

# Patient admission prediction using a pruned fuzzy min–max neural network with rule extraction

Jin Wang · Chee Peng Lim · Douglas Creighton · Abbas Khorsavi · Saeid Nahavandi · Julien Ugon · Peter Vamplew · Andrew Stranieri · Laura Martin · Anton Freischmidt

Received: 10 February 2014 / Accepted: 16 May 2014 / Published online: 5 June 2014  
© Springer-Verlag London 2014

**Abstract** A useful patient admission prediction model that helps the emergency department of a hospital admit patients efficiently is of great importance. It not only improves the care quality provided by the emergency department but also reduces waiting time of patients. This paper proposes an automatic prediction method for patient admission based on a fuzzy min–max neural network (FMM) with rules extraction. The FMM neural network forms a set of hyperboxes by learning through data samples, and the learned knowledge is used for prediction. In addition to providing predictions, decision rules are extracted from the FMM hyperboxes to provide an explanation for each prediction. In order to simplify the structure of FMM and the decision rules, an optimization method that simultaneously maximizes prediction accuracy and minimizes the number of FMM hyperboxes is proposed. Specifically, a genetic algorithm is formulated to find the

optimal configuration of the decision rules. The experimental results using a large data set consisting of 450740 real patient records reveal that the proposed method achieves comparable or even better prediction accuracy than state-of-the-art classifiers with the additional ability to extract a set of explanatory rules to justify its predictions.

**Keywords** Patient admission prediction · Fuzzy min–max neural network · Genetic algorithm · Rule extraction

## 1 Introduction

For more than a decade, an increasing number of patients visit hospital emergency departments daily. This poses a great challenge for emergency departments to manage limited medical resources (e.g., available hospital beds and

---

J. Wang (✉) · C. P. Lim · D. Creighton · A. Khorsavi · S. Nahavandi  
Center for Intelligent Systems Research, Deakin University,  
Waurin Ponds, VIC 3217, Australia  
e-mail: jay.wangjin@gmail.com

C. P. Lim  
e-mail: chee.lim@deakin.edu.au

D. Creighton  
e-mail: douglas.creighton@deakin.edu.au

A. Khorsavi  
e-mail: abbas.khorsavi@deakin.edu.au

S. Nahavandi  
e-mail: saeid.nahavandi@deakin.edu.au

J. Ugon · P. Vamplew · A. Stranieri  
School of Science, Information Technology and Engineering,  
Federation University, Ballarat, VIC 3350, Australia  
e-mail: j.ugon@federation.edu.au

P. Vamplew  
e-mail: p.vamplew@federation.edu.au

A. Stranieri  
e-mail: a.stranieri@federation.edu.au

L. Martin  
Redesign and Innovation Unit, Ballarat Health Services,  
Ballarat, VIC 3350, Australia

A. Freischmidt  
Department of Health, Commission for Hospital Improvement,  
Ballarat, VIC, Australia  
e-mail: anton.freischmidt@health.vic.gov.au

doctors) for patients seeking emergency care. Traditionally, patient admission is manually determined by medical practitioners. This manual operation has several drawbacks. Firstly, comparing with the large number of patients, medical practitioners, especially the ones with extensive experience, are very limited. Secondly, the manual admission process is always time-consuming and labor-intensive. Finally, the manual admission process heavily relies on the knowledge and experience of medical practitioners, which sometimes entails a subjective reasoning process. Therefore, a decision support system that is able to automatically and accurately predict patient admission is very valuable for real clinical application and is highly sought after by emergency departments. It is even more valuable if the decision support system could justify its prediction pertaining to patient admission objectively. Therefore, the decision support system helps emergency departments to deliver timely emergency care, improve quality of care, and reduce waiting time of presented patients.

Although automatic prediction of patient admission has great potential applications in clinical environments, few works have been conducted to investigate this problem [1–5]. Automatic prediction of patient admission can be cast as a pattern recognition and classification problem that can be addressed by various types of classifiers such as artificial neural networks (ANNs) [6], Adaboost [7, 8], Random Forest [9], Classification And Regression Tree (CART) [10], Support Vector Machine (SVM) [8].

Among them, the fuzzy min–max (FMM) neural network [11, 12] is a promising model to undertake pattern recognition and classification tasks in medical prediction and diagnosis [13]. As a reliable and promising prediction model, FMM has several desirable properties. Firstly, FMM is an online learning model. It incrementally learns information from new coming training samples (i.e., patient information) without forgetting previously learned information. Furthermore, FMM is a nonlinear classifier, i.e., it is able to learn decision boundaries to separate classes that consist of different sizes and shapes. In addition, FMM provides a soft decision through its membership function, which is tolerant of imprecision, partial truth, and uncertainty. This soft decision can be further converted into a hard decision by setting a decision threshold.

