Journal of Biomedical Informatics 49 (2014) 61–72

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

A medical diagnostic tool based on radial basis function classifiers and evolutionary simulated annealing

Alex Alexandridis *, Eva Chondrodima

Department of Electronic Engineering, Technological Educational Institute of Athens, Agiou Spiridonos, Aigaleo 12210, Greece

ARTICLE INFO

Article history:
Received 18 June 2013
Accepted 13 March 2014
Available online 21 March 2014

Keywords:
Decision support systems
Evolutionary computation
Medical diagnosis
Neural networks
Radial basis function
Simulated annealing

ABSTRACT

Objective: The profusion of data accumulating in the form of medical records could be of great help for developing medical decision support systems. The objective of this paper is to present a methodology for designing data-driven medical diagnostic tools, based on neural network classifiers.

Methods: The proposed approach adopts the radial basis function (RBF) neural network architecture and the non-symmetric fuzzy means (NSFM) training algorithm, which presents certain advantages including better approximation capabilities and shorter computational times. The novelty in this work consists of adapting the NSFM algorithm to train RBF classifiers, and suitably tailoring the evolutionary simulated annealing (ESA) technique to optimize the produced RBF models. The integration of ESA is critical as it helps the optimization procedure to escape from local minima, which could arise from the application of the traditional simulated annealing algorithm, and thus discover improved solutions. The resulting method is evaluated in nine different medical benchmark datasets, where the common objective is to train a suitable classifier. The evaluation includes a comparison with two different schemes for training classifiers, including a standard RBF training technique and support vector machines (SVMs). Accuracy% and the Matthews Correlation Coefficient (MCC) are used for comparing the performance of the three classifiers.

Results: Results show that the use of ESA helps to greatly improve the performance of the NSFM algorithm and provide satisfactory classification accuracy. In almost all benchmark datasets, the best solution found by the ESA-NSFM algorithm outperforms the results produced by the SFM algorithm and SVMs, considering either the accuracy% or the MCC criterion. Furthermore, in the majority of datasets, the average solution of the ESA-NSFM population is statistically significantly higher in terms of accuracy% and MCC at the 95% confidence level, compared to the global optimum solution that its rivals could achieve. As far as computational times are concerned, the proposed approach was found to be faster compared to SVMs.

Conclusions: The results of this study suggest that the ESA-NSFM algorithm can form the basis of a generic method for knowledge extraction from data originating from different kinds of medical records. Testing the proposed approach on a number of benchmark datasets, indicates that it provides increased diagnostic accuracy in comparison with two different classifier training methods.

1. Introduction

Medical diagnosis refers to the act of identifying a disease from its signs and symptoms, or, in other words, classifying input symptoms into some known disease category. As medical diagnosis often involves a degree of uncertainty, physicians could reap the benefits of appropriate decision support systems, able to help them assess medical situations. The profusion of data accumulating in the form of medical records could be of great help for designing such systems; however, unlocking the knowledge contained in historical data is associated with a number of difficulties. The relationship between a diagnosis and the associated symptoms or clinical findings is usually rather complex and nonlinear in nature and the mechanisms that govern the operation of the human body are quite often poorly understood; these facts impede the formulation of explicit laws that could be fitted to the available data, and thus, make the extraction of knowledge from medical records rather cumbersome.

Though the aforementioned difficulties hinder the task, recent advances in machine learning and data mining methodologies are catalyzing the development of medical decision support

---

* Corresponding author.
E-mail address: alexx@teiath.gr (A. Alexandridis).

http://dx.doi.org/10.1016/j.jbi.2014.03.008

systems. In this context, a variety of techniques have been used for building medical diagnostic tools [1], a significant portion of which involve a family of algorithms known as neural networks (NNs). NNs are a set of powerful mathematical tools [2] that simulate the way that the human brain deals with information and the procedure of learning. NNs have the ability to identify and learn highly complex and nonlinear relationships from input–output data only, without the use of first principle equations describing the system. This is accomplished by storing information in computational nodes called neurons. The development of an NN is based on a set of input–output examples, which the network learns during the training phase. In this stage, the network parameters associated with the neurons and/or the interconnection links are determined using an optimization procedure, which minimizes the errors between the true outputs and the network predictions over a set of training examples. A successfully trained NN model should be capable of producing accurate estimations given a new set of input data.

A number of publications exploiting the use of NNs for building medical diagnostic tools have appeared during the last decades. A method for using variations in electrical impedance over frequency to distinguish basal cell carcinoma from benign skin lesions based on NN classifiers is introduced in [3]. The authors in [4] present a neural network-based biomarker association information extraction approach for cancer classification. A system called GEMS (gene expression model selector) has been developed for the automated development and evaluation of high quality cancer diagnostic models, including NN methodologies [5]. Several NN-based classifier approaches have been proposed for diagnosing heart diseases [6]. Training NN classifiers for medical decision making is studied in [7], focusing on the effects of imbalanced datasets on classification performance. The important issue of feature selection in building support vector machine (SVM) classifiers for biomedical diagnostic datasets has been studied in [8]. The use of NNs for medical data classification tasks is reviewed in [9]. Other similar machine learning approaches employed for building medical diagnostic tools include Bayesian methods [10–12], fuzzy logic [13], decision trees [14] and multivariable statistical techniques [15].

Depending on the way the nodes of the network are interconnected and the calculations that each node performs, NNs are categorized to a variety of architectures. Radial basis function (RBF) networks [16] constitute a special network architecture that presents some remarkable advantages over other NN types including: (a) better approximation capabilities when performing interpolation, i.e. providing predictions in-between the training data points, (b) simpler network structures comprising a single hidden layer and (c) faster learning algorithms which are usually split into two stages [17,18]. Due to these advantages, RBF networks have been used extensively for modeling complex systems, with many successful applications in developing medical diagnostic tools [19–21]. Especially in this area, improving the accuracy of the network predictions is vital in achieving high performance diagnosis. Unfortunately, calculating optimal values for the parameters of an RBF network – as with other NN architectures – is a rather cumbersome task. When viewing the NN training procedure as an optimization problem, one realizes that the objective function usually presents some rather unwelcome properties including: (a) multimodality, possibly trapping the algorithm in a non-satisfactory local optimum, (b) non-differentiability, which impedes the use of derivative-based methods and (c) high levels of noise, typically present in the training data [17]. As these characteristics make the use of standard optimization methods inefficient, it is no surprise that interest is shifting to optimizing the RBF training procedure through the use of alternative approaches, like evolutionary-based computation techniques.

Evolutionary computation [22] refers to a family of probabilistic optimization techniques that mimic processes from natural evolution, where the main concept is the survival of the fittest. Evolutionary algorithms (EAs) form a class of generic purpose search methods, which strike a remarkable balance between exploration and exploitation—two apparently conflicting objectives in any optimization technique. There are certain differences between EAs and conventional optimization methods: EAs manipulate coded versions of the problem parameters instead of the parameters themselves; they do not use any auxiliary information about the objective function, such as derivatives; they operate in parallel on a population of solutions instead of manipulating a single solution. Hence, EAs are considered to be efficient and robust optimization methods that are able to escape from local optima. These properties have made EAs very popular in various and diverse scientific fields, including applications of medical interest [23–26]. Not surprisingly, EAs have also been used for RBF network optimization. In [27], a new algorithm is introduced for developing dynamic RBF models based on genetic algorithms. Particle swarm optimization (PSO) is another methodology drawn from the arsenal of EA methods that has been exploited in this area; a novel algorithm for training RBF regression models based on PSO is presented in [17].

The contribution of this paper lies in presenting a new methodology for building and optimizing RBF-based diagnostic tools with increased classification capabilities. The proposed approach uses the non-symmetric fuzzy means (NSFM) algorithm [28], which in this work is modified properly to train RBF classifiers. Furthermore, the evolutionary simulated annealing (ESA) [29] technique is suitably tailored and incorporated to the proposed scheme to optimize the fuzzy partition of the produced classifiers, with respect to the classifying accuracy. The resulting classification method is applied on medical diagnosis-related datasets.

The rest of this paper is organized as follows: In the next section, we describe the applied techniques, including a brief introduction to the RBF architecture and the NSFM algorithm, full presentation of the proposed approach for training RBF classifiers, and a description of the methods used for evaluation and testing. Experimental results are presented in Section 3, followed by discussion in Section 4. The paper concludes by outlining the advantages of the proposed approach and setting directions for future work.

