduces significant improvement in the sensitivity with a small reduction in the specificity. This is important since correct classifications for the positive cases are being made more accurately, and in this case of study the positive class has a greater importance than the negative one. Thus, the sensitivity measure is seen as a more important measure than the specificity.

The graphical evolution in Fig. 2, confirms the convergence of the BFSS algorithm formulated in this paper during the FS process.

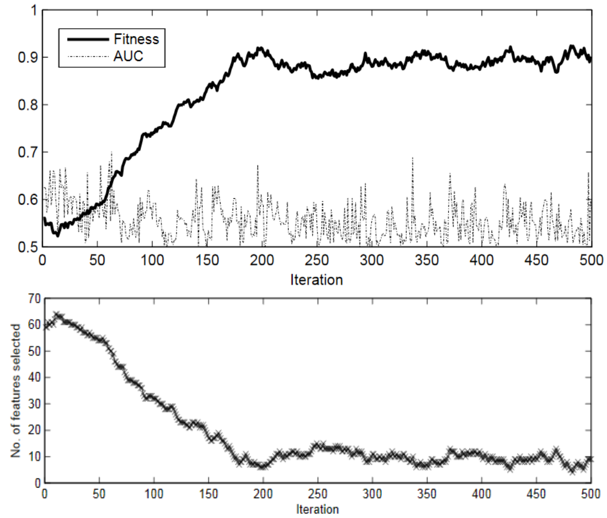

Fig. 2. Graphical evolution of the FS best fish per iteration, BFSS algorithms using the readmission dataset with gradient of 0.9 for the distribution of the weighted mean. Fitness and AUC (above), and respective number of features selected (below).

VI. Conclusions

This paper presented a new bio-inspired algorithm for feature selection, the binary fish school search (BFSS). The BFSS algorithm achieved slightly better results in comparison with the other FS methods in three benchmark databases considered in this paper. Regarding the prediction of ICU readmissions, the BFSS achieved superior results, mainly due to the significant improvement in the sensitivity, which is very important for medical applications, and the lower number of features selected. The main contribution to achieve such results was the presence of the collective-volitive operator, which had a major role to provide a higher exploration capability, allowing the algorithm to escape from local optima.

In this work, various modifications to the FSS algorithm were proposed. Nevertheless, not all the possible paths were explored. As already done to the original FSS algorithm [15], the reduction of the number of parameters and refinement of the BFSS operators are logical paths for future investigation.

TABLE IV
COMPARISON OF DIFERENT METHODS FOR FEATURE SELECTION USING THE READMISSION DATASET

Method	AUC			No. Features	Accuracy (%)			Sensitivity			Specificity		
	Mean		Std		Mean		Std	Mean		Std	Mean		Std
NO-FS	0.64	±	0.10	132	67.58	±	11.73	0.63	±	0.21	0.68	±	0.14
SFS	0.68	±	0.10	9	66.43	±	14.59	0.71	±	0.14	0.66	±	0.17
BPSO	0.67	±	0.11	8	59.13	±	10.23	0.78	±	0.11	0.57	±	0.14
BFSS	0.69	±	0.11	6	55.53	±	14.12	0.87	±	0.11	0.51	±	0.16

REFERENCES

[1] G. E. Moore. Cramming more components onto integrated circuits. Electronics Magazine, 4. 1965.

[2] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From data mining to knowledge discovery in databases. AI Magazine, 17(3), pages 37-54. 1996.

[3] H. Liu, H. Motoda. Computational Methods of Feature Selection. Chapman and Hall. 2007.

[4] A. S. Fialho, F. Cismondi, S.M. Vieira, S.R. Reti, J.M.C. Sousa, S.N. Finkelstein. Data mining using clinical physiology at discharge to predict ICU readmissions. Expert Systems with Applications 39. 2012.

[5] S.M. Vieira, L. F. Mendonca¸ G. J. Farinha, João M.C. Sousa, Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients, Applied Soft Computing, 2013

[6] Y. Saeys, I. Inza, P. Larranaga. A review of feature selection techniques in bioinformatics. Bioinformatics, 23(19), pages 2507–2517. 2007.

[7] C. J. A Bastos-Filho. , F. B. Lima-Neto, A. J. C. C. Lins, A. I. S. Nascimento, M. P. Lima. A Novel Search Algorithm based on Fish School Behavior. IEEE International Conference on Systems, Man, and Cybernetics - SMC. 2008.

[8] W. Baigelman, R. Katz, G. Geary. Patient readmission to critical care unit during the same hospitalization at a community teaching hospital. Intensive Care Med, 9(5), pages 253-256. 1983.

[9] A. L. Rosenberg, C. M. Watts. Patients readmitted to intensive care units: A systematic review of risk factors and outcomes. Chest, pages 492–502, 2000.

[10] C. G. Durbin Jr, R. F. Kopel. A case control study of patients readmitted to the intensive care unit. Critic Care Med, 21, pages 1547-1553. 1993.

[11] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, editors. Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing). Springer. August 2006.

[12] J. M. C. Sousa, U. Kaymak. Fuzzy Decision Making in Modeling and Control. World Scientific Singapore. 2002.

[13] R. Babuska. Fuzzy Modeling for Control. International Series in Intelligent Technologies. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition. April 1998.

[14] M. Saeed, C. Lieu, R. Mark. MIMIC II. a massive temporal icu database to support research in intelligence patient monitoring. Computers in Cardiology, 29, pages 641-644. 2002.

[15] A. G. K. Janecek, Y. Tan. Feeding the fish - Weight update strategies for the fish school search algorithm. ICSI'11 Proceedings of the Second international conference on Advances in swarm intelligence, Volume Part II. 2011.