The FMM neural network forms a group of hyperboxes with fuzzy sets theory to represent the information learned from training data. A fuzzy hyperbox is an  $n$  dimension box ( $n$  is the dimension of input patterns) that is defined by a set of minimum and maximum points. The goodness-of-fit of an input pattern to a hyperbox is determined by the degree of membership to the corresponding hyperbox. The hyperbox size is controlled by a user-defined parameter. A larger hyperbox size leads to a larger coverage of the pattern

space. The resulting hyperbox can contain a larger number of data samples, which reduces the complexity of the FMM neural network. By contrast, a smaller hyperbox can only cover a smaller region that contains fewer data samples, which in turn increases the complexity of FMM. Therefore, the hyperbox size is of importance to the FMM neural network. An FMM model with large hyperboxes neglects salient and discriminative information of input patterns, while one with small hyperboxes over-fits the training data.

A number of extensions have been proposed to improve the performance of FMM. For instance, Nandedkar and Biswas [14] extended the standard FMM network to cope with granular data by representing granular data using hyperbox fuzzy sets. Seera et al. derived a hybrid model of FMM and CART (Classification and Regression Tree) to detect and classify fault conditions of induction motors in both offline [15] and online [16] learning modes, as well as with hardware implementation [17]. The proposed FMM–CART model exploits the advantages of both FMM and CART for data classification and rule extraction. Zhang et al. [18] added a kind of overlapped neuron with new membership function based on data core to the FMM neural network to characterize the overlapping regions of hyperboxes in different classes. Since the characteristics of the data and influence of noise are simultaneously captured by the membership function, the data core-based FMM is more robust to noise. However, one drawback of this method is that it cannot correctly classify datasets consisting of a large proportion of data samples that are located in overlapping regions. In order to address this problem, the work in [19] extended an FMM in a multi-level structure to classify patterns. The multi-level fuzzy min–max neural network classifier (MLF) learns smaller hyperboxes in different levels to classify the samples that are located in boundary regions. MLF is able to learn nonlinear boundaries and is more precise and robust.

Although the FMM neural network is a reliable and promising prediction model, it lacks an important ability for medical prediction applications, i.e., decision rule extraction. Decision rules are important in medical prediction applications because they explain why and how the neural network, which is commonly criticized as a “black-box” [20], makes a certain decision for an input pattern. Without the ability to explain and justify the decision output, it is hard for medical practitioners to accept and trust the network prediction in medical applications. Furthermore, since decision rules provide an explicit relationship between features and decision classes, it is very helpful to identify the salient and discriminative feature patterns in the input data sets. In addition, decision rules can improve the robustness of a neural network by identifying conditions under which the prior learning of the network cannot be generalized.

A number of methods have been proposed to extract decision rules from neural networks. Three techniques are proposed in [21] to extract rules from neural networks. The first one is able to extract a group of binary decision rules from any type of neural networks, while the other two are specific to feed-forward neural networks with a single hidden layer with nodes using sigmoid activation functions. Another work in [22] translates the system state into a set of rules by quantizing the continuous network weights. In order to simplify the network structure, the recognition categories and weights of the network are excessively removed. Most of the investigations focus on extracting rules from neural networks for classification. The work in [23] proposes a method to extract rules from trained neural networks for regression. Allowing each rule to correspond to a subregion of the input space, the rules are extracted by approximating the activation function at each hidden neuron. Since decision rules extracted from neural networks can be complex in real applications, some methods to simplify the extracted rules have been explored. For instance, the work in [24] employs a genetic algorithm to select a small number of significant rules. This method is further improved by introducing a “don’t care” attribute in the fuzzy *if-then* rules [25, 26]. The features that have limited or no impact on the final decision rule are defined as “don’t care” features. Some works have also been conducted to extract decision rules from FMM [13, 27, 28]. Each hyperbox of FMM is transformed into a descriptive rule by quantizing the minimum and maximum values of input features.

In order to extract a compact set of decision rules, the works in [13, 27, 28] propose to prune FMM using a confidence factor (CF). The CF identifies hyperboxes that are frequently used and those that are rarely used but highly accurate. Each FMM hyperbox is tagged with a CF. Hyperboxes that have CF scores lower than a user-defined threshold are pruned. The resulting FMM network reduces the complexity of original FMM and its decision rules. However, it cannot guarantee that hyperboxes of original FMM are optimally pruned in terms of prediction accuracy. Furthermore, it is difficult to determine the optimal threshold of CF.

In order to address these problems, this paper proposes a genetic algorithm (GA)-based pruning scheme to optimally prune the hyperboxes of FMM. The resulting model is

known as FMM-GA. Unlike the method to prune hyperboxes based on CF (FMM-CF), FMM-GA defines an objective function to find the optimal configuration of hyperboxes. Specifically, the objective function simultaneously maximizes prediction accuracy and minimizes the number of hyperboxes in FMM-GA. The advantage of the proposed FMM-GA model is that it is able to achieve the highest prediction accuracy with the smallest number of hyperboxes. Therefore, the number of decision rules extracted is in a parsimonious state, without sacrificing prediction accuracy. Figure 1 outlines the general flowchart of the proposed method.

