An intelligent temporal pattern classification system using fuzzy temporal rules and particle swarm optimization

S GANAPATHY[1,*], R SETHUKKARASI[1], P YOGESH[1], P VIJAYAKUMAR[2] and A KANNAN[1]

[1]Department of Information Science and Technology, Anna University, Chennai 600 025, India
[2]Department of Computer Science and Engineering, University College of Engineering, Tindivanam, Villupuram 604 001, India
e-mail: ganapathy.sannasi@gmail.com; sethumaaran@gmail.com; yogesh@annauniv.edu; vijibond2000@gmail.com; kannan@annauniv.edu

Abstract. In this paper, we propose a new pattern classification system by combining Temporal features with Fuzzy Min–Max (TFMM) neural network based classifier for effective decision support in medical diagnosis. Moreover, a Particle Swarm Optimization (PSO) algorithm based rule extractor is also proposed in this work for improving the detection accuracy. Intelligent fuzzy rules are extracted from the temporal features with Fuzzy Min–Max neural network based classifier, and then PSO rule extractor is used to minimize the number of features in the extracted rules. We empirically evaluated the effectiveness of the proposed TFMM-PSO system using the UCI Machine Learning Repository Data Set. The results are analysed and compared with other published results. In addition, the detection accuracy is validated by using the ten-fold cross validation.

Keywords. Temporal fuzzy min–max (TFMM) neural network; particle swarm optimization algorithm (PSOA); pattern classification; rule extraction.

1. Introduction

Data mining is concerned with analysing large volumes of data to automatically discover interesting relationships which in turn lead to better understanding of the underlying processes. Temporal data mining is the application of data mining techniques on temporal data which is helpful for retrieving the temporal relationships from an ordered data streams (Laxman & Sastry 2006). Basically, temporal data mining provides facilities for the analysis of past, present and future data for finding temporal patterns and regularities in sets of temporal data. Storage, retrieval and mining of temporal patterns from temporal databases pertaining

*For correspondence

to medical applications are gaining popularity recently. This helps to perform temporal pattern analysis, where temporal databases are used for storing patient histories. These databases provide features for temporal rules extraction Meamarzadeh *et al* (2009), which may be further used for decision support system.

Classification has been an active area of research for many decades in data mining. Recently, there is a growing interest in developing accurate classifiers by using soft computing paradigms. From the time, Fuzzy set theory was introduced by Zadeh (1965), pattern recognition and classification problem have been intensively researched and were solved using fuzzy sets (Zhang *et al* 2009). The success of this approach is based on the fact that a fuzzy logic system using rules can suitably model the qualitative aspects of commonsense knowledge and reasoning processes of human without applying precise quantitative analysis (Wai & Lee 2008). Recently, a great deal of attention is paid to the integration of fuzzy logic and neural networks for developing effective intelligent systems (Wai & Lee 2008; Zhang *et al* 2009).

Fuzzy Min-Max neural network is a classifier that consists of computing the union of fuzzy membership function values produced from different fuzzy set hyperboxes (Simpson 1992; 1993; Quteishat *et al* 2010). Moreover, a hyperbox is used to define a pattern for full class membership, which is a region of the n-dimensional pattern space. A hyperbox is completely defined by its minimum and maximum points. A membership function in this context is defined with respect to these hyperbox minimum and maximum points. In the Weighted Fuzzy Min–Max neural network (WFMM) proposed by Kim & Yang (2005), the membership function assigns values by considering not only the occurrence of input patterns but also the frequency of the occurrences. The WFMM works better for datasets with highly uneven distribution of features. Quteishat *et al* (2010) proposed a modified FMM neural network with a confidence factor which is calculated by each hyperbox, where a user-defined threshold was developed for pruning the hyperboxes with low confidence factors. An Euclidean distance measure was introduced in that work for predicting the target class associated with new input patterns.

There is a general belief that social sharing of information among individuals of a population, may provide an evolutionary advantage, and there are numerous examples coming from nature to Particle Swarm Optimization (PSO). The PSO method is a member of the wide category of Swarm Intelligence methods (Kennedy & Eberhart 2001). Initially, Kennedy & Eberhart (1995) proposed a PSO for the simulation of social behaviour and also introduced as an optimization method. The main advantage of PSO is that it can be easily implemented and is computationally inexpensive due to its less memory and CPU speed requirements (Eberhart *et al* 1996). Furthermore, it works without gradient information of the objective function being and hence PSO has been proved to be an efficient method for numerous general optimization problems. PSO has been successfully applied in the recent days to a range of problems, from function optimization to the training of neural networks.

In this paper, we propose a new temporal pattern classification system by extending the Fuzzy Min–Max (FMM) neural network Quteishat *et al* (2010) with temporal features from temporal clinical datasets. In addition, we propose a new PSO-based rule extractor in order to retrieve and optimize the fuzzy temporal rules. First, temporal FMM neural network is proposed to generate four hyperboxes. These hyperboxes are pruned based on rules. Second, 'open hyperboxes' are generated from the original hyperboxes in which PSO is used to minimize the number of input features in the rules and to maximize classification accuracy. Third, the proposed TFMM-PSO system has been evaluated using University of California Irvine (UCI) Machine Learning Repository Data Set and the results are analysed and compared with other published results. Finally, the detection accuracy is validated by using the tenfold cross validation. The remainder of this paper is organized as follows. Section 2 provides the features of some of the related works.

Section 3 discusses the system architecture of the proposed work. Section 4 explains about our proposed temporal constraint based classification and rule extraction technique used in this proposed work. Section 5 analyses the comparative performance of our proposed algorithm with the other existing classification techniques. Section 6 gives concluding remarks and suggests future directions.

2. Literature survey

In the past, lots of works have been carried out in classification areas based upon the theory of statistics, and the theory of databases (Simpson 1992, 1993; Quteishat *et al* 2010; Oong & Isa 2011). Temporal classification techniques have been paid much attention in the past two decades. Moreover, representation and analysis of temporal data pertaining to medical diagnosis systems is gaining importance recently. This helps to perform temporal pattern analysis, where temporal databases are used for storing patient histories. These databases provide features for temporal rules extraction Meamarzadeh *et al* (2009), which may be further used for decision support system. Most of the existing systems Adlassnig *et al* (2006) concentrate on mining clinical data based on snapshot of time points. Since clinical data is a series of observations taken at different time points, it is necessary to classify the data based on the time period of observation.

Data mining in time series medical databases is becoming increasingly important in the field of medical diagnosis and in an expert decision making system. Nowicki (2009) presented a new approach to fuzzy classification within the case of missing information. The rough fuzzy sets are incorporated into Mamdani-type neuro-fuzzy structures, and therefore the rough neuro-fuzzy classifier is derived. Han & Qiao (2010) presented a novel growing-and-pruning (GP) approach, which optimizes the structure of a fuzzy neural network (FNN). Oong & Isa (2011) presented a novel replacement evolutionary approach known as the Hybrid Evolutionary Artificial Neural Network (HEANN) for simultaneously evolving a man-made neural networks (ANNs) topology and weights. Gui & Qiao (2012) outlined a learning algorithm for nonlinear modelling and classification with that of radial basis operate neural networks to quicken the learning speed and optimization method for the RBF neural networks.