2. Methods

2.1. Radial basis function network classifiers

Fig. 1 depicts the typical structure of an RBF classifier, able to discern between M different classes. The input layer distributes the N input variables to the L nodes of the hidden layer. Each node in the hidden layer is associated with a center, equal in dimension with the number of input variables. Thus, the hidden layer

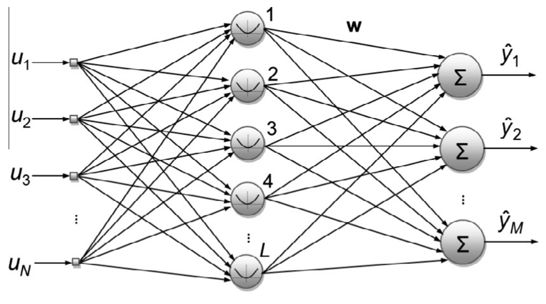

Fig. 1. Typical structure of an RBF classifier.

performs a nonlinear transformation and maps the input space onto a new higher dimensional space. The activity $\mu_l(\mathbf{u}(k))$ of the lth node is the Euclidean norm of the difference between the kth input vector and the node center and is given by:

$$\mu_l(\mathbf{u}(k)) = \|\mathbf{u}(k) - \hat{\mathbf{u}}_l\| = \sqrt{\sum_{i=1}^{N}(u_i(k) - \hat{u}_{i,l})^2}, \quad k = 1,\ldots,K \tag{1}$$

where K is the total number of data, $\mathbf{u}^T(k) = [u_1(k), u_2(k), \ldots, u_N(k)]$ is the input vector and $\hat{\mathbf{u}}_l^T = [\hat{u}_{1,l}, \hat{u}_{2,l}, \ldots, \hat{u}_{N,l}]$ is the center of the lth node.

The activation function for each node is a radially symmetric function. This work employs the thin plate spline function:

$$g(\mu) = \mu^2\log(\mu) \tag{2}$$

Thus, the hidden node responses $\mathbf{z}(k)$ become:

$$\mathbf{z}(k) = [g(\mu_1(\mathbf{u}(k))), g(\mu_2(\mathbf{u}(k))), \ldots, g(\mu_L(\mathbf{u}(k)))] \tag{3}$$

The output layer of the RBF classifier consists of M summation units, where M is the number of possible output classes. The numerical output $\hat{y}_m(k)$ for each summation unit is produced by a linear combination of the hidden node responses:

$$\hat{y}_m(k) = \mathbf{z}(k)\cdot\mathbf{w}_m = \sum_{l=1}^{L} w_{l,m}\, g(\mu_l(\mathbf{u}(k))) \tag{4}$$

where $\mathbf{w}_m = [w_{1,m}, w_{2,m}, \ldots, w_{L,m}]^T$ is a vector containing the synaptic weights corresponding to output m. The classifier prediction for the output class $\mathscr{C}(k)$ corresponds to the summation unit triggering the maximum numerical output [2]:

$$\mathscr{C}(k) = \arg\max_m \hat{y}_m(k), \quad m = 1, 2, \ldots, M \tag{5}$$

After fixing the RBF centers and nonlinearities in the hidden layer, the synaptic weights are typically calculated using linear regression of the hidden layer outputs to the real measured outputs (target values). The regression problem can be trivially solved using linear least squares in matrix form:

$$\mathbf{W} = (\mathbf{Z}^T\cdot\mathbf{Z})^{-1}\cdot\mathbf{Z}^T\cdot\mathbf{Y} \tag{6}$$

where $\mathbf{Z} = [\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(K)]^T$ is a matrix containing the hidden layer responses for all input vectors, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M]$ is a matrix containing the synaptic weights for the output layer units and $\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(K)]^T$ is a matrix containing the target vectors. Each target vector $\mathbf{y}(k)$ carries the information of which class the k-th input vector actual belongs to. When a target vector $\mathbf{y}(k)$ belongs to output class m, then all its elements $y_i(k)$ are set equal to zero, except from the m-th element, which is set equal to 1:

$$y_i(k) = \begin{cases} 1, & \text{if } i = m \\ 0, & \text{if } i \neq m \end{cases}, \quad i = 1, 2, \ldots, M \tag{7}$$

2.1.1. Non-symmetric fuzzy means for RBF classifier training

The FM algorithm [30] is a method for selecting the hidden node centers of an RBF network. It has several advantages over the standard approaches like the k-means algorithm [31], including automatic determination of the size of the network, i.e. the number of hidden nodes, and faster computational times. The FM algorithm has been used successfully in a number of applications [32,33]. A variant of this algorithm, utilizing a non-symmetric fuzzy partition of the input space (NSFM algorithm) has been recently proposed [28]. What follows is a brief description of the main concepts behind the NSFM algorithm.

Consider a system with N normalized input variables u_i, where $i = 1,\ldots,N$. A non-symmetric fuzzy partition of the input space, implies that the domain of each input variable i is partitioned into a different number c_i of one-dimensional triangular fuzzy sets. Each fuzzy set can be written as:

$$A_{i,j} = \{a_{i,j}, \delta\alpha_i\}, \quad i = 1,\ldots,N, \quad j = 1,\ldots,c_i \tag{8}$$

where $a_{i,j}$ is the center element of fuzzy set $A_{i,j}$ and $\delta\,\alpha_i$ is half of the respective width. It should be noted that the widths are different for each input direction, depending on the selected number of fuzzy sets c_i.

This partitioning technique creates a total of C multi-dimensional fuzzy subspaces $\mathbf{A}^l$, where $l = 1,\ldots,C$:

$$C = \prod_{i=1}^{N} c_i \tag{9}$$

Each multi-dimensional fuzzy subspace is generated by combining N one-dimensional fuzzy sets, one for each input direction. Thus, the fuzzy subspaces can be defined using a center vector $\boldsymbol{\alpha}^l$ and a side vector $\delta\boldsymbol{\alpha}$:

$$\mathbf{A}^l = \{\boldsymbol{\alpha}^l, \delta\boldsymbol{\alpha}\} = \{[a^l_{1,j_1}, a^l_{2,j_2}, \ldots, a^l_{N,j_N}], [\delta\alpha_1, \delta\alpha_2, \ldots\delta\alpha_N]\}, \quad l = 1,\ldots,C \tag{10}$$

where a^l_{i,j_i} is the center element of the one-dimensional fuzzy set A_{i,j_i} that has been assigned to input i. The produced fuzzy subspaces form a grid where each node is candidate for becoming an RBF center. An example of the non-symmetric partition is displayed in Fig. 2 using a two-dimensional case for visualization purposes. The domain of input variable u_1 is partitioned into 7 fuzzy sets, whereas the domain of u_2 is partitioned into 5 fuzzy sets, thus defining a total of 35 fuzzy subspaces. The grid forms 24 equal rectangles in the input space and the two edges of each rectangle are twice the widths of the respective fuzzy sets $\delta\alpha_1$, $\delta\alpha_2$.

The objective of the FM algorithm is to assemble the RBF network hidden layer by selecting only a small subset of the fuzzy subspaces. The number of the selected fuzzy subspaces should be kept low so as to produce a parsimonious model, but at the same time the produced RBF centers should cover sufficiently the available input data. The NSFM algorithm performs the selection based on the multidimensional membership function $\mu_{\mathbf{A}^l}(\mathbf{u}(k))$ of an input vector $\mathbf{u}(k)$ to a fuzzy subspace $\mathbf{A}^l$ [34]:

$$\mu_{\mathbf{A}^l}(\mathbf{u}(k)) = \begin{cases} 1 - d_r^l(\mathbf{u}(k)), & \text{if } d_r^l(\mathbf{u}(k)) \leqslant 1 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $d_r^l(\mathbf{u}(k))$ is a function of the distance between the fuzzy subspace $\mathbf{A}^l$ and the input data vector $\mathbf{u}(k)$. The function $d_r^l(\mathbf{u}(k))$

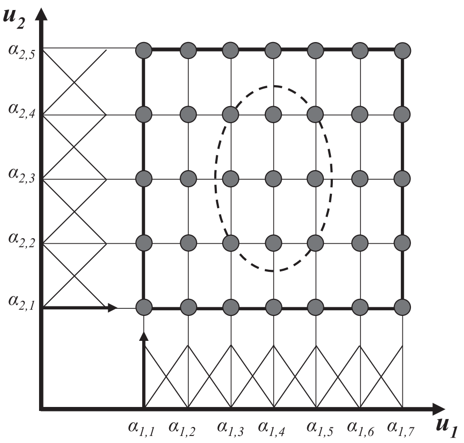

Fig. 2. Non-symmetric partition on a two-dimensional input space with an elliptical membership function.

defines a hyper-ellipse on the input space, described by the following equation:

$$d_r^l(\mathbf{u}(k)) = \sqrt{\sum_{i=1}^{N}\left(\left(a^l_{i,j_i} - u_i(k)\right)^2 \Big/ N(\delta a_i)^2\right)} \tag{12}$$

Fig. 2 depicts the ellipse that is formulated in the two-dimensional case.