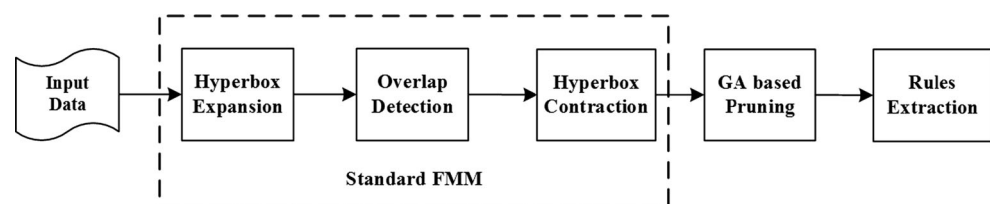
The motivation of this paper was to develop a patient admission prediction model based on FMM, which is able to achieve high prediction accuracy and simultaneously produce a small number of decision rules. The main contribution of the work is twofold: (i) an objective function that simultaneously maximizes prediction accuracy and minimizes the number of hyperboxes is proposed to optimize the standard FMM network and (ii) a series of experiments to evaluate the performance of the proposed FMM-GA model using a large database with real (anonymous) patient admission records.

The rest of the paper is organized as follows. Section 2 describes FMM in detail. Section 3 explains the details of hyperbox pruning based on GA. Rules extraction is explained in Sect. 4. Section 5 presents the experimental setup, results, and discussion on the patient admission dataset, while Sect. 6 compares the experimental results on three benchmark datasets from the UCI machine-learning repository. Conclusion and future works are given in Sect. 7.

## 2 Fuzzy min–max neural network

The FMM neural network is an online incremental learning model, which is based on hyperbox fuzzy sets. In general, FMM forms the hyperboxes with fuzzy sets to represent knowledge learned by the model from training data. One benefit of FMM is that it is able to refine its current model online as new training samples are available. The learning procedure of FMM consists of a series of expansion and contraction processes that fine-tune the hyperboxes to establish decision boundaries among different classes. The

**Fig. 1** A general procedure of the proposed method



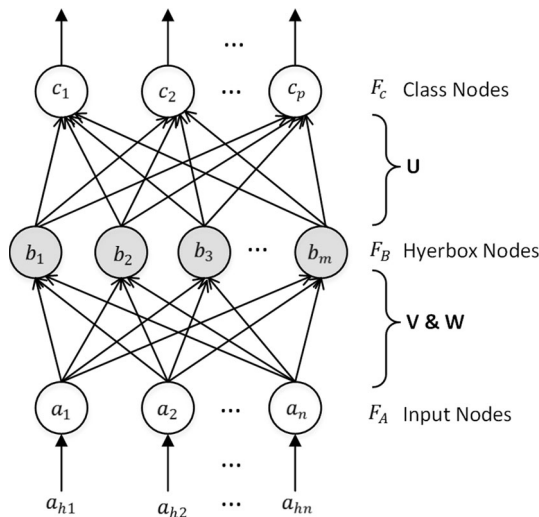


Fig. 2 Structure of FMM

FMM operations are described in the following section, and the details are in [11].

Figure 2 depicts the structure of FMM, which consists of three layers. Layer  $F_A$  is the input layer. Each node in layer  $F_A$  represents a feature of the input pattern. As such, the number of nodes in layer  $F_A$  equals to the dimension of the input pattern. Layer  $F_C$  is the output layer, which has nodes equal to the number of target classes. The output of a node in layer  $F_C$  represents the degree to which an input pattern fits within a target class. For hard-decision classification, the node that produces the highest value corresponds to the predicted target class. Hidden layer  $F_B$  is also known as the hyperbox. Each node in layer  $F_B$  represents a hyperbox fuzzy set. The connections between  $F_A$  and  $F_B$  represent the min–max points of hyperboxes, and the  $F_B$  transfer function is the hyperbox membership function. The minimum and maximum points of the hyperboxes are stored in matrices  $V$  and  $W$ , respectively. The connections between  $F_B$  and  $F_C$  are binary values, which are stored in matrix  $U$  [11].

The hyperboxes are confined between zero and one. Therefore, the input patterns are normalized and projected into an  $n$ -dimensional unit hypercube ( $I^n$ ). The membership function of a hyperbox that describes the degree to which a pattern located in the hyperbox is defined based on the minimum and maximum points of the hyperbox. A hyperbox ( $B_j$ ) can be defined as [11]:

$$B_j = \{A, V_j, W_j, f(A, V_j, W_j)\} \quad \forall A \in I^n, \quad (1)$$

where  $A = (a_1, a_2, \dots, a_n)$  is the input pattern;  $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$  and  $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$  are the minimum and maximum points of the hyperbox, respectively; and  $f(A, V_j, W_j)$  is the hyperbox membership function.