In the past, many researchers have proposed various rule extraction methods (Quteishat *et al* 2010; Taylor & Darrah 2005; Campos *et al* 2004; Setiono *et al* 2002) based on ANN models. A method for extracting IF–THEN rules from a multilayer perceptron network was proposed by Campos *et al* (2004), in which the content of the antecedent and consequent parts of the rules were same as in the original database. Another rule extraction technique proposed by Carpenter & Tan (1995) comprised of two stages namely pruning and quantization. In pruning, the network structure was used to remove excessive recognition categories such as weights. In quantizing, the continuous network weights were applied for translating the system states into a set of rules. A rule extraction technique was proposed by Setiono *et al* (2002) activation function was approximated at each hidden neuron using a linear function for converting into three or five pieces. In Xiuju & Lipo (2002), a data dimensionality reduction technique was introduced for removing redundant features from the input data. Moreover, a gradient-based rule extraction method was also used in that work to extract rules.

Quteishat *et al* (2010) introduced a pattern classification and rule extraction system which comprises of a modified FMM network and a GA rule extractor which is denoted as FMM-GA. This FMM-GA (Quteishat *et al* 2010) system is a modified FMM network in which they used a pruning procedure to eliminate hyperboxes with low confidence factors. Zhang *et al* (2011)

proposed a new algorithm for pattern classification using a new Data Core based Fuzzy Min-Max Neural Network. Comparing with FMCN, this work is different since the hyperbox can be expanded and can overlap repeated with the previous hyperboxes. Therefore, it generates minimum number of hyperboxes for rule extraction.

Comparing with all the works present in the literature, the system proposed in this paper is different and efficient in many ways. First, a new classification algorithm using Fuzzy temporal approach has been proposed to capture the dynamic nature of diseases patterns of the medical dataset. Second, it uses fuzzy rules for effective decision making. Third, it proposes PSO based rule extraction method for effective classification. Finally, it uses the bench mark dataset and ten-fold cross validation for proving the efficiency of the proposed system.

3. System architecture

The architecture of the system proposed in this work is shown in figure 1. It consists of six major components namely, UCI Machine Learning Repository Data Set, data collection agent, classification module, rule manager, rule base and user interface.

UCI machine learning repository data set: Input to the TFMM-PSO system is referred from the bench mark UCI Machine Learning Repository Data Set which is used in this work for carrying out the experiments.

Data collection agent: The Data collection agent collects the necessary data from the data set. These data are sent to the classification module for classification of the data.
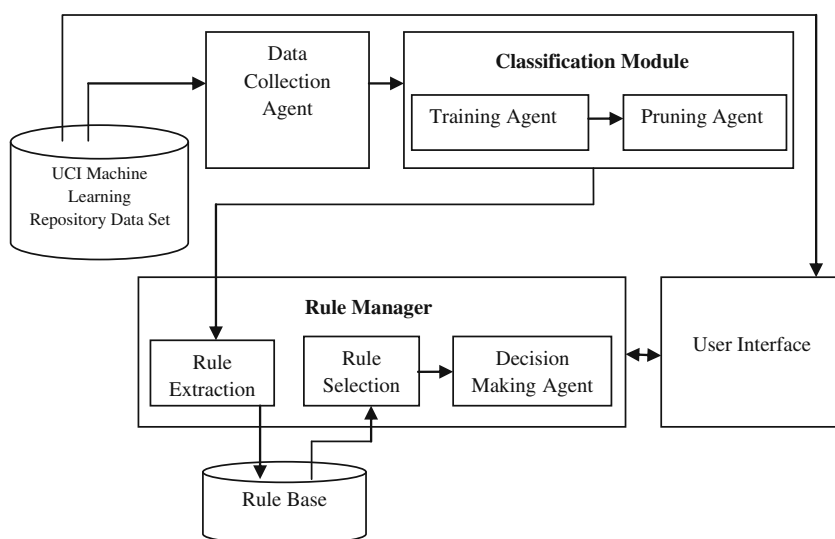
Figure 1. System architecture.

Classification module: This module is used to classify the data by the help of training and pruning agents.

 (i) Training agent – This agent trains the data which are received from the data collection agent.
(ii) Pruning agent – This agent prunes the data which are received from the training agent.

Rule manager: The rule manager is responsible for rule extraction, rule selection and decision making. First, rules are extracted based on the information provided by the classification module. They are stored in the rule base. During testing, the rule selection subsystem selects the suitable rules and sends them to the decision making agent. The decision making agent classifies the test data using these rules and makes a suitable decision on diagnosis and planning for therapy.

Rule base: It contains IF THEN fuzzy temporal rules for classifying the data. The data set features are classified based on the diagnostic parameters. Fuzzy IF-THEN rules are formed based on the classification module and stored in the rule base.

User interface: It collects the data from the UCI Machine Learning Repository Data Set and sends them to decision making agent for classification. Moreover, it performs validation using ten-fold cross validation whenever the user initiates it.

4. Classification and rule extraction

In this section, we explain the proposed classifier called Temporal Fuzzy Min-Max Network based pattern classifier which has been developed by extending the Modified Fuzzy Min Max network (Quteishat *et al* 2010). In addition, the rule extraction method using Particle Swarm Optimization proposed in this work is also explained.

4.1 *Temporal fuzzy min max pattern classifier*

In Temporal Fuzzy Min-Max Pattern Classification algorithm, we introduce timestamp analysis in addition to the fuzzy min–max procedure. Therefore, instead of two subsets (training and testing) as in the existing original FMM, the input/output data set is divided into four subsets namely training, rule extraction, rule selection and testing. After that, the hyperboxes are created which undergo a pruning process based on a user defined threshold. The training data set is used for creating relevant hyperboxes in the proposed temporal FMM. The details are as follows.