Having defined the membership function, the algorithm proceeds with finding the subset of fuzzy subspaces that assign a nonzero multidimensional degree to all input training vectors. This is accomplished using a non-iterative algorithm which requires only one pass of the input data, thus rendering the center calculation procedure extremely fast, even in the presence of a large database of input examples. Algorithm 1, presents an overview of the NSFM algorithm.

Algorithm 1 – NSFM Algorithm

```
Input: {U_train, Y_train}: Training Dataset,
       $\mathbf{c} = [c_1, c_2, \ldots, c_N]$: Number of fuzzy sets
       for partitioning each input dimension
Output: L: Number of selected RBF centers,
       $\widehat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_L]^T$: Selected RBF center
       locations
Take the first data point: $k \leftarrow 1$
Begin calculations for the first RBF center: $L \leftarrow 1$
For $i = 1{:}N$ Do:
  Calculate the fuzzy set with maximum membership in
  each dimension $i$:
  $A_i^1 = \{a_i^1, \delta a_i\} \leftarrow \max_{1\leqslant j\leqslant s_i}[\mu_{A_{i,j}}(u_i(1))]$
End For
Generate the first RBF center $\hat{\mathbf{u}}_1$: $\hat{\mathbf{u}}_1 = [a_1^1, a_2^1, \ldots, a_N^1]$
For $k = 2{:}K$ Do:
  If data point $k$ lies outside the hyper-ellipses defined by
  the already selected centers:
  $\min_{1\leqslant l\leqslant L}[d_r^l(\mathbf{u}(k))] > 1$, where $d_r^l$ is calculated by (12)
    Add a new RBF center: $L \leftarrow L + 1$
    For $i = 1{:}N$ Do:
      Calculate the fuzzy set with maximum membership
      in each dimension $i$:
      $A_i^L = \{a_i^L, \delta a_i\} \leftarrow \max_{1\leqslant j\leqslant s_i}[\mu_{A_{i,j}}(u_i(k))]$
    End For
    Generate the $L$-th RBF center $\hat{\mathbf{u}}_L$: $\hat{\mathbf{u}}_L = [a_1^L, a_2^L, \ldots, a_N^L]$
  End If
End for
```

Though the NSFM algorithm can lead to significant improvements in both accuracy and parsimony of the produced model, the non-symmetric partition necessitates the determination of an increased number of operational parameters, in an optimal way. For problems involving a very low number of samples and input variables, the optimal number of fuzzy sets for each variable could be calculated using an exhaustive search procedure, where all possible combinations of non-symmetric partitions are tested. Unfortunately, as the number of possible combinations increases explosively with the dimension of the input space, the excessive computational burden imposed by even slightly larger problems, makes the use of exhaustive search practically infeasible in this case. In this work, we introduce a novel method for optimizing the non-symmetric partition of the input space through the use of evolutionary simulated annealing (ESA), thus leading to a new integrated procedure for developing high accuracy RBF-based classifiers.

2.2. RBF classifier optimization based on evolutionary simulated annealing

2.2.1. Simulated annealing

Simulated Annealing (SA) is a stochastic optimization technique based on the work of Metropolis [35]. The SA method parallelizes the search of the minimum energy state, which occurs during the annealing process applied in metallurgy, to the search for the minimum of a mathematical function, through an iterative procedure. The original SA algorithm operates on a single solution which is usually expressed as a vector of integer or real values. The algorithm starts with a randomly selected solution of the problem. In each iteration, a new solution is generated by applying a random change to the previous one. The new solution is evaluated using the objective function and can be accepted or deleted depending on the difference with the previous evaluation. During the first iterations even a worst solution compared to the previous one can probably be accepted; this corresponds to the random movement of molecules in high temperatures. Such degradations in the solution quality can be proven essential to avoid paths leading to local minima. As the algorithm progresses, the probability p_B that a worst solution will be accepted gradually becomes smaller, imitating the reduced mobility of the molecules as the temperature decreases. In fact, this probability is given by the Boltzmann–Gibbs distribution:

$$p_B = \begin{cases} 1, & \text{if } f(\mathbf{x}_i) < f(\mathbf{x}_{i-1}) \\ \exp\left(-\dfrac{f(\mathbf{x}_i) - f(\mathbf{x}_{i-1})}{kT}\right), & \text{otherwise} \end{cases} \tag{13}$$

where T is the temperature at each iteration, k is the Boltzmann constant, f is the objective function and $\mathbf{x}_i$, $\mathbf{x}_{i-1}$ represent two consecutive solutions during the annealing procedure. An annealing schedule is applied, starting from an initial high temperature $T_{\max}$ and gradually decreasing until reaching the lower temperature $T_{\min}$. The most common annealing schedule is exponential cooling, which lowers the temperature in successive discrete steps t according to the following equation:

$$T(t+1) = aT(t), \quad 0 < a < 1 \tag{14}$$

where a is the exponential cooling constant

2.2.2. Evolutionary simulated annealing

Unlike classic evolutionary techniques, e.g. genetic algorithms (GAs) or particle swarm optimization (PSO), the original SA algorithm operates on a single solution, instead of using a population of them. This is beneficial with respect to the computational burden of the algorithm, but decreases the possibility of finding a global minimum. Another difference of SA compared to the aforementioned techniques, is that though detrimental moves are allowed during the first iterations of SA, later-on the method behaves almost like a hill-climbing algorithm, accepting only an improved solution. This makes it harder to escape from local minima during the last stages of the algorithm.

To obviate these drawbacks, SA was hybridized with evolutionary techniques, thus creating the evolutionary simulated annealing algorithm (ESA) [29]. ESA starts with a population of randomly initialized solutions, coded as chromosomes. In contrast with genetic algorithms though, there are no genetic operators like crossover or mutation involved. The only operator acting on the chromosomes is in fact the SA operator. This implies that each chromosome in the current generation is used as an initial solution for performing the original SA algorithm. The terminal solution produced by SA is then compared to the initial solution and the best of the two survives for the next generation. The algorithm is repeated until a stopping condition is met. Criteria that could be used as stopping conditions for ESA include:

- Maximum number of generations is reached.
- No improvement is accomplished in the objective function over a number of successive generations.
- An acceptable solution is found.

It should be noted that the two first criteria refer to the generations corresponding to the evolutionary procedure. The iterations performed by SA are kept to a constant value selected by the user.

2.2.3. Encoding fuzzy partitioning for ESA

The ESA algorithm can be used as a search method for calculating the optimum non-symmetric partitioning given as input to the NSFM algorithm. In this case, the chromosomes that participate in the evolutionary procedure represent different ways of partitioning the input space. Applying this formulation, each chromosome $\mathbf{c}_p(q)$, corresponds to the number of fuzzy sets in each dimension:

$$\mathbf{c}_p(q) = [c_1(q), c_2(q), \ldots, c_N(q)]^T, \quad p = 1, 2, \ldots, P, \quad q = 1, 2, \ldots, Q \qquad (15)$$

where P and Q denote the size of the population and the total number of generations, respectively. In each generation, all chromosomes in the population undergo the SA operator. The random change which is applied to the chromosomes in each SA iteration is of the form:

$$\mathbf{c}_p(q+1) = \text{round}\{\mathbf{c}_p(q) + \mathbf{r}(q)\} \qquad (16)$$

where $\mathbf{r}(q)$ is a vector containing N random numbers drawn from a Gaussian distribution $\sim N(0, \sigma)$. The parameter σ defines the standard deviation of the random applied changes to the chromosomes. The round operator is used to round the chromosomes to the nearest integer, as the number of fuzzy sets in each dimension can receive only integer values.