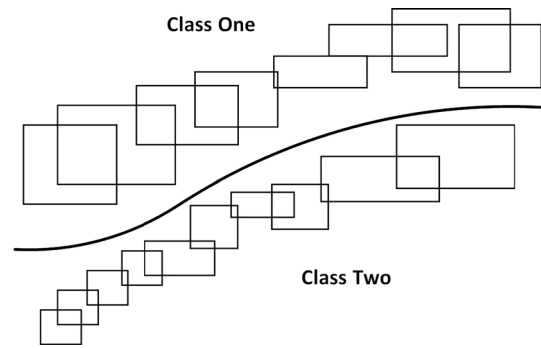


Fig. 3 A demonstration of hyperboxes and the corresponding decision boundary learned by FMM for a binary class problem [11]

The membership function for the  $j$ th hyperbox, i.e.,  $b_j(A_h)$ , where  $0 \leq b_j(A_h) \leq 1$ , measures the degree to which the  $h$ th input pattern ( $A_h$ ) fits in hyperbox  $B_j$ . The membership function is defined as:

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

where  $\gamma$  is a sensitivity parameter that regulates how fast the membership value decreases with respect to the distance between  $A_h$  and  $B_j$ . Based on the definition of fuzzy set in FMM, the combined fuzzy set that discriminates the  $K$ th pattern class ( $C_k$ ) is defined as:

$$C_k = \bigcup_{j \in K} B_j, \quad (3)$$

where  $K$  is the index set of the hyperboxes corresponding to class  $K$ . The main learning procedure of FMM is to find and fine-tune the class boundaries. Figure 3 shows an example of two-dimensional hyperboxes and the corresponding decision boundary learned by FMM for a binary class problem. The FMM learning algorithm is summarized as follows.

- (a) *Expansion:* This step identifies hyperboxes that are expandable and expand them. If no expandable hyperbox is found, a new hyperbox corresponding to this class is inserted. For hyperbox  $B_j$  to expand, the following constraint must be met:

$$n\theta \geq \sum_{i=1}^n (\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})), \quad (4)$$

where  $0 < \theta < 1$  is a user-defined parameter to control the hyperbox size. If this expansion constraint is met, the minimum and maximum points of the hyperbox  $B_j$  are modified to:

$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \quad \forall i, i = 1, 2, \dots, n; \quad (5)$$

$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \quad \forall i, i = 1, 2, \dots, n. \quad (6)$$

- (b) *Overlapping Test:* This step determines whether there are any hyperboxes belonging to different classes that are overlapped with each other. For each dimension of the input pattern, if at least one of the following four cases is satisfied, then the two hyperboxes are overlapped with each other. If overlapping is detected, the contraction process is implemented based on the index of the feature dimension and the smallest overlap value. Assuming that  $\delta^{old} = 1$  initially, the four overlapping cases and the corresponding minimum overlapping value are:

**Case 1:**  $v_{ji} < v_{ki} < w_{ji} < w_{ki}$ ,  
 $\delta^{new} = \min(w_{ji} - v_{ki}, \delta^{old}); \quad (7)$

**Case 2:**  $v_{ki} < v_{ji} < w_{ki} < w_{ji}$ ,  
 $\delta^{new} = \min(w_{ki} - v_{ji}, \delta^{old}); \quad (8)$

**Case 3:**  $v_{ji} < v_{ki} < w_{ki} < w_{ji}$ ,  
 $\delta^{new} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old}); \quad (9)$

**Case 4:**  $v_{ki} < v_{ji} < w_{ji} < w_{ki}$ ,  
 $\delta^{new} = \min(\min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{old}); \quad (10)$

where  $j$  corresponds to hyperbox  $B_j$  that has been expanded in the previous step, and  $k$  corresponds to hyperbox  $B_k$  that belongs to another class being examined for possible overlapping. If  $\delta^{old} - \delta^{new} > 0$ , then the  $\Delta = i$ th dimension is identified as overlapping, and  $\delta^{old}$  is set to  $\delta^{new}$ . This process is repeated for all the dimensions. If no overlapping is identified,  $\Delta$  is set to a negative value to indicate that it is not necessary to contract the hyperboxes.

- (c) *Contraction:* If there exists overlapped hyperboxes that belong to different classes, eliminate the overlapped region. In order to keep the hyperbox size as large as possible, only one of the  $n$  dimensions in each hyperbox is selected for adjustment, with the aim to maintain the hyperbox size as large as possible while eliminating the overlapped region. The following four cases are examined to properly adjust the hyperbox.