4.1a *Learning in temporal FMM network*: Temporal FMM is an incremental and intelligent learning system. In Temporal FMM, two fundamental techniques of granular computing are used as shown in figure 2. In this model, time intervals are used to construct hyperboxes and fuzzy sets are utilized for decision making. In TFMM, hyperboxes with fuzzy sets are formed to represent the rules learned by the network. Moreover, it learns incrementally in two passes through the data set and refines the existing pattern classes. It also has the ability to add new pattern classes whenever it is necessary through the user interface. Learning in TFMM consists of series of expansion and contraction processes that improve its hyperboxes for establishing decision boundaries between the available classes. Moreover, we introduce additional boundaries
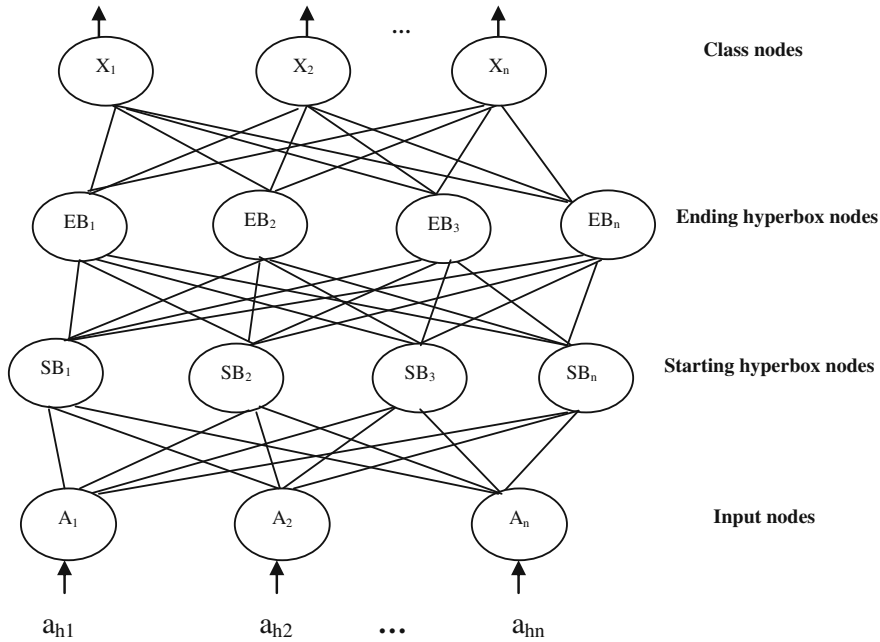
Figure 2. Four layer TFMM network.

within the classes based on the timestamps t_1 and t_2. Whenever overlapping hyperboxes from different classes occur in the input space, contraction is performed to eliminate the overlapped regions as in the FMM model (Simpson 1992).

The structure of TFMM consists of four layers of nodes, as shown in figure 2. The first layer is the input layer. It contains input nodes that are equal to the number of dimensions of the input pattern. Second layer is the starting hyperbox layer. It contains hidden nodes that are equal to the number of nodes in the input layers. The second hidden layer is called the ending hyperbox layer. Each node of this second hidden layer represents a hyperbox temporal fuzzy set. The nodes in the hidden layers have transfer functions for transforming the hyperbox membership functions. The minimum and maximum points are stored in matrices *V* and *W,* respectively, where *V* is computed from starting hyperbox and *W* is computed from the ending hyperbox. In decision making, if a soft decision is required, the threshold is kept minimum. If a hard decision is required, the procedure in Simpson (1992) is extended by identifying the output layer node with the maximum value and set as the threshold.

The hyperboxes in TFMM have a range between 0 and 1 along each dimension. Hence, the pattern spaces are represented using *n*-dimensional unit cube (I^n). The hyperbox membership functions are defined with respect to the temporal minimum and maximum points of the hyperboxes. They denote the degree to which a pattern fits in the hyperbox. A pattern that is fully contained in the hyperbox has a membership value of '1'. The definition of each hyperbox (B_j) is extended as

$$B_j, t_1, t_2 = \left\{ X, V_1 t_1, W_j t_2 f\left(X, V_j t_1, W_j t_2 \right) \right\} \forall X \in I^n, \tag{1}$$

where, $X = (x_1, x_2, \ldots, x_n)$ is the input pattern; $V_j = (v_{j1}\ t_{11}, v_{j2}\ t_{12}, \ldots, v_{jn}\ t_{1n})$ and $W_j = (w_{j1}\ t_{21}, w_{j2}\ t_{22}, \ldots, w_{jn}\ t_{2n})$ are the minimum and maximum points of $B_j, t_1, t_2,$

respectively; and $f(X, V_j, W_j, t_1, t_2)$ is the membership function. As an example, figure 3 shows the minimum and maximum points of a two class problem.

The learning algorithm of TFMM allows temporal overlapping of hyperboxes of the same class in an interval, but eliminates overlapping of hyperboxes from different classes for the corresponding interval. The membership function for the jth hyperbox and t time, i.e. $B_j(A_h, t_h)$, where $0 \leq B_j(A_j, t_j) \leq 1$, measures the degree to which the hth input pattern (A_h) falls outside hyperbox B_j in the particular time interval $[t_1, t_2]$ (Simpson 1992). As $B_j(A_h, t_h)$ approaches one, the pattern is said to be more 'contained' within the hyperbox. The resulting membership function is,

$$B_j(A_h, t_h) = \frac{1}{2n} \sum_{i-1}^{n} \left[max\left(0, \gamma min\left(1, a_{hi} - w_{ji}, t_1\right), t_2\right) + max\left(0, 1max\left(0, 1 - max\left(0, \gamma min\left(1, v_{ji} - a_{hi}, t_1\right), t_2\right), t_1\right)\right], \quad (2)$$

where $A_h = (a_{h1}, a_{h1}, \ldots, a_{hn}) \in I^n$ is the hth input pattern and γ is a sensitivity parameter that regulates how fast the membership value decreases as the distance between A_h and B_j increases in the time period $[t_1, t_2]$. Applying the definition of fuzzy set in TFMM, the combined fuzzy set that classifies the Kth pattern class (C_k) is defined as

$$C_k = U_{j \in k} B_j, t_1, t_2, \quad (3)$$

where, K is the index set of the hyperboxes associated with class k.

The main processing step in TFMM is concerned with finding and fine tuning the class boundaries based on the distances and also fine tuning the boundaries within the class itself based on the time interval. An example of a decision boundary for a two-class problem is shown in figure 3. The TFMM learning methodology contains both expansion and contraction procedures. Let a training set D consists of a set of M ordered pairs $\{X_h, d_h\}$ where, $X_h = (x_{h1}, x_{h2}, \ldots, x_{hn}) \in I^n$ is the input pattern and $d_h \in \{1, 2, \ldots, m\}$ is the index of one of the m classes and $t = [t_1, t_2]$.