2.2.4. RBF classifier training using NSFM and ESA

Optimization of the non-symmetric fuzzy partitioning of the input space using ESA, forms the basis for an integrated RBF classifier training methodology. The algorithm starts by randomly dividing the available data in three datasets, namely the training, validation and testing datasets. The classifier is trained on the training dataset, while the validation dataset is used for model selection. Alternatively, model selection can be performed using a k-fold cross-validation procedure, albeit at the cost of increasing the computational burden of the algorithm. As k-fold cross-validation is by itself a computationally expensive procedure, incorporating it to an evolutionary cycle would mean repeating the RBF network training procedure many times for each chromosome and each iteration. In any case, the existence of a third independent dataset for testing the produced RBF classifier is crucial, since the optimization procedure could result in overfitting the model to the data used for validation. During the particle initialization stage, P different partitions of the input space are selected randomly, and coded as chromosomes. Each chromosome then gives birth to an RBF classifier by applying the NSFM algorithm. The latter, calculates the number and coordinates of the RBF centers based on Eq. (11). After fixing the RBF centers, the synaptic weights can be trivially calculated using Eq. (6). Based on the emerging RBF networks, a fitness value is calculated for each chromosome, by applying an error-related criterion. The proposed algorithm adopts the accuracy% criterion, defined as follows:

$$\text{Accuracy\%} = \frac{\text{Total number of correctly classified data}}{\text{Total number of data}} \cdot 100 \qquad (17)$$

Alternative error functions can also be used. The next step is to create a new generation of chromosomes by applying the SA operator. SA starts with the initial chromosome and performs random changes using Eq. (16). The resulting solution can be accepted or rejected with a probability calculated by the Boltzmann–Gibbs distribution (Eq. (13)). As soon as the temperature reaches its minimum value, the SA algorithm returns the produced solution, which is compared to the initial one. The best of the two solutions is passed onto the next generation. The algorithm then returns at the RBF network generation stage, until a stopping condition has been met. The stopping condition in this case is satisfied when a maximum number of generations is reached. A schematic overview of the ESA-NSFM method is given in Fig. 3, whereas the algorithm structure is presented in Algorithm 2.

Algorithm 2 – ESA-NSFM Algorithm

```
Input:  ${\mathbf{U}_{train}, \mathbf{Y}_{train}}$: Training Dataset,
        ${\mathbf{U}_{val}, \mathbf{Y}_{val}}$: Validation Dataset,
        $s_{min}, s_{max}, \sigma$: Minimum and maximum number of
        fuzzy sets, and standard deviation of the random
        applied changes to the number of fuzzy sets
        $P$: Population,
        $T_{min}, T_{max}, k, a$: ESA operational parameters
        (Minimum and maximum temperatures, Boltzmann
        constant, and cooling constant, respectively)
        $Q$: Maximum number of generations
Output: $L_f$: Number of selected RBF centers,
        $\hat{\mathbf{U}}_f$: Selected RBF center locations,
        $\mathbf{W}_f$: Selected RBF synaptic weights
For $p = 1:P$ Do:
        Initialize chromosome coordinates at random
        integer
        numbers between $s_{min}$ and $s_{max}$:
        $\mathbf{c}_p(0) \leftarrow [\underbrace{\text{rand}, \ \text{rand}, \ \ldots, \ \text{rand}}_{N}]$
End For
For $q = 1:Q$ Do:
    For $p = 1:P$ Do:
        Pass $\mathbf{c}_p(q)$ and the training dataset ${\mathbf{U}_{train}, \mathbf{Y}_{train}}$ to
        algorithm 1, in order to calculate the total number of
        RBF centers $L_i(t)$ and their locations $\hat{\mathbf{u}}_p(t)$
        Use (5) to calculate the synaptic weights $\mathbf{W}_p(q)$
        Calculate fitness function $f(\mathbf{c}_p(q))$ on the validation
        dataset ${\mathbf{U}_{val}, \mathbf{Y}_{val}}$
        Initialize the solution $\mathbf{c}'_p(0)$ for the annealing process
        $\mathbf{c}'_p(0) \leftarrow \mathbf{c}_p(q)$ and the annealing iterations $t \leftarrow 0$
        While the temperature $T$ is greater than $T_{min}$ Do:
            Proceed to the next annealing iteration $t \leftarrow t + 1$
            Calculate a new solution $\mathbf{c}'_p(t)$ using (16)
            If the old solution $\mathbf{c}'_p(t-1)$ is better than the new
            solution $\mathbf{c}'_p(t)$
                Calculate the Boltzmann probability $p_B$ using (13)
                If a random number is greater than $p_B$
                    Substitute the old solution with the new one
                    $\mathbf{c}'_p(t) \leftarrow \mathbf{c}'_p(t-1)$
                End If
            End If
            Update temperature using (14)
        End While
        If SA terminal solution $\mathbf{c}'_p(t)$ is better than the current
        generation solution $\mathbf{c}_p(q)$
            Substitute the current generation solution with SA
            terminal solution $\mathbf{c}_p(q) \leftarrow \mathbf{c}'_p(t)$
        End If
    End For
End For
```

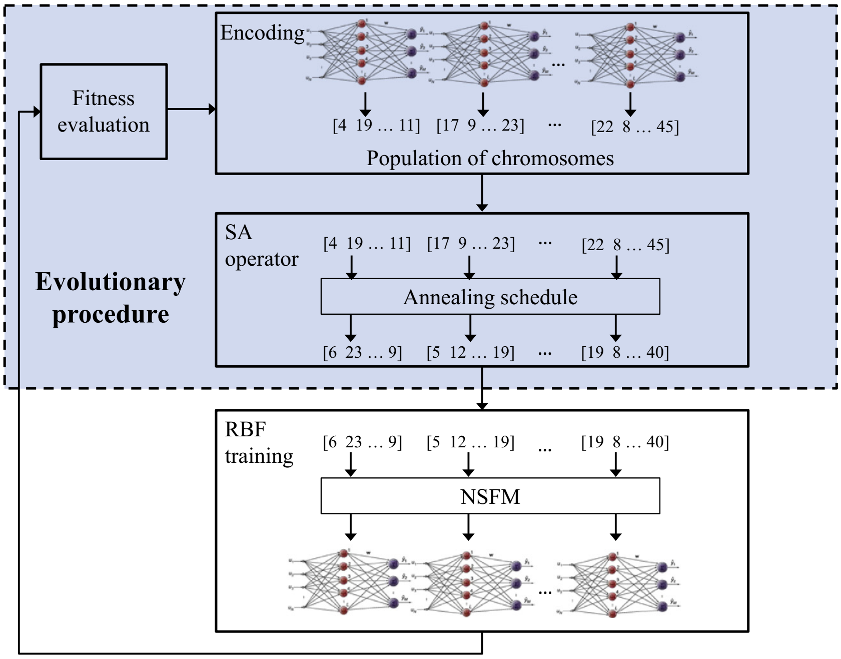

Fig. 3. A schematic overview of the ESA-NSFM algorithm. (For interpretation to colours in this figure, the reader is referred to the web version of this paper.)

2.3. Method evaluation

To demonstrate the effectiveness of the proposed method as a diagnostic tool in medicine, the algorithm was applied on several datasets of medical interest, downloaded from the UCI Machine Learning Repository [36]. The common objective in all datasets is to predict the output class (category) based on a number of input variables (attributes), which are continuous or ordinal. Table 1 presents an overview of the conducted experiments, depicting the numbers of input variables, total data points, output classes, and positive cases (only for datasets with two output classes), for the tested datasets. Some datasets included examples with missing values, which, for the purposes of this study, were discarded. A brief description for each dataset is given in Appendix A, whereas additional information can be found in [36].

For each dataset, data were split randomly to training, validation and testing subsets. In general, there is no consensus in the relative literature regarding the amount of data that should be allocated to each subset. In this work, a 50%–25%–25% ratio (Training–Validation–Testing) was adopted, taking special care so as to maintain the same splitting ratio not only for the whole dataset, but also for each class individually.