**Case 1:**  $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$ ,  
 $w_{j\Delta}^{new} = v_{k\Delta}^{new} = \frac{w_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}; \quad (11)$

**Case 2:**  $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$ ,  
 $w_{k\Delta}^{new} = v_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}; \quad (12)$

**Case 3a:**  $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$  and  $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$ ,  
 $v_{j\Delta}^{new} = w_{k\Delta}^{old}; \quad (13)$

**Case 3b:**  $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$  and  $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$ ,  
 $w_{j\Delta}^{new} = v_{k\Delta}^{old}; \quad (14)$

**Case 4a:**  $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$  and  $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$ ,  
 $w_{k\Delta}^{new} = v_{j\Delta}^{old}; \quad (15)$

**Case 4b:**  $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$  and  $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$ ,  
 $v_{k\Delta}^{new} = w_{j\Delta}^{old}. \quad (16)$

### 3 FMM network pruning

#### 3.1 FMM pruning based on confidence factor

In order to obtain a small network size with a high-classification performance, the works in [13, 27, 28] introduce a pruning procedure into FMM. The basic idea is to calculate a confidence factor ( $CF_j$ ) for each hyperbox  $B_j$  to measure the degree of usage frequency and prediction accuracy. A hyperbox with a CF lower than a pre-defined threshold is removed. The CF is defined as:

$$CF_j = (1 - \gamma)U_j + \gamma A_j, \quad (17)$$

where  $U_j$  and  $A_j$  are the usage and accuracy indices of hyperbox  $B_j$ , and  $\gamma \in [0, 1]$  is a weighting factor. Using a training set,  $U_j$  is calculated as the number of patterns classified by hyperbox  $B_j$  divided by the total number of patterns classified by all the hyperboxes that belong to the same class. Using a prediction data set,  $A_j$  is calculated as the number of correctly classified patterns by hyperbox  $B_j$  divided by the total number of correctly classified patterns in the same class. Once the CF for each hyperbox is calculated, the hyperboxes that have CF scores lower than a pre-defined threshold are pruned. After pruning, the remaining hyperboxes are used in the next step for decision rule extraction.



### 3.2 FMM pruning by GA

Pruning of FMM based on CF (i.e., FMM-CF) is able to remove hyperboxes that are rarely used and are less accurate. However, FMM-CF cannot guarantee that the pruned FMM network is optimal in terms of prediction accuracy. Another disadvantage of FMM-CF is that it is difficult to determine the optimal CF threshold to prune FMM, which affects prediction accuracy. In order to address this problem, the genetic algorithm (GA) [29] is employed to prune original FMM (i.e., FMM-GA), which is able to determine the optimal configuration of hyperboxes in FMM.

Suppose that there are  $P$  hyperboxes in original FMM, the chromosome of the GA(S) is a binary string that encodes a solution consisting of all possible hyperbox configurations, as follows:

$$S = \{s_1, s_2, \dots, s_P\}, \quad (18)$$

where the binary value of the allele  $s_i$  indicates whether the  $i$ -th hyperbox should be pruned, i.e., 0 indicates that the  $i$ -th hyperbox should be pruned, and 1 otherwise.

Similar to [26], an objective function (fitness function) is defined to maximize prediction accuracy and minimize the number of hyperboxes, as follows:

$$\text{maximize}(f(S)) = \text{NCP}(S) - \lambda|S|, \quad (19)$$

where  $\text{NCP}(S)$  is the number of data samples that are correctly classified by pruned FMM (with selected hyperboxes),  $|S|$  is the number of hyperboxes after pruning in pruned FMM, and  $\lambda$  is a positive weight factor that is  $0 < \lambda < 1$ . The general process of the genetic operation is implemented as follows.

1. *Initialization* Randomly generate many ( $N_{\text{pop}}$ ) individual population strings (solutions) in each generation. The population strings are generated by assigning 1 for keeping the hyperbox and 0 for pruning the hyperbox.
2. *Selection* Select  $N_{\text{pop}}/2$  pairs of population strings. The selection probability  $P(S)$  of a string  $S$  in a population  $\Psi$  is determined by:

$$P(S) = \frac{\{f(S) - f_{\min}(\Psi)\}}{\sum_{S \in \Psi} \{f(S) - f_{\min}(\Psi)\}}, \quad (20)$$

where  $f_{\min}(\Psi) = \min\{f(S) \mid S \in \Psi\}$  is the minimum of the fitness function of the population  $\Psi$ .

3. *Crossover* Randomly select the bit position with a probability and interchange the bit values at the selected positions for each of the selected pairs.
4. *Mutation* Mutate new offspring at each bit value of the generated strings. The following mutation operation is implemented:

$s_i = 1 \rightarrow s_i = 0$  with probability  $P_m(1 \rightarrow 0)$ ;

$s_i = 0 \rightarrow s_i = 1$  with probability  $P_m(0 \rightarrow 1)$ .

5. *Replacement* Randomly remove one string from  $N_{\text{pop}}$ , and add the offspring that has the highest fitness function value into the current population.
6. *Termination* If the termination conditions are satisfied, then stop. Otherwise, return to the first step to repeat this process.