During learning, an ordered pair from D is selected in the particular time period, and then, a hyperbox from the same class is selected and included in the input pattern and expanded to include it as in the existing work (Simpson 1992). If the existing hyperboxes cannot be expanded
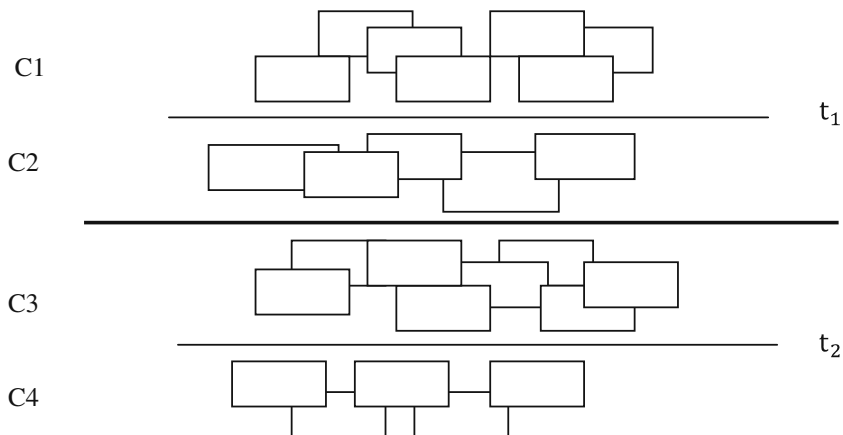

Figure 3. Example of FMM hyper boxes placed along the boundary of four class problem.

to contain the input pattern satisfying the time constraints, new hyperboxes are created. However, expanding the hyperboxes may lead to overlapping between adjacent hyperboxes. If overlapping occurs between hyperboxes of different classes, then the contraction process is initiated. However, it is not a problem when overlapping occurs for hyperboxes of the same class. The TFMM learning algorithm which is extension of MFMM Quteishat *et al* (2010) is summarized as follows:

(i) Expansion: Expandable hyperboxes are identified and expanded. If an expandable hyperbox is not found, a new hyperbox for that class is added with time period. For hyperbox B_j, t_2 to expand, the following constraint must be met:

$$n\theta \geq n\sum_{i=1}^{n}(max(w_{ji}, x_{hi}, t_2) - min(v_{ji}, x_{hi}, t_1)), \tag{4}$$

where, $0 \leq \theta \leq 1$ is a user defined threshold that determines the maximum size of a hyperbox. If the expansion and temporal constraints are met, the minimum and maximum points of the hyperbox are adjusted based on the time as follows:

$$v_{ji}^{new} = min\left(v_{ji}^{old}, x_{hi}, t_1\right) \quad \forall i, i = 1, 2, \ldots, n, \tag{5}$$

$$w_{ji}^{new} = max\left(w_{ji}^{old}, x_{hi}, t_1\right) \quad \forall i, i = 1, 2, \ldots, n. \tag{6}$$

(ii) Overlapping test: Overlapping tests are carried out to check whether there is an overlap between hyperboxes from different classes during an interval $[t_1, t_2]$, with the assumption that the overlapping value is initially set to 1, i.e., $\delta^{old} = 1$. Therefore, the new overlapping value δ^{new} is calculated as given in the following cases.

$$\begin{aligned} \text{Case 1}: v_{ji} &< v_{ki} < w_{ji} < w_{ki}, \\ \delta^{new} &= min\left(w_{ji} - v_{ki}, \delta^{old}, t_1\right), \end{aligned} \tag{7}$$

$$\begin{aligned} \text{Case 2}: v_{ki} &< v_{ji} < w_{ki} < w_{ji}, \\ \delta^{new} &= min\left(w_{ki} - v_{ji}, \delta^{old}, t_1\right), \end{aligned} \tag{8}$$

$$\begin{aligned} \text{Case 3}: v_{ji} &< v_{ki} < w_{ki} < w_{ji}, \\ \delta^{new} &= min\left(min\left(w_{ki} - v_{ji}, w_{ji} - v_{ki}, t_1\right), \end{aligned} \tag{9}$$

$$\begin{aligned} \text{Case 4}: v_{ki} &< v_{ji} < w_{ji} < w_{ki}, \\ \delta^{new} &= min\left(min\left(w_{ji}, v_{ki}, w_{ki} - v_{ji}, t_1\right), \delta^{old}, t_1\right), \end{aligned} \tag{10}$$

where, j corresponds to hyperbox B_j, t that has been expanded in the previous step and k corresponds to hyperbox B_k, t that represents another class currently being tested for possible overlapping. If $\delta^{old} - \delta^{new} > 0$ in the interval $[t_1, t_2]$, then $\Delta = i$ and $\delta^{old} = \delta^{new}$ in $[t_1, t_2]$. If this is met, there is an overlap in the Δth dimension, and the overlap test continues for the next dimension. Otherwise, the test stops, and the minimum overlap index variable (Δ) is set to indicate that the next contraction step is not necessary, i.e., Δ is set to a value smaller than '0' for the interval $[t_1, t_2]$.

(iii) Contraction: If there is an existence of overlapping between hyperboxes of different classes, then the following procedure is used to eradicate it. If $\Delta > 0$ during $[t_1, t_2]$, then the Δth

dimension of the two overlapping hyperboxes are adjusted. Any one of the n dimensions of the hyperbox is adjusted to maintain the hyperbox as large as possible. Four cases are examined in this work to make sure that a proper adjustment has been made similar to FMM Quteishat *et al* (2010) with temporal constraints.

$$\text{Case 1}: v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta} \text{ during } [t_1, t_2]$$
$$\text{w}_{j\Delta}^{\text{new}} = \text{v}_{k\Delta}^{\text{new}} = \frac{\text{w}_{j\Delta}^{\text{old}} + \text{v}_{k\Delta}^{\text{old}}}{2}, \tag{11}$$

$$\text{Case 2}: v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta} \text{ during } [t_1, t_2]$$
$$w_{k\Delta}^{new} = v_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{new}}{2}, \tag{12}$$

$$\text{Case 3a}: v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta} \text{ during } [t_1, t_2]$$
$$\left(w_{k\Delta} - v_{j\Delta}\right) < \left(w_{j\Delta} < v_{k\Delta}\right), \quad v_{j\Delta}^{new} = w_{k\Delta}^{old}, \tag{13}$$

$$\text{Case 3b}: v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta} \quad \text{during } [t_1, t_2]$$
$$\left(w_{k\Delta} - v_{j\Delta}\right) > \left(w_{j\Delta} < v_{k\Delta}\right), w_{j\Delta}^{new} = v_{k\Delta}^{old}, \tag{14}$$

$$\text{Case 4a}: v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta} \text{ during } [t_1, t_2]$$
$$\left(w_{k\Delta} - v_{j\Delta}\right) < \left(w_{j\Delta} < v_{k\Delta}\right), w_{k\Delta}^{new} = v_{j\Delta}^{old}, \tag{15}$$

$$\text{Case 4b}: v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta} \quad \text{during } [t_1, t_2]$$
$$\left(w_{k\Delta} - v_{j\Delta}\right) > \left(w_{j\Delta} v_{k\Delta}\right), \quad v_{k\Delta}^{new} = w_{j\Delta}^{old}. \tag{16}$$

4.1b *Pruning the TFMM neural network*: In this proposed work, a pruning procedure is incorporated into the TFMM after its learning phase in order to reduce the search space. The objective is to form an optimal neural network size with a high temporal pattern classification performance. During pruning, a confidence factor (CF_j) for each hyperbox B_j, t in terms of its usage frequency and predictive accuracy for time interval $[t_1, t_2]$ on a separate prediction data set is calculated as follows:

$$CF_j(t) = (1 - \gamma)U_j, t + \gamma A_j, t \text{ with } t_1 \leq t \leq t_2, \tag{17}$$

where, U_j, t and A_j, t are the usage and accuracy indices of hyperbox B_j, t and $\gamma \in [0, 1]$ is a weighting factor. The usage index of a hyperbox is defined as the number of prediction set patterns classified by any hyperbox B_j, t divided by the maximum number of prediction patterns classified by any hyperbox with the same classification class in the specified time period. On the other hand, the accuracy index of a hyperbox is defined as the number of correctly predicted set of patterns classified by any hyperbox B_j, t divided by the maximum correctly classified patterns with the same classification class in the given time period.