As far as the operational parameter selection is concerned, it was based on theoretical aspects, as well as experimentation. Regarding the NSFM part-related parameters, the minimum and maximum number of fuzzy sets define the lower and upper bounds of the search space, respectively. Previous experiments with both symmetric and non-symmetric versions of the FM algorithm have shown that partitions outside these bounds are not likely to result to successful models. The standard deviation of applied changes σ actually defines the expected magnitude of change that will be applied to the fuzzy sets, in each iteration of the SA procedure. This parameter can be used to control the exploitation–exploration trade-off in the ESA-NSFM algorithm. Small values of σ will result to a thorough local search but poor exploration of the search space, while large values will have the opposite effect. Obviously, the choice for this parameter is closely related to the size of the search space. The latter depends on the values for $s_{\min}$ and $s_{\max}$, which define its bounds, and on the dimensionality of the input space, which coincides with the dimensionality of the search space. As the dimensionality of the input space increases, the parameter σ should be also allowed to increase, in order to guarantee that the search space will be sufficiently explored within reasonable simulation times. In this work, two different values for σ are used, depending on the input space dimensionality of the problem. The values were selected after experimental testing, as they were found to provide a good balance between exploration and exploitation.

Table 1
Dataset overview.

Dataset	No. of inputs	No. of examples	No. of classes	No. of positive cases[a]
Cardiotocography FHR	21	2126	10	–
Cardiotocography NSP	21	2126	3	–
EEG Eye State	14	14,980	2	6723
Pima Indians Diabetes	8	768	2	268
Thyroid ANN	6	7200	3	–
Vertebral Column – 2 Classes	6	310	2	210
Vertebral Column – 3 Classes	6	310	3	–
Wisconsin Diagnostic Breast Cancer (WDBC)	30	569	2	212
Wisconsin Original Breast Cancer (WOBC)	10	699	2	241

[a] This column refers only to datasets with two output classes.

As far as the ESA parameters are concerned, the size of the population and the maximum number of generations are parameters that significantly affect the computational load of the algorithm. Likewise, the minimum and maximum temperatures, together with the cooling constant, actually determine the total number of iterations that will be performed by the SA operator in each generation. In this study their respective values were chosen so as to perform 500 iterations. Experiments showed that increasing these three parameters (population, number of generations and number of SA iterations), could result to improved classifier accuracy, albeit at the cost of additional computational burden. For the purposes of this study we have kept these parameters at small values, as our experiments showed that satisfactory results can be obtained in relatively fast computational times. Finally, the Boltzmann constant controls whether a worse solution compared to the current one will be accepted or rejected inside the SA loop. Choice for this parameter also affects the exploitation–exploration balance, although its effect is somewhat muffled by the evolutionary cycle which repeats the SA procedure using as an initial solution, the terminal solution of the previous SA run. Thus, a more uniform distribution of random moves is provided throughout the ESA procedure. This was also verified experimentally, as the results showed that there was small sensitivity when changing this parameter around the value of 0.05, while the results deteriorated only for significantly lower or higher values.

Table 2 presents the selected operational parameters for the ESA-NSFM algorithm. It should be noted that the results presented in the next sub-section were all derived using the same operational parameter values in all datasets, with the exception of σ. The rationale behind this choice was to highlight the generic character of the method, which can be applied in medical records of diverse nature without a tedious tuning procedure. Nevertheless, further improvements could be achieved by customizing the parameters individually for each dataset along the lines discussed in the previous paragraphs.

For comparison purposes, two additional types of classifier were also tested, namely RBF classifiers trained with the symmetric version of the FM algorithm [37] and standard SVM classifiers [38] with Gaussian kernel function [39]. In all cases, the classifier parameters were determined using the training dataset and then model selection was performed using the validation dataset, with respect to the accuracy% criterion. In the case of RBF classifiers trained with SFM, model selection was controlled by only one parameter, namely the number of fuzzy sets c for partitioning each input variable. Selection of c was made after testing all partitions from 4 to 50 fuzzy sets. The SVM classifiers with Gaussian kernel function were optimized with respect to the width of the Gaussian kernels σ and the penalty factor C. An exhaustive search procedure based on a grid search method [40] was applied, testing different combinations of σ and C. In the case of datasets with more than two output classes, a one-against-all technique was applied, as recommended by Vapnik [39].

Table 2
Operational parameters for the ESA-NSFM Algorithm.

Algorithm	Parameter	Symbol	Value
ESA	Population	P	12
	Maximum number of generations	G	20
	Maximum temperature	$T_{\max}$	100
	Minimum temperature	$T_{\min}$	1
	Cooling constant	a	0.991
	Boltzmann constant	k	0.05
NSFM	Minimum number of fuzzy sets	$s_{\min}$	4
	Maximum number of fuzzy sets	$s_{\max}$	50
	Standard deviation of random applied changes[a]	σ	S:8, L:12

[a] Value depending on input space dimensionality: Small problems (S): 1–10 input variables, Large problems (L): more than 10 input variables.

A standard method for comparing the performance of different classifiers, which is also used in this study, is the accuracy% criterion. This metric comes as a natural choice, as it assesses the total success rate of each classifier; however there are cases where additional information from the confusion matrix is needed to fully characterize the classifier performance. A common example is the classification of data with imbalanced class distribution, where even a total failure in predicting a rare class, would have only a small impact on the total accuracy%. To overcome these weaknesses, the Matthews correlation coefficient (MCC) was introduced [41], summarizing the confusion matrix of a binary classification task into a single value. As in this work datasets with more than two classes are employed, we adopt a generalization of MCC for the multi-class case [42], which has been reported as a good generic comparison basis, among different classifier performance metrics [43]. MCC is defined as:

$$\mathrm{MCC} = \frac{\sum_{k,l,m=1}^{M} C_{kk}C_{ml} - C_{lk}C_{km}}{\sqrt{\sum_{k=1}^{M}\left(\sum_{l=1}^{M} C_{lk}\right)\left(\sum_{\substack{f,g=1\\ f\neq k}}^{M} C_{gf}\right)}\sqrt{\sum_{k=1}^{M}\left(\sum_{l=1}^{M} C_{kl}\right)\left(\sum_{\substack{f,g=1\\ f\neq k}}^{M} C_{fg}\right)}} \tag{18}$$

where C_{ij} is the number of elements of true class i that have been assigned to class j by the classifier. MCC lives in the range $[-1\,1]$, 1 indicating perfect classification.

3. Results

The results for all datasets are summarized in Tables 3 and 4; Table 3 depicts the accuracy% and the MCC in the validation and testing datasets, while the optimized parameters and computational times for all three methodologies, together with the selected fuzzy partitions for the two RBF models, can be seen in Table 4. All computational times were measured on a PC with Intel Core 2 Quad processor (2.83 GHz) with 4 GB of RAM. All values shown in the two tables correspond to the model achieving the best result in terms of accuracy% in the validation dataset; for the ESA-NSFM algorithm, the population average and 95% confidence intervals are also given in parentheses. Fig. 4 presents the evolution of the accuracy% in the validation dataset for the best chromosome produced by ESA-NSFM in each case; the horizontal axis depicts the total iterations run (TIs), combining the generations of the evolutionary procedure with the iterations of SA. It should be noted that for the sake of improving clarity, each graph displays only the TIs needed for reaching for the first time the respective best solution. The computational times shown in Table 4, however, correspond to the full time needed for running all 10,000 iterations which were performed in each case.

4. Discussion

Each iteration in the graphs shown in Fig. 4 actually represents the performance of NSFM on a specific fuzzy partition tested by ESA. It is obvious that the performance of NSFM varies significantly depending on the fuzzy partition, but in all tested cases, ESA eventually manages to greatly improve the performance of NSFM. It should also be noted that the ESA-NSFM classifier accuracy often remains steady for a number of iterations; this means that the algorithm is briefly trapped to a solution corresponding to a local minimum. The use of ESA helps the algorithm to escape the local

Table 3
Results for all datasets – Accuracy% and MCC in validation and testing subsets.