### 4 Fuzzy if then rule extracting

All the evolved hyperboxes are used for rules extraction. The minimum and maximum values of each input pattern are quantized into  $Q$  levels, which equals to the number of fuzzy partitions in the quantized rules [22]. As an example, with  $Q = 5$ , input feature  $A_q$  is quantized to *very low*, *low*, *medium*, *high*, and *very high* in a fuzzy rule. For quantization, interval  $[0, 1]$  is divided into  $Q$  intervals, and the input pattern is assigned to the quantization points evenly with one at each of the end points using [22]:

$$A_q = \frac{q - 1}{Q - 1}, \quad (21)$$

where  $q = 1, 2, \dots, Q$ . The fuzzy if-then rule extracted is as follows:

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

THEN  $x_p$  is class  $C_j$  with  $\text{CF} = \text{CF}_j, j = 1, 2, \dots, N$ , where  $N$  is the number of hyperboxes,  $x_p = (x_{p1}, \dots, x_{pn})$  is an input pattern vector with dimension of  $n$ ,  $A_q$  is the antecedent value, and  $\text{CF}_j$  is the CF of the  $j$ th hyperbox (rule).

### 5 Experiment on patient admission dataset

#### 5.1 Dataset and experiment setup

The patient admission dataset was collected from the Emergency Department of Ballarat Base Hospital, Australia. This dataset consists of 450740 real patient records. Each record contains information pertaining to a patient visiting the emergency department. The following information is extracted from the patient record, while all other private details (e.g., name, address, and contact numbers) are removed, in order to protect privacy of patients.

- **Compensable**: describes if it is compensable. There are two values, i.e., 1 = Compensable and 2 = NonCompensable.

- **ArrivalMode**: describes how the patients arrived at the emergency department. There are three arrive models, i.e., by ambulance, by police, and by others.
- **TriageCategory**: describes the triage categories. Values are from 1 to 6.
- **ServiceType**: describes service types, values are from 1 to 9.
- **AgeInDays**: describes a patient' age in days at the time of presentation.
- **Sex**: describes a patient' sex. 0 = Female, 1 = Male
- **IndigStatus**: describes if a patient is Aboriginal. 0 = Not Aboriginal, 1 = Aboriginal or Torres Strait or Both, and 9 = Unknown/NotStated
- **Weeknum**: Weeknumber of year.
- **EDQueueDurationMinutes**: gives queue duration in the emergency department in minutes.
- **EDServiceDurationMinutes**: gives service duration in the emergency department in minutes.

The above information constitutes ten input features to FMM-GA for experimentation. All the features are normalized so that they are at the same scale. A total of 2,000 data samples are used to train and prune the FMM model (1,000 for training of FMM and 1,000 for pruning with the GA), and the remaining samples (448740) are used to test the prediction performance. It should be noted that the test dataset is very large, which implies that the prediction result is based on a large sample size.

In order to prune the hyperboxes efficiently, the GA parameters were set as follows.

- objective function weight:  $\lambda = 0.5$ ;
- population size:  $N_{\text{pop}} = 50$ ;
- crossover probability: 0.9;
- mutation probability: 0.1;
- stopping condition: 500 generations, or no change in the fitness function value for 100 consecutive generations.

The performance of the proposed FMM-GA model was evaluated using the following three classification criteria:

- **Accuracy**: The number of correctly classified samples divided by the total number of test samples;
- **Sensitivity**: The number of positive samples (i.e., Admission Accept samples) correctly predicted divided by the number of samples that are actually positive;
- **Specificity**: The number of negative samples (i.e., Not Admitted samples) correctly predicted divided by the number of samples that are actually negative.

In order to assess the results fairly, all the experiments were repeated twenty times, each time with a random sequence of the training data samples. To evaluate the results statistically, the bootstrap method [30] was used to compute the averages of performance indicators, including

accuracy, sensitivity, specificity, and the number of hyperboxes<sup>1</sup>.

## 5.2 Prediction results and extracted decision rules

Three FMM models were experimented, i.e., original FMM, FMM-CF, and FMM-GA. The hyperbox size was varied from 0.05 to 0.7 with an interval of 0.05. Setting the hyperbox size larger than 0.7 caused prediction accuracy to deteriorate below 50 %, which is not useful in binary classification (i.e., worse than the probability of coin tossing). Table 1 summarizes bootstrapped prediction accuracy, sensitivity, specificity, and the corresponding number of hyperboxes by FMM, FMM-CF and FMM-GA with various setting of  $\theta$ . In FMM-CF, the CF threshold was empirically set to 0.6 (this parameter was varied later). It can be seen that both FMM-CF and FMM-GA eliminate a large number of hyperboxes from original FMM, which simplifies the network structure considerably. The number of hyperboxes learned by original FMM ranges from 28 to 989. After pruning by FMM-CF, the number of hyperboxes decreases from 2 to 458. By contrast, pruning with the GA (FMM-GA) is able to greatly reduce the number of hyperboxes, resulting in only 3–23 hyperboxes. Notice that with small setting of  $\theta$  (0.05 and 0.1), the numbers of hyperboxes are pruned from 989 and 933 to 458 and 176 by FMM-CF, respectively, which are still too many for rule extraction. By contrast, the numbers of hyperboxes produced by FMM-GA are only 23 and 19, respectively. This indicates a parsimonious rule base for domain users (medical practitioners) to interpret the predictions from the resulting decision support system.