The confidence factor is used to identify the hyperboxes that are frequently used and generally accurate in prediction, as well as hyperboxes that are rarely used but nevertheless highly accurate. Confidence factor of each hyperbox is compared with that of a user defined threshold and hyperboxes with a confidence factor smaller than a user-defined threshold are pruned. The confidence factor is also attached to the fuzzy temporal IF–THEN rules extracted from the corresponding hyperbox in the particular time period. Thus, the degree of certainty of each fuzzy rule is tagged with a confidence measure to indicate its certainty level to the domain users. The pruned set of hyperboxes with high confidence values in the given time interval are used in next section of the proposed TFMM-PSO system.

4.2 *Particle swarm optimization based rule extractor*

In this section, the remaining hyperboxes (after pruning) from stage 1 are used to generate 'open' hyperboxes. All hyperboxes (closed and open) are fed to the PSO for evolution. The resulting hyperboxes are used for rule extraction. These procedures are as follows.

Stage 1: Four types of hyperboxes are generated by TFMM namely input, starting hyperbox, ending hyperbox and output.

Stage 2: This stage is used to evolve and select a set of hyperboxes that yields a good classification accuracy rate with a small number of features. Let 'p' be the number of hyperboxes after pruning and 'd' is the number of dimensions of each hyperbox in the particular time interval $[t_1, t_2]$. The chromosome (S) of the PSO is a binary string that represents a solution comprising all the possible open hyperboxes as follows:

$$S = \left\{ D_1^1, D_2^1, \ldots\ldots, D_d^1, D_1^2, D_2^2, \ldots, D_d^2, \ldots, D_1^p, D_2^p, \ldots, D_d^p \right\} \text{ during } [t_1, t_2]. \quad (18)$$

During a particular time interval $[t_1, t_2]$, a value '1' for the member in 'S' indicates that the membership grade is calculated from the minimum and maximum points of a closed hyperbox. On the other hand, a value of '0' for the member in S during $[t_1, t_2]$ indicates that the dimension is a 'dont care' dimension, and its membership value is set to 1. The number of possible IF–THEN rules generated is n^d, where d is the dimension of the input features and n is the number of fuzzy partitions in each dimension. This linguistic set of fuzzy partitions of n is set to six as suggested in Ishibuchi *et al* (1997) to indicate 'small,' 'medium small,' 'medium,' 'medium large,' 'large,' and 'dont care'. This leads to the 'curse-of-dimensionality' problem when a high-dimensional data set is used. In TFMM-PSO, the length of S and the PSO search space are subject to the number of hyperboxes, which is controlled by pruning in time. In other words, the 'curse-of-dimensionality' problem in TFMM-PSO can be mitigated by pruning.

To achieve the objectives of maximizing classification accuracy and minimizing the number of input features, the PSO fitness function used an extension of Ishibuchi *et al* (1997) with temporal constraints.

$$\text{Maximize } (S, t) = W_{NCP}.NCP(S, t) - W_s, t.|S| \text{ } Subject \text{ } to \text{ } t_1 \leq t \leq t_2, \quad (19)$$

where, $NCP(S, t)$ is the number of correctly classified patterns by the selected hyperbox set at time t, $|S|$ is the number of features used, W_{NCP} and W_S are positive weights, and $0 < W_S \ll W_{NCP}$. The genetic operation implemented consists of the following steps.

Stage 1: Initialize the population strings N_{pop} for the time interval $[t_1, t_2]$. The strings are generated by assigning '0' for 'dont care' features and 1 for other features.

Stage 2: The current best fitness achieved by particle S is set as St_1 *best*. The selection probability $P(S, t)$ of a string S in a population $\Psi(t)$ is specified as

$$P(S, t) = \frac{\{f(S, t) - f_{min}(\Psi(t))\}}{\sum_{S' \epsilon \Psi} \{f(S, t) - f_{min}(\Psi(t))\}}, \quad (20)$$

where,

$$f_{min}(\psi(t)) = min\{f(S, t)|S, t \in \psi(t)\}. \quad (21)$$

Stage 3: Evaluate the desired optimization fitness function *fp(t)* for each particle as the Mean Square Error (MSE) over a given data set.

Stage 4: Compare the evaluated fitness value $f_S(t)$ of each particle with its St_1best value. If *fs(t)* $< St_1$ *best* then St_1 *best* $= fs(t)$ and $best_tS = t$, *tS* is the current coordinates of particle *S*, and $best_tS$ is the coordinates corresponding to particle S*'s* best fitness so far.

Stage 5: The objective function value is calculated for new positions of each particle. If a better position is achieved by an agent, St_1 *best* value is replaced by the current value. As in Stage 1, t_2best value is selected among St_1 *best* values. If the new t_2best value is better than previous t_2best value, the t_2best value is replaced by the current t_2best value and this value is stored. If $fS(t) < t_2best$ then $t_2best = S$, where t_2best is the particle having the overall best fitness over all particles in the swarm.

Stage 6: Change the velocity and location of the particle according to Eqs. 20 and 21, respectively.

$$S_v, t_1 = 1 \rightarrow S_v, t_1 = -1 \text{ with probability } P_y$$
$$L_v, t_1 = S_v, t_1 \times t_2 \text{ with probability } P_y.$$

Stage 7: Fly each particle *p* according to Eq. 21.

Stage 8: If the maximum number of predetermined iterations (epochs) is exceeded, then stop; otherwise, Loop to step 3 until convergence. In this work, 50 populations of weights were evolved for 100 generations.

The hyperbox set evolved and selected by the PSO is used for rule extraction as follows:

4.2a *Extracting temporal fuzzy IF–THEN rules*: For rule extraction, each hyperbox is transformed into one fuzzy temporal rule. The rule extraction procedure starts by quantizing first the minimum and maximum values of each input feature during the interval $[t_1, t_2]$.

For quantization, the interval $[t_1, t_2]$ with $t_2 - t_1 = Q$ is divided into q intervals, and the input feature is assigned to the quantization points evenly with '1' at each of the end points based on as (Carpenter & Tan 1995).

$$A_q(t) = \frac{(q - 1)}{(Q - 1)}. \tag{22}$$

The temporal fuzzy IF–THEN rules extracted are in the following format:

Rule R_j : IF x_{p1}, t is A_q, t and $\dots x_{pn}, t,$ is $A_q, t,$

THEN x_p, t is class C_j, t with $CF = CF_j, t \quad j = 1, 2, \dots, N, t_1 \le t \le t_2,$

where, *N* is the number of hyperboxes, $x_p = (x_{p1}, \dots, x_{pn})$ is an *n*-dimensional pattern vector, $A_q(t)$ is the antecedent value, and CF_j, t is the confidence factor of the jth hyperbox.