Dataset	Algorithm	Accuracy% validation	Accuracy% testing	MCC validation	MCC testing
Cardiotocography FHR	ESA-NSFM	**83.8** (83.2 [83.0–83.4])	**82.9** (82.5 [82.1–82.9])	**0.80** (0.79 [0.79–0.79])	**0.79** (0.79 [0.79–0.79])
	SFM	80.8	80.8	0.76	0.77
	SVM	80.5	80.7	0.74	0.75
Cardiotocography NSP	ESA-NSFM	**95.1** (94.9 [94.8–95.0])	93.2 (93.1 [92.8–93.4])	**0.86** (0.85 [0.85–0.85])	**0.82** (0.81 [0.81–0.82])
	SFM	94.0	**93.6**	0.83	0.80
	SVM	92.9	93.0	0.80	0.80
EEG Eye State	ESA-NSFM	**96.7** (96.4 [96.4–96.5])	**96.8** (96.7 [96.7–96.8])	**0.93** (0.93 [0.93–0.93])	**0.94** (0.93 [0.93–0.94])
	SFM	95.9	96.2	0.92	0.92
	SVM	96.3	96.6	**0.93**	0.93
Pima Indians Diabetes	ESA-NSFM	**82.8** (81.6 [81.0–82.1])	**81.8** (79.8 [78.8–80.8])	**0.75** (0.72 [0.71–0.73])	**0.76** (0.72 [0.70–0.74])
	SFM	80.2	78.7	0.68	0.67
	SVM	78.1	78.7	0.70	0.71
Thyroid ANN	ESA-NSFM	**95.5** (95.4 [95.4–95.5])	**95.7** (95.4 [95.4–95.5])	**0.81** (0.79 [0.79–0.79])	**0.79** (0.78 [0.78–0.79])
	SFM	94.8	95.1	0.75	0.74
	SVM	95.3	95.2	**0.81**	0.78
Vertebral Column – 2 Classes	ESA-NSFM	**93.6** (93.5 [93.5–93.6])	**88.3** (87.0 [86.5–87.5])	**0.83** (0.82 [0.81–0.83])	**0.80** (0.79 [0.78–0.80])
	SFM	91.0	84.4	0.76	0.71
	SVM	89.7	84.4	0.76	0.73
Vertebral Column – 3 Classes	ESA-NSFM	**94.9** (93.3 [92.8–93.8])	**88.3** (86.4 [85.9–86.9])	**0.92** (0.88 [0.87–0.89])	**0.89** (0.86 [0.85–0.87])
	SFM	91.0	80.5	0.83	0.77
	SVM	88.6	81.6	0.81	0.78
Wisconsin Diagnostic Breast Cancer (WDBC)	ESA-NSFM	99.3 (99.0 [98.8–99.2])	**96.5** (94.3 [93.1–95.5])	0.99 (0.98 [0.98–0.98])	**0.94** (0.91 [0.86–0.96])
	SFM	97.9	93.0	0.96	0.88
	SVM	**100.0**	96.5	**1**	0.93
Wisconsin Original Breast Cancer (WOBC)	ESA-NSFM	**99.4** (98.9 [98.8–99.0])	**99.4** (98.9 [98.8–99.0])	**0.99** (0.98 [0.98–0.98])	**0.99** (0.98 [0.98–0.99])
	SFM	97.7	98.2	0.95	0.96
	SVM	98.3	98.8	0.96	0.97

The table depicts the best result in terms of accuracy% in the validation dataset for all tested classifiers. In the case of the ESA-NSFM classifier, the best result is followed in parentheses by the average and 95% confidence intervals. Bold numbers indicate the best performance per dataset, for each one of the evaluation criteria.

minimum and discover a path that leads ultimately to an improved solution. It should be noted that the path towards the improved solution, often passes transiently through worse solutions.

A comparison between the three classifiers shows that in almost all cases the best solution found by the ESA-NSFM algorithm outperforms the SFM algorithm and the SVMs in terms of the best accuracy% calculated in both validation and testing sets. A similar result is observed when comparing the methods in terms of the MCC, as the best solution found by ESA-NSFM outperforms its rivals in 8 out of 9 datasets, taking into account the validation or testing data. As the proposed approach is a stochastic process, generating a different result for each run and each chromosome, it is also interesting to attempt a comparison using the average solution of the ESA-NSFM population, instead of the best. The proposed approach again proves to be superior in the majority of the datasets, considering either the accuracy% or the MCC criterion, even when applying the lowest bound of the 95% confidence interval as a comparison measure. The superiority of ESA-NSFM becomes more obvious, keeping in mind that the solution found by the comparison classifiers was the global optimum solution that these classifiers could achieve, as a result of the exhaustive procedure used for training them. Even though comparing the average solution of a number of runs of the proposed approach to the best possible solution found by its rivals puts ESA-NSFM at a disadvantage, the latter can still provide statistically significantly higher accuracy.

As far as speed is concerned, the ESA-NSFM algorithm requires more computational effort to arrive to completion compared to the SFM training method. This is expected due to the significant increase in the size of the search space that needs to be explored by the proposed approach. However, it is important to note that in most cases, the RBF model generated by the ESA-NSFM algorithm outperforms the final outcome of SFM in terms of accuracy%, within a few iterations. Thus, the proposed approach generates a highly accurate model within the first iterations, but further improvements are being made until completion of the algorithm. SVM training on the other hand proves to be the slowest method, as the exhaustive search procedure in this case runs concurrently for two parameters.

The total computational time needed for all classifiers is profoundly affected by the size of the dataset, defined by the numbers of training data and input variables, though the latter does not appear to have any explicit effects on their accuracy performance. The number of output classes on the other hand, affects the classifier performance; to be more specific, the prediction accuracy seems to drop as the number of output classes in the dataset increases. A good example can be drawn from the two Cardiotocography datasets, where there is a significant deviation between the performance of the classifiers for the NSP case (3 classes) and the FHR case (10 classes). This can be attributed to the fact that the classification task becomes more difficult, as the number of possible output categories increases. It is interesting to note, however,

Table 4
Results for all datasets – Fuzzy partition, optimized parameters and computational time.

Dataset	Algorithm	Fuzzy partition	Optimized parameters[a]	Computational time (s)
Cardiotocography FHR	ESA-NSFM	[23 6 19 4 4 44 44 4 4 45 4 5 22 15 25 4 23 9 36 10 4]	352 (453 [402–504])	603 (834 [739–929])
	SFM	10	219	69
	SVM	–	36.76–3.48	13535
Cardiotocography NSP	ESA-NSFM	[24 4 17 50 28 15 50 38 12 18 15 12 5 28 41 39 30 37 4 23 4]	825 (800 [755–845])	604 (811 [733–889])
	SFM	26	845	70
	SVM	–	21.11–2.55	5612
EEG Eye State	ESA-NSFM	[21 20 18 37 48 35 48 32 50 37 32 33 30 26]	2801 (2737 [2673–2801])	1968 (1832 [1800–1864])
	SFM	23	1867	181
	SVM	–	16.00–1.15	23412
Pima Indians Diabetes	ESA-NSFM	[4 4 9 4 6 4 36 4]	31 (45 [38–52])	102 (99 [93–105])
	SFM	7	35	8
	SVM	–	2.00–3.25	1383
Thyroid ANN	ESA-NSFM	[6 50 43 46 4 50]	183 (270 [252–288])	832 (819 [730–908])
	SFM	28	300	104
	SVM	–	48.50–2.30	20675
Vertebral Column – 2 Classes	ESA-NSFM	[9 31 4 28 4 4]	36 (29 [26–32])	14 (13 [11–15])
	SFM	9	25	1
	SVM	–	3.61–1.23	212
Vertebral Column – 3 Classes	ESA-NSFM	[4 14 4 4 5 50]	22 (23 [20–26])	11 (13 [12–14])
	SFM	13	45	2
	SVM	–	25.99–3.73	674
Wisconsin Diagnostic Breast Cancer (WDBC)	ESA-NSFM	[34 4 23 50 36 43 26 7 7 13 50 4 20 50 18 4 4 14 21 45 4 4 4 10 10 42 14 4 5 9]	149 (239 [215–263])	95 (88 [78–98])
	SFM	19	221	8
	SVM	–	0.38–4.93	912
Wisconsin Original Breast Cancer (WOBC)	ESA-NSFM	[7 4 35 4 50 28 8 8 50 4]	171 (198 [189–207])	42 (45 [44–46])
	SFM	17	203	7
	SVM	–	0.03–3.03	851

The table depicts the parameters corresponding to the run achieving the highest accuracy% in the validation dataset. In the case of the ESA-NSFM classifier, the best result is followed in parentheses by the average and 95% confidence intervals.