In general, FMM-GA achieves considerably higher accuracy than both FMM and FMM-CF, which mainly attributes to the optimization process. The highest prediction accuracy achieved by FMM-GA is 77.97 % (with  $\theta = 0.3$ ), while those obtained by FMM and FMM-CF are 75.36 % (with  $\theta = 0.05$ ) and 74.09 % (with  $\theta = 0.05$ ), respectively. Furthermore, the corresponding sensitivity (77.88 %) and specificity (77.58 %) achieved by FMM-GA are also more promising than those obtained by FMM and FMM-CF, which are two important performance metrics in medical prediction applications. The corresponding numbers of hyperboxes created by FMM, FMM-CF, and FMM-GA are 989, 458, and 17, respectively. Overall, FMM-GA achieves a higher accuracy with a better-scale hyperbox number than both FMM and FMM-CF. This mainly attributes to the fact that the GA-based pruning process in the

<sup>1</sup> Since the number of hyperbox is an integral number, the nearest integral value is reported.

**Table 1** Prediction results with respect to hyperbox size parameter (i.e.,  $\theta$  in Eq. 4) by FMM, FMM-CF, and FMM-GA

$\theta$	FMM				FMM-CF				FMM-GA			
	Acc	Sen	Spe	Num	Acc	Sen	Spe	Num	Acc	Sen	Spe	Num
0.05	75.36	70.81	76.24	989	74.09	74.20	74.14	458	77.72	77.57	77.82	23
0.1	72.61	69.03	73.26	933	70.69	77.66	69.52	176	77.31	77.78	77.28	19
0.15	71.60	70.68	71.77	841	66.42	83.70	62.83	41	77.28	78.02	77.18	17
0.2	67.35	70.37	66.60	713	67.55	78.13	65.72	22	77.72	77.21	78.09	14
0.25	67.61	69.70	67.00	587	68.91	77.36	67.31	13	77.92	78.19	77.91	12
0.3	67.04	70.42	66.02	460	69.35	81.97	66.29	8	77.97	77.88	77.58	17
0.35	70.45	69.78	70.55	364	65.16	73.93	63.59	6	77.46	77.45	77.45	15
0.4	66.47	61.31	67.18	269	72.66	75.82	72.44	5	74.74	71.77	74.94	13
0.45	70.44	66.94	71.44	210	71.34	61.15	73.05	3	77.60	75.29	78.10	10
0.5	74.06	60.26	76.79	145	73.07	70.30	73.97	3	77.39	73.34	78.32	7
0.55	69.47	70.07	69.44	100	70.42	64.44	71.72	2	77.34	73.91	77.99	5
0.6	72.09	59.64	74.08	73	64.92	65.82	64.92	2	73.09	69.15	73.40	3
0.65	66.43	61.39	68.50	49	58.97	69.15	56.81	2	71.02	60.99	73.39	3
0.7	64.98	51.27	65.82	28	64.88	53.73	65.85	2	65.55	60.77	66.69	3

*Acc* means accuracy, *Sen* means sensitivity, *Spe* is specificity, and *Num* means the corresponding number of hyperboxes

**Table 2** Prediction results with respect to the confidence factor by FMM-CF

Threshold	Acc	Sen	Spe	Num
0.05	70.65	71.29	70.33	224
0.1	69.77	69.52	69.92	214
0.15	66.81	71.83	65.59	204
0.2	68.93	72.98	68.24	204
0.25	70.70	73.13	70.10	201
0.3	72.51	74.08	72.43	198
0.35	70.37	74.72	69.53	195
0.4	72.28	74.91	72.01	193
0.45	69.89	80.01	67.82	192
0.5	72.01	77.09	70.69	20
0.55	64.97	84.28	60.61	9
0.6	69.35	81.97	66.29	8
0.65	66.18	73.68	64.11	4
0.7	70.58	68.56	71.07	3

*Acc* means accuracy, *Sen* means sensitivity, *Spe* is specificity, and *Num* is the number of hyperboxes

proposed method optimizes the FMM model to better fit the underlying data samples dataset with fewer hyperboxes.