4.2b *Rule firing*: The rules extracted from the training phase are stored in the rule base. During the testing phase, rules are selected and fired using forward chaining inference for making effective decisions. The rule manager developed into this work uses fuzzy temporal constraints in rule matching and hence the decisions on medical datasets are provided with necessary accuracy. Rule execution is carried out during the testing phase for checking the input data effectively.

5. Results and discussion

Our proposed pattern classification approach have been experimented and tested by different medical dataset from the UCI Machine Learning Repository Data Set (Murphy & Aha 1995). Due to the specific features of clinical data, we present a brief description of their characteristics.

5.1 *Data description of input dataset*

5.1a *Diabetes data set*: This data set is also referred from the UCI Machine Learning Repository Data Set (Murphy & Aha 1995), which is prepared by Michael Kahn, Washington University, St. Louis. The DIABETES data sets in this directory are provided for use in 1994, AI in Medicine symposium submissions. Diabetes files consist of four fields per record such as Date, Time, Code and Value. The Code field is deciphered in Murphy & Aha (1995) as shown in table 1.

There are 70 text files in the UCI repository and each file is assigned to one patient and stores her or his disease history. The patients are insulin deficient. This disease is manifested by many so-called metabolic effects, the main one being high blood glucose, which can be detected by measurements. The goal of the therapy is to bring the average blood glucose level of the patient down to the normal range. However, in spite of insulin injections, due to many reasons, blood glucose concentrations around the mean can increase to dangerous levels, so the concentration should be continuously monitored. The exercises performed by patients can cause blood glucose concentrations to decrease with variable time delays of this effect. Another factor is that a large meal can cause longer and higher elevations of blood glucose. Controlling glucose in the blood is a challenging task. The effects of the insulin dose vary with time and can be only approximately estimated by doctors. The frequency of injections and blood glucose measurements are obviously very important for the patient.

Table 1. Code field of diabetes data set.

---

33 = Regular insulin dose
34 = NPH insulin dose
35 = UltraLente insulin dose
48 = Unspecified blood glucose measurement
57 = Unspecified blood glucose measurement
58 = Pre-breakfast blood glucose measurement
59 = Post-breakfast blood glucose measurement
60 = Pre-lunch blood glucose measurement
61 = Post-lunch blood glucose measurement
62 = Pre-supper blood glucose measurement
63 = Post-supper blood glucose measurement
64 = Pre-snack blood glucose measurement
65 = Hypoglycemic symptoms
66 = Typical meal ingestion
67 = More-than-usual meal ingestion
68 = Less-than-usual meal ingestion
69 = Typical exercise activity
70 = More-than-usual exercise activity
71 = Less-than-usual exercise activity
72 = Unspecified special event

Table 2. Inputs attributes list.

Selected attributes and their depicted values as in Cleveland database
Patient id: patient identification number
Age: age in years
CP: (chest pain type) 1-type1 angina, 2-type 2 angina, 3-non angina, 4-asymptomatic
Trestbps: resting blood pressure
Restecg: resting electrocardio graphic results 0-normal, 1- ST-T wave abnormal, 2-left ventricular hypertrophy
Thalach: maximum heart rate achieved
Exang: exercise induced angina. 1-yes, 0-No
Oldpeak: ST depression induced by exercise relative to rest
Slope: the slope of the peak exercise ST segment. 1-unsloping, 2-flat,3-downsloping
CA: number of major vessels coloured by fluoroscopy values 0–3
Fbs: fasting blood sugar. 1 > 120 mg/dl
0 < 120 mg/dl

5.1b *Heart data set*: Clinical dataset contains attributes that are irrelevant, redundant and similar and hence it has to be eliminated or clustered in order to build a more efficient model. The sample Cleveland database consists of a number of 303 patient records which includes a set of 76 attributes of numeric, categorical and different data types. Table 2 shows the list of attributes for heart dataset.

The main focus of experimental analysis is with heart and diabetic data sets. In addition to that, some other data sets from the UCI Machine Learning Repository Data Set such as Glass, Wine and PID are used to test the classification accuracy and the number of rules generated by TFMM-PSO system. The medical dataset used in this work has the third dimension namely the time. In addition to the tuples and attributes as the first and second dimension, temporal features are provided through time stamping and hence the data set considered is of higher dimension.

5.2 *Experimental evaluation of TFMM-PSO classifier*

In this work, three different networks namely FMM (Simpson 1992), FMM-GA (Quteishat *et al* 2010) and TFMM-PSO have been constructed and evaluated with ten-fold cross-validation using the various subsets of UCI Machine Learning Repository Data Sets. All the data samples were divided into ten subsets, where nine of them were used for training, and the remaining one was used for testing. As TFMM-PSO required a prediction data set for rule extraction, 10% of the training data samples were taken to form the prediction data set. This procedure was repeated ten times until each of the ten subsets was evaluated. To quantify the performance, the experiment was repeated ten times (each with randomized data sequences), and the results were averaged.

In all the experiments, the pruning threshold was set to 0.7, i.e., hyperboxes with *CF* smaller than 0.7 were pruned. The objective is to maintain a small number of hyperboxes, each with a reasonably high *CF* with temporal constraints. Open hyperboxes were generated after pruning. The resulting hyperboxes were sent to the PSO rule extractor for evolution. Finally, fuzzy IF–THEN rules were extracted from the selected hyperboxes by quantization. To minimize the effort in fine-tuning the network parameters, the PSO parameters were used in all the experiments were set as follows:

(i) Population size Npop $= 100$;

 (ii) Rate of change of velocity $= 0.9$;
(iii) Location change rate $= 0.1$;
(iv) Number of hyperboxes replaced in each population $= 20$;
 (v) Stopping condition $= 1000$ generations, or no change in the maximization function value for 2 consecutive iterations.

Table 3 shows the classification results (the percentages of test accuracy rates and the number of hyperboxes) from FMM, pruned FMM, and TFMM-PSO at 5% level. In general, TFMM performed better than pruned FMM. As pruning with temporal constraints eliminated a large number of hyperboxes from the network, the numbers of hyperboxes in FMM were at least 91% less than those in FMM. Table 3 also shows the accuracy rates of TFMM-PSO, which is better than FMM and pruned FMM. With the help of the PSO, TFMM-PSO has improved the performance of pruned FMM. As pruned FMM and TFMM-PSO used the same numbers of hyperboxes for classification, the PSO was proven useful for providing better performance.

From table 4, it can be observed that the proposed TFMM-PSO has taken less time when it is compared with FMM. This decrease in time is due to use of optimization with PSO.