[a] For the two RBF classifiers, optimized parameters correspond to the selected number of nodes; for the SVM classifier, the optimized parameters consist of the penalty factor C and the width of the Gaussian kernel σ, given in the form: $C – \sigma$.

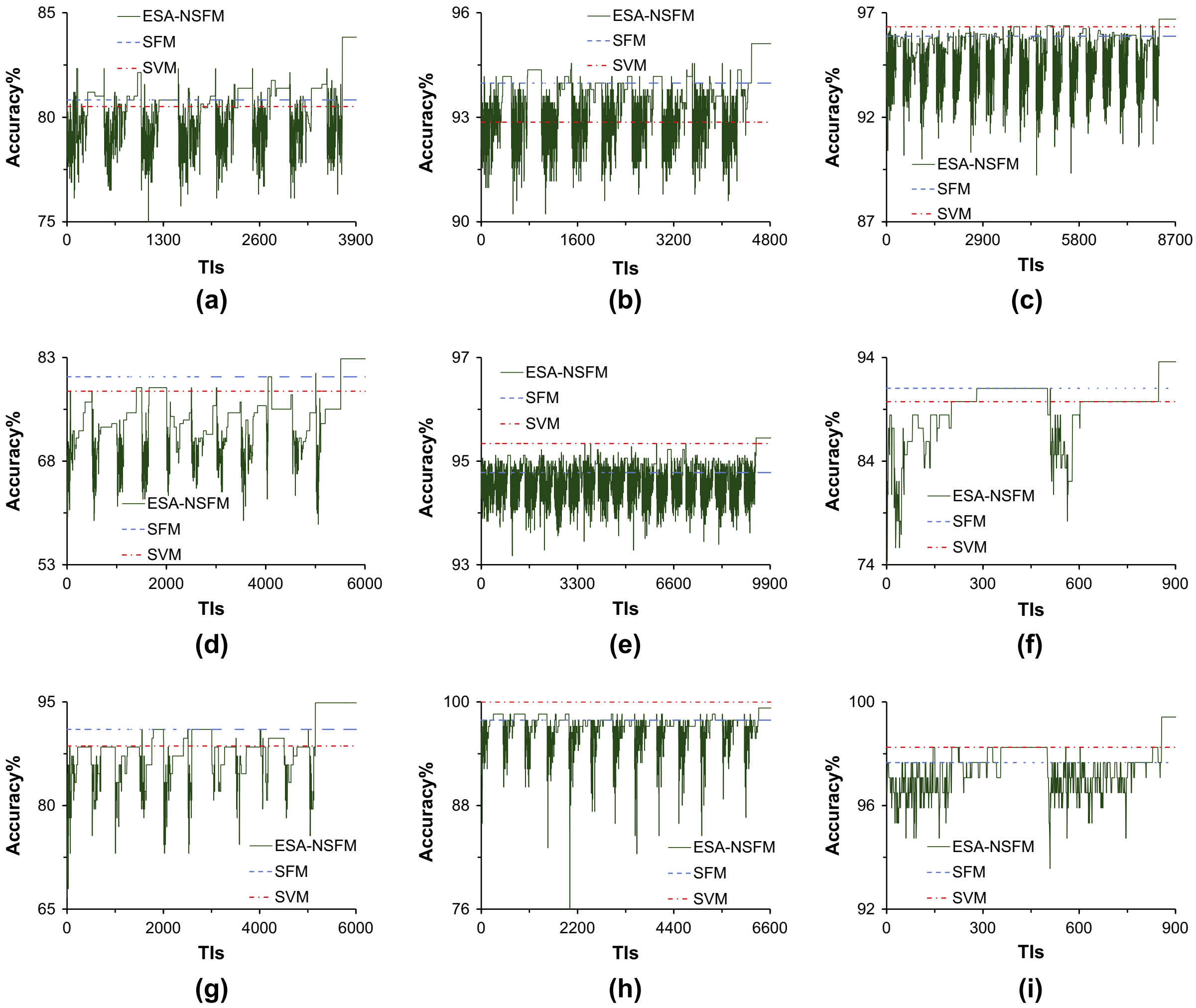

Fig. 4. Evolution of accuracy% per iteration in the validation set for the three methodologies. The best solution found in terms of accuracy in the validation dataset is depicted for the ESA-NSFM algorithm. (a) Cardiotocography FHR, (b) Cardiotocography NSP, (c) EEG Eye State, (d) Pima Indians Diabetes, (e) Thyroid ANN, (f) Vertebral Column – 2 Classes, (g) Vertebral Column – 3 Classes, (h) Wisconsin Diagnostic Breast Cancer (WDBC), (i) Wisconsin Original Breast Cancer (WOBC).

that the ESA-NSFM algorithm appears to have a slightly larger advantage over the SVM classifier in datasets with more than two output classes. A possible explanation is that SVMs are binary classifiers which need to be modified accordingly for predicting more than two output classes, in contrast to the ESA-NSFM algorithm which can inherently handle such cases.

As with all data-driven classification approaches, the presence of outliers in the training data may have a detrimental effect on the performance of the proposed classifier; therefore, it is recommended to perform data preprocessing prior to applying the method to raw medical records (e.g. see [44]). As far as the treatment of missing values is concerned, the technique used in this study, i.e. discarding the respective examples, works adequately for the particular datasets. However, it could result in problems with smaller datasets, as there may not be enough data left for successfully training the RBF network. In this case, a number of more elaborate techniques, which have been found to work well with RBF classifiers [45], could be applied in conjunction with the proposed methodology for missing value imputation. Finally, it should be noted that an increase in the number of examples and the problem dimensionality, results to higher computational times. The increase in complexity could ultimately lead to computational times that are not practically feasible. However, the performance of the proposed approach on datasets with relatively high complexity and the advantage of acquiring an adequate model within the first few iterations of the algorithm, indicate that the algorithm can handle successfully such cases.

5. Conclusions

This paper introduces a new method for training classifiers based on RBF neural networks. At the heart of the proposed approach lies the NSFM training algorithm, which has been found to produce RBF networks of increased accuracy in shorter computational times. The NSFM algorithm operates on a given non-symmetric fuzzy partition of the input space, so as to calculate the number of RBF hidden nodes and their locations and give birth to a fully trained RBF classifier. The ESA technique is then suitably modified and wrapped around the NSFM algorithm to optimize the fuzzy partition. The evolutionary nature of ESA, helps the iterative search procedure to avoid getting stuck in local minima that could result from the traditional SA algorithm, thus producing classifiers with increased accuracy. To assess the performance of the ESA-NSFM algorithm, nine datasets of medical interest are utilized. Two different types of classifiers are used for comparison purposes, namely RBF neural network classifiers trained with the SFM algorithm and standard SVM classifiers with Gaussian kernel functions.

The results show that the proposed approach outperforms its rivals in terms of more accurate diagnosis, in the majority of tested cases; furthermore it proves to be faster, compared to the SVM approach.

Future research plans include exploiting the inherent potential of ESA for parallel implementation to design a distributed optimizer. This is expected to be of particular interest in the case of large databases of medical records. The authors also plan to release to the research community a distributable implementation of the ESA-NSFM algorithm.

Appendix A. Brief dataset descriptions

A.1. Cardiotocography datasets (FHR, NSP)

The two cardiotocography datasets contain a number of fetal cardiotocograms (CTGs), which were automatically processed with their respective diagnostic features. The aim of the "Cardiotocography dataset FHR" is to predict the fetal heart rate (FHR) morphologic pattern (1–10), whereas the purpose of the "Cardiotocography dataset NSP" is to predict the fetal state (Normal, Suspect, Pathologic). The available features for building the classifier are common for the two datasets.

A.2. EEG Eye State

This dataset is formulated from continuous EEG measurement with the Emotiv EEG Neuroheadset. The objective is to determine the state of the eye, based on 14 different EEG features.

A.3. Pima Indian Diabetes

The scope of the Pima Indian Diabetes dataset is to build a classifier for predicting whether a patient will develop diabetes, based on clinical findings. All patients in the database were females at least 21 years old of Pima Indian heritage. The dataset contains 39 instances with missing values.