The rules learnt using decision trees and random forest are suitable to classify the complete data. However, the medical dataset used in this work evolves continuously with respect to time and it is fuzzy in nature. Therefore, uncertainty has to be considered in decision making. The classifier proposed in this work can handle uncertain and incomplete medical data effectively by applying fuzzy temporal constraints. It also uses instant comparison and interval comparison operators with fuzzy membership function in order to perform effective classification. Hence, it is essential to have a new classifier instead of applying decision rules and random forests.

5.3 *Experimental evaluation on fuzzy temporal rules*

5.3a *Diabetics data*: Diabetic patient data set can be broadly classified into two decision classes such as diabetic or non-diabetic. The main classification is performed with insulin dosage to maintain the blood glucose level. The numeric data set can be fuzzified into five linguistic term set and the rule extractor helps to derive fuzzy temporal IF-THEN rules. When classifiers are used, they apply the rules based on the classification algorithm. However, generation of rules help to apply fuzzy and temporal constraints effectively. Moreover, when rules are used, it is possible to fire both active and passive rules. Rules with priority and rules with exception can also be added for effective decision making. Rules can be used to perform deductive inference either using forward chaining or backward chaining control flow mechanism based on the type of data. Rules matching algorithms are able to generate discriminate networks and are able to perform rule execution even when rules chaining occur. Therefore, application of fuzzy and temporal rules helps in making effective decisions.

A sample set of rules for diabetic dataset is presented below:

 Rule 1: *Dont Care*

> **IF** the patient's blood sugar value is below 120
> **THEN** the patient does not require insulin dosage.

 Rule 2: *Low*

> **IF** the patient's blood sugar value is in the range of 120 to 150

THEN the patient requires insulin dosage and he/she is suffering from diabetes slightly.

Rule 3: *Medium*

IF the patient's blood sugar value is in the range of 150 to 200
THEN the patient requires insulin dosage and he/she is suffering from diabetes slightly.

Rule 4: *High*

IF the patient's *insulin value is 200–250*
THEN the patient requires insulin dosage and he/she is suffering from diabetes severely.

Rule 5: *Very High*

IF the patient's insulin value is *above 250*
THEN the patient requires insulin dosage and he/she is suffering from diabetes severely.

The rules for severely diabetic patient with NPH 12 units before breakfast, 6 units before dinner and regular insulin:

Rule 1: **IF** blood glucose is 70–100 mg/dl **THEN** it requires 4 units of regular insulin dosage during pre-breakfast and pre-dinner.

Rule 2: **IF** blood glucose is 90–145 mg/dl **THEN** it requires 4–5 units of regular insulin dosage during pre-breakfast and pre-dinner.

Rule 3: **IF** blood glucose is 135–190 mg/dl **THEN** it requires 5–6 units of regular insulin dosage pre-breakfast and pre-dinner.

Rule 4: **IF** blood glucose is 185–245 mg/dl **THEN** it requires 6–7 units of regular insulin dosage pre-breakfast and pre-dinner.

Rule 5: **IF** blood glucose is 235–290 mg/dl **THEN** it requires 7–8 units of regular insulin dosage pre-breakfast and pre-dinner.

Rule 6: **IF** blood glucose is 251–300 mg/dl **THEN** it requires 8–9 units of regular insulin dosage pre-breakfast, 1 unit of insulin pre-launch and 8 unit of insulin pre-dinner, and 1 unit at bed-time.

Rule 7: **IF** blood glucose > 300 **THEN** 9 units of insulin dosage pre-breakfast, 2 units pre launch, 9 units pre-dinner, and 2 units at bed-time.

5.3b *Heart dataset*:   Heart data includes 4 decision classes according to the type of chest pain. In the case of heart dataset is possible to have a fourth dimension namely spatial features. In such a scenario, the spatio-temporal constraints can be provided in the form of rules in order to perform decision making from high dimensional spatio-temporal data.

In TFMM-PSO, the rules have been extracted by normalizing the input values to a score. The rule extractor is working based on the score assigned to each of the risk assessing factor as shown in table 5.

The probability of the severity of heart disease is diagnosed with respect to the score obtained from the risk assessing parameters. Since the score values are numeric they are fuzzified to the range [0, 1] for classifying the type and severity of heart disease based on the following fuzzy rules.

Rule 1: *Dont Care*

    IF Age is *below 45*
        Chestpain is asymptomatic
        No exang
        Normal restecg
        Slope is from 0 mm to 1 mm
    THEN the severity of heart disease is Nil.

Rule 2: *Low*

    IF Age is between 45 and 55
        Chestpain is non-angina
        No exang
        Abnormal restecg
        Slope is from 1 mm to 2 mm
    THEN the severity of heart disease is low.

Rule 3: *Medium*

    IF Age is between 55 and 65
        Chestpain is typical angina
        History of exang
        Abnormal restecg
        Slope is from 1 mm to 2 mm
    THEN the severity of heart disease is medium.

Rule 4: *High*

    IF Age is between 60 and 70
        Chestpain is typical angina
        History of exang
        Abnormal restecg
        Slope is from 2 mm to 3 mm
    THEN the severity of heart disease is high.

Rule 5: *Very High*

    IF Age is above 70
        Chestpain is atypical angina
        History of exang
        Abnormal restecg
        Slope is from 2 mm to 3 mm
    THEN the severity of heart disease is very high.

Table 3. Classification results of FMM, FMM-GA and TFMM-PSO with $\theta = 0.05$.

Data set	FMM (Simpson 1992)		FMM-GA (Quteishat *et al* 2010)		TFMM-PSO	
	Test accuracy (%)	Number of hyperboxes	Test accuracy (%)	Number of hyperboxes	Test accuracy (%)	Number of hyperboxes
Diabetic	71.2	124	76.4	14	81.3	10
Heart	74.2	135	72.1	12	82.05	12
Glass	69.07	182	79.39	7	80.17	7
Wine	96.85	160	99.78	14	99.88	14
PID	68.42	190	89.74	37	90.44	37

Table 4. Computational time for the bench mark data sets.

Data set	Computational time for stage one (FMM learning) (sec)	Computational time for stage one (TFMM learning) (sec)
Diabetic	23.72	23.02
Heart	22.52	21.3
Glass	24.96	24.56
Wine	10.83	10.41
PID	639.72	631.52

Table 5. Score assigned for each risk assessing factor.

Factor	Value	Score
Age	<50	0
	50–60	1
	60–70	2
	>70	3
ChestPain	asymptomatic	0
	non angina	1
	typical angina	2
	atypical angina	3
Restecg	Normal	0
	Abnormal	1
History of Exang	Not present	0
	present	1
Slope	0–1 mm	1
	1–2 mm	2
	2–3 mm	3

Table 6. Comparison of number of rules generated for FMM-GA and TFMM-PSO.