A.4. Thyroid ANN

The objective in this dataset is to discriminate between three possible states of the thyroid gland (euthyroidism, hyperthyroidism or hypothyroidism). Input data to the classifier include 5 different hormones that are acquired after testing blood samples from the patients.

A.5. Vertebral Column Datasets (3 Classes–2 Classes)

Vertebral Column datasets are related to diagnosis of people with vertebral column disorders. The data have been organized in two different, but related classification tasks. The task in "Vertebral Column Dataset – 3 Classes" is to classify patients to one out of three categories: Normal (healthy people), Disk Hernia or Spondylolisthesis. The task in "Vertebral Column Dataset – 2 Classes" on the other hand is to discriminate healthy people (Normal category) from those who suffer from Disk Hernia or Spondylolisthesis (Abnormal category).

A.6. Wisconsin Original Breast Cancer (WOBC)–Wisconsin Diagnostic Breast Cancer (WDBC)

These datasets [46,47] involve measurements taken from digitized images of breast mass. The objective in both datasets is to diagnose breast cancer in patients, classifying tumors as benign or malignant; each dataset is based on different input variables. The WOBC dataset contains 16 instances with missing values.

References

[1] Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. Artif Intell Med 2001;23:89–109.
[2] Haykin S. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River (NJ): Prentice Hall; 1999.
[3] Dua R, Beetner DG, Stoecker WV, Wunsch DC. Detection of basal cell carcinoma using electrical impedance and neural networks. IEEE Trans Biomed Eng 2004;51:66–71.
[4] Wang HQ, Wong HS, Zhu H, Yip TT. A neural network-based biomarker association information extraction approach for cancer classification. J Biomed Inform 2009;42:654–66.
[5] Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF. GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. Int J Med Inform 2005;74:491–503.
[6] Nahar J, Imam T, Tickle KS, Phoebe Chen Y-P. Computational intelligence for heart disease diagnosis: a medical knowledge driven approach. Exp Syst Appl 2013;40:96–104.
[7] Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JADTG. Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. Neural Networks 2008;21:427–36.
[8] Peng YW, Zhiqing, Jiang J. A novel feature selection approach for biomedical data classification. J Biomed Inform 2010;43:15–23.
[9] Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform 2002;35:352–9.
[10] Lakoumentas J, Drakos J, Karakantza M, Nikiforidis GC, Sakellaropoulos GC. Bayesian clustering of flow cytometry data for the diagnosis of B-Chronic Lymphocytic Leukemia. J Biomed Inform 2009;42:251–61.
[11] Zhao D, Weng C. Combining PubMed knowledge and EHR data to develop a weighted bayesian network for pancreatic cancer prediction. J Biomed Inform 2011;44:859–68.
[12] Tortajada S, Fuster-Garcia E, Vicente J, Wesseling P, Howe FA, Julià-Sapé M, et al. Incremental Gaussian discriminant analysis based on Graybill and deal weighted combination of estimators for brain tumour diagnosis. J Biomed Inform 2011;44:677–87.
[13] Akbarzadeh-T MR, Moshtagh-Khorasani M. A hierarchical fuzzy rule-based approach to aphasia diagnosis. J Biomed Inform 2007;40:465–75.
[14] Son CS, Kim YN, Kim HS, Park HS, Kim MS. Decision-making model for early diagnosis of congestive heart failure using rough set and decision tree approaches. J Biomed Inform 2012;45:999–1008.
[15] Cova TF, Pereira JL, Pais AA. Is standard multivariate analysis sufficient in clinical and epidemiological studies? J Biomed Inform 2013;46:75–86.
[16] Moody J, Darken C. Fast learning in networks of locally-tuned processing units. Neural Comput 1989;2:281–94.
[17] Alexandridis A, Chondrodima E, Sarimveis H. Radial Basis Function network training using a non-symmetric partition of the input space and Particle Swarm Optimization. IEEE Trans Neural Netw Learn Syst 2013;24:219–30.
[18] Bishop CM. Neural networks for pattern recognition. Oxford: University Press; 1995.
[19] Sateesh Babu G, Suresh S. Parkinson's disease prediction using gene expression – a projection based learning meta-cognitive neural classifier approach. Exp Syst Appl 2013;40:1519–29.
[20] Ghosh-Dastidar S, Adeli H, Dadmehr N. Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. IEEE Trans Biomed Eng 2008;55:512–8.
[21] Wei CH, Li Y, Huang PJ. Mammogram retrieval through machine learning within BI-RADS standards. J Biomed Inform 2011;44:607–14.
[22] Engelbrecht A. Computational intelligence: an introduction. 2nd ed. Chichester: John Wiley & Sons, Ltd.; 2007.
[23] Ilhan I, Tezel G. A genetic algorithm-support vector machine method with parameter optimization for selecting the tag SNPs. J Biomed Inform 2013;46:328–40.
[24] Lambrou A, Papadopoulos H, Gammerman A. Reliable confidence measures for medical diagnosis with evolutionary algorithms. IEEE Trans Inf Technol Biomed 2011;15:93–9.
[25] Esposito M, De Falco I, De Pietro G. An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease. Int J Med Inform 2011;80. pp. e245-e54.
[26] Avci E. A new intelligent diagnosis system for the heart valve diseases by using genetic-SVM classifier. Exp Syst Appl 2009;36:10618–26.
[27] Sarimveis H, Alexandridis A, Mazarakis S, Bafas G. A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms. Comput Chem Eng 2004;28:209–17.
[28] Alexandridis A, Sarimveis H, Ninos K. A radial basis function network training algorithm using a non-symmetric partition of the input space – application to a model predictive control configuration. Adv Eng Software 2011;42:830–7.
[29] Aydin ME, Fogarty TC. A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems. J Heuristics 2004;10:269–92.
[30] Alexandridis A, Sarimveis H, Bafas G. A new algorithm for online structure and parameter adaptation of RBF networks. Neural Networks 2003;16:1003–17.
[31] Darken C, Moody J. Fast adaptive K-means clustering: some empirical results. In: IEEE INNS international joint conference on neural networks. San Diego, CA;1990. p. 233–8.

[32] Alexandridis A, Stogiannos M, Kyriou A, Sarimveis H. An offset-free neural controller based on a non-extrapolating scheme for approximating the inverse process dynamics. J Process Control 2013;23:968–79.
[33] Alexandridis A, Patrinos P, Sarimveis H, Tsekouras G. A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models. Chemom Intell Lab Syst 2005;75:149–62.
[34] Nie J. Fuzzy control of multivariable nonlinear servomechanisms with explicit decoupling scheme. IEEE Trans Fuzzy Syst 1997;5:304.
[35] Metropolis M, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953;21:1087–92.
[36] Asuncion A, Newman DJ. UCI machine learning repository. Irvine (CA): Univ. California Irvine; 2007.
[37] Sarimveis H, Doganis P, Alexandridis A. A classification technique based on radial basis function neural networks. Adv Eng Software 2006;37:218–21.
[38] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. 5th Annual ACM workshop on computational learning theory; 1992. p. 144–52.
[39] Vapnik V. The nature of statistical learning theory. London: Springer-Verlag; 1995.
[40] Hsu CW, Chang CC, Lin CJ. A practical guide to support vector classification. Department of Computer Science, National Taiwan University; 2003.
[41] Matthews B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta – Protein Struct 1975;405:442–51.
[42] Gorodkin J. Comparing two K-category assignments by a K-category correlation coefficient. Comput Biol Chem 2004;28:367–74.
[43] Jurman G, Furlanello C. A unifying view for performance measures in multi-class prediction. PLoS One 2012;7:e41882.
[44] Di Noia T, Ostuni VC, Pesce F, Binetti G, Naso D, Schena FP, et al. An end stage kidney disease predictor based on an artificial neural networks ensemble. Exp Syst Appl 2013;40:4438–45.
[45] Luengo J, García S, Herrera F. A study on the use of imputation methods for experimentation with Radial Basis Function Network classifiers handling missing attribute values: the good synergy between RBFNs and EventCovering method. Neural Networks 2010;23:406–18.
[46] Wolberg WH, Mangasarian OL. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In: Proceedings of the National Academy of Sciences. USA; 1990. p. 9193–6.
[47] Mangasarian OL, Wolberg WH. Cancer diagnosis via linear programming. SIAM News 1990;23:1–18.