Data set	Threshold for hyper box pruning $(0 \leq \theta \leq 1)$	FMM-GA (Quteishat *et al* 2010)		TFMM-PSO	
		Average number of rules	Classification accuracy	Average number of rules	Classification accuracy
Diabetes	0.2	10.3	0.62195	10.01	0.67423
	0.4	9.35	0.7082	9.2857	0.70337
	0.5	8.55	0.6832	8.2857	0.68539
	0.6	7.85	0.7448	6.8	0.7448
	0.7	6.4	0.7962	6.02	0.7962
Heart disease	0.2	16.5	0.7103	13.5	0.7305
	0.4	15.05	0.73	12.01	0.742
	0.5	13.6	0.7502	10.25	0.7702
	0.6	12.5	0.7923	10.01	0.8257
	0.7	11.25	0.8125	9.8	0.8475

The above rules which are inferred by TFMM-PSO system suggest the probability of incidents causing severity of heart disease with the selected set of risk assessment factors from the data set. The same can be applied to derive the different set of rules with respect to other factor also.

Table 6 shows the comparative analysis of the proposed TFMM-PSO with the existing technique. From table 6, it is seen that this proposed TFMM-PSO provides better performance than FMM-GA with respect to the number of rules and classification accuracy. This is because in the proposed TFMM-PSO, a minimal rule set is obtained by applying PSO experimental results and it is proved that the proposed system provides better classification accuracy.

6. Conclusion and future enhancement

A pattern classification and rule extraction system has been proposed in this paper. The system comprises a Temporal FMM network and a PSO rule extractor. The proposed TFMM-PSO system has a number of novel properties. First, a modified FMM network that uses a pruning procedure to eliminate hyperboxes with low confidence factors has been proposed. Second, the proposed system uses a PSO rule extractor to generate and select a compact rule set with high classification accuracy. Third, a rule extraction procedure to extract fuzzy IF–THEN rules with 'dont care' antecedents has been introduced. The performances of TFMM-PSO have been evaluated using five benchmark problems and a real medical diagnosis task. The results have been compared with pruned FMM, as well as other methods published in the literature. The outcomes have shown that the classification performances of TFMM-PSO are better than those from other systems. One of the main advantages of TFMM-PSO is the comprehensibility of the extracted rules. Future works in this direction can be the proposal of a temporal logic for better reasoning.

References

Adlassnig K P, Combi C, Das A K, Keravnou E T and Pozzi G 2006 Temporal Representation and Reasoning in Medicine: Research Directions and Challenges. *Artificial Intelligence in Medicine* 38: 101–113

Campos P G, Oliveira E M J, Ludermir T B and Araujo A F R 2004 MLP Networks for Classification and Prediction with Rule Extraction Mechanism. In *Proc. IEEE Int. Joint Conf. Neural Netw.* 2: 1387–1392

Carpenter G and Tan A 1995 Rule extraction: From neural architecture to symbolic representation. *Connection Sci.* 7(1): 3–27

Eberhart R C, Simpson P and Dobbins R 1996 Computational Intelligence PC Tools. *Academic Press*

Gui H and Qiao J 2012 Adaptive Computation Algorithm for RBF Neural Network. *IEEE Transactions on Neural Networks and Learning Systems* 23(2): 342–347

Han H and Qiao J 2010 A Self-Organizing Fuzzy Neural Network Based on a Growing-and-Pruning Algorithm. *IEEE Transactions on Fuzzy Systems* 18(6): 1129–1143

Ishibuchi H, Murata T and Turksen I B 1997 Single-objective and Two Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems. *Fuzzy Sets Syst.* 89(2): 135–150

Kennedy J and Eberhart R C 1995 Particle Swarm Optimisation. *Proceedings of IEEE International Conference on Neural Networks*, IV, p. 1942–1948

Kennedy J and Eberhart R C 2001 Swarm Intelligence. *Morgan Kaufman Publishers*

Kim H J and Yang H S 2005 A Weighted Fuzzy Min-Max Neural Network and its Application to Feature Analysis. *Advances in Natural Computation* (Lecture Notes in Computer Science), 3612 New York: Springer-Verlag 1178–1181

Laxman S and Sastry P S 2006 A Survey of Temporal Data Mining. *Sadhana* 31(2): 173–198

Meamarzadeh H, Khayyambashi M R and Saraee M H 2009 Extracting Temporal Rules for Medical Data. *Int. Conf. Comput. Technol. Development* 327–331

Murphy P M and Aha D W 1995 UCI Repository of Machine Learning Databases, (Machine-Readable Data Repository). Irvine, CA: *Dept. Inf. Comput. Sci., University of California*

Nowicki Robert 2009 Rough Neuro-Fuzzy Structures for Classification With Missing Data. *IEEE Transactions on Systems, Man, Cybernetics* 39(6): 1334–1347

Oong T H and Isa A 2011 Adaptive Evolutionary Artificial Neural Networks for Pattern Classification. *IEEE Transactions on Neural Networks* 22(11): 1823–1836

Quteishat A, Lim C P and Tan K S 2010 A Modified Fuzzy Min–Max Neural Network with a Genetic-Algorithm-Based Rule Extractor for Pattern Classification, *IEEE Transactions on Systems. Man, and Cybernetics—Part A: Systems and Humans* 40(3): 641–650

Setiono R, Leow W K and Zurada J M 2002 Extraction of Rules from Artificial Neural Networks for Nonlinear Regression. *IEEE Transactions on Neural Networks* 13(3): 564–577

Simpson P K 1993 Fuzzy Min-Max Neural Network-part II: Clustering. *IEEE Trans. Fuzzy Syst.* 1(1): 32–45

Simpson P K 1992 Fuzzy Min-Max Neural Networks-part I: Classification. *IEEE Trans. Neural Netw.* 3(5): 776–786

Taylor B J and Darrah M A 2005 Rule Extraction as a Formal Method for the Verification and Validation of Neural Networks. In *Proc. IEEE IJCNN,* pp. 2915–2920

Wai R J and Lee J D 2008 Adaptive Fuzzy-Neural-Network Control for Maglev Transportation System. *IEEE Trans. Neural Networks* 19(1): 54–70

Xiuju F and Lipo W 2002 Rule Extraction Using a Novel Gradient-based Method and Data Dimensionality Reduction. In *Proc. Int. Joint Conf. Neural Netw.*1275–1280

Zadeh L A 1965 Fuzzy sets. *Inf. Control* 8(3): 338–353

Zhang H G, Li M, Yang J and Yang D D 2009 Fuzzy Model-based Robust Networked Control for a Class of Nonlinear Systems. *IEEE Trans. Syst. Man, Cybern.–Part A: Syst. Humans* 39(2): 437–447

Zhang Huaguang, Liu Jinhai, Ma Dazhong and Wang Zhanshan 2011 Data-Core-Based Fuzzy Min–Max Neural Network for Pattern Classification. *IEEE Transactions on Neural Networks* 22(12): 2339–2352