In order to fairly compare the performance of both GA-based and CF-based pruning methods, we fixed the hyperbox size parameter  $\theta$  to 0.3 and varied the CF threshold from 0.05 to 0.7. Since a CF threshold larger than 0.7 removes nearly all the hyperboxes from original FMM, the CF threshold is restricted to 0.7 or lower. Table 2 gives the prediction results as well as the corresponding hyperboxes number obtained by FMM-CF with different CF thresholds. Figures 4a, b compare prediction accuracy and the hyperboxes number by FMM-CF and FMM-GA. We can see that

the highest prediction accuracy obtained by FMM-CF with different CF threshold is still considerably lower than that by FMM-GA. Notice that the hyperbox number fluctuates around 200 when the CF threshold is smaller than 0.55, while it suddenly drops to 10 and fewer when the CF threshold is 0.6 and larger. This observation highlights the drawback of FMM-CF, i.e., it is difficult to determine the optimal CF threshold. By contrast, FMM-GA does not require this parameter. Instead, FMM-GA defines the optimization objective function to adaptively prune the hyperboxes learned by original FMM.

Table 3 shows the decision rules extracted by FMM-GA ( $\theta = 0.3$ ) with a quantization level of 5 for the decision-making process<sup>2</sup>. As an example, Rule 1 is interpreted as in Table 4. It is interesting that the values in the rule for ArrivalMode and Sex cover the complete range (i.e., a “don’t care” condition), which could be omitted in order to simplify interpretation of this rule.

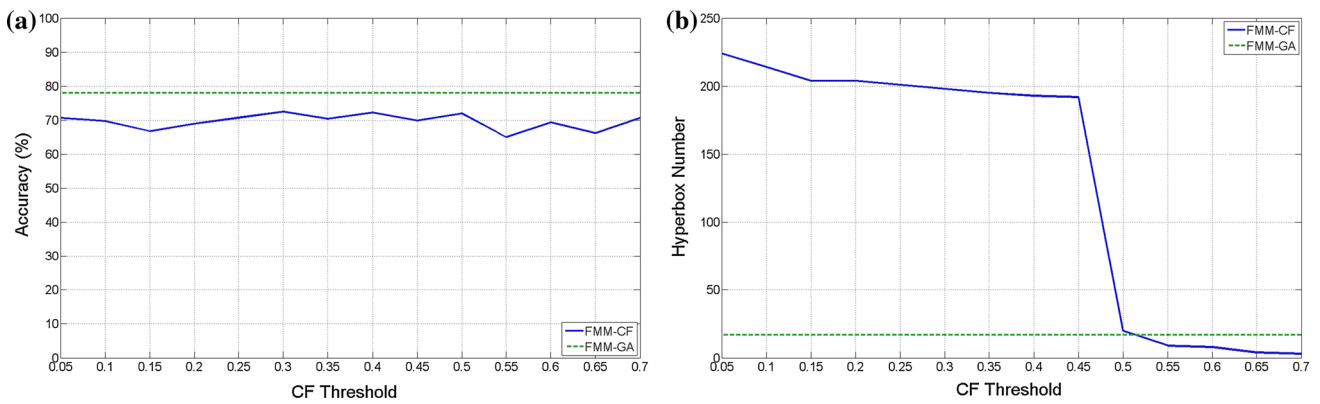
### 5.3 Comparison with state-of-the-art classifiers

We compared the performance of the proposed FMM-GA model against several state-of-the-art classifiers. The following classifiers were used for comparison:

- Multi-Layer Perceptron (MLP) [6]: MLP is a feed-forward neural network that contains multiple layers of nodes in a direct graph. All nodes in a layer are fully connected with the other nodes in the next layer. Except for the input nodes, each node is a neuron with a

<sup>2</sup> The number of the decision rules is not the same as the number of hyperboxes reported in Table 1 because the number of hyperboxes is the mean of multiple implementations.





**Fig. 4** Prediction accuracies on the Patient Admission Dataset by FMM-CF and FMM-GA (a) and the corresponding hyperbox numbers extracted by FMM-CF and FMM-GA (b)

**Table 3** Decision rules extracted by FMM-GA

Rules	IF (features)										THEN (classes)
	1	2	3	4	5	6	7	8	9	10	
Rule 1	5	1–5	2–3	1	2–4	1–5	1	2–4	1	1	0
Rule 2	5	1–5	2–3	1	2–4	1–5	1	4–5	1	1	1
Rule 3	5	5	1–3	1	1–4	1–5	1	2–4	1	1–2	1
Rule 4	5	1–5	2–4	1	1–5	1	1	3–5	1	1–2	1
Rule 5	5	5	2	1	2–3	1–5	1	1–2	1	2–4	1
Rule 6	5	5	2–3	1	1–3	1–5	1	1–3	1–2	2–3	1
Rule 7	5	1–5	1–2	1	1	5	1	4–5	1	1–2	1
Rule 8	5	1	2	1	2–4	1–5	1	3–4	1	1	1
Rule 9	5	5	2–3	1	1–2	1–5	1	1–5	1	1–3	1
Rule 10	5	1–5	3	1	1–3	1–5	1	3–5	1	1	0
Rule 11	5	5	1–3	1	1–4	1–5	1	1–5	1	1	0
Rule 12	5	5	3	1	1–4	1–5	1	2–5	1–3	1	0
Rule 13	5	5	2–4	1	1–4	1–5	1	3–5	1	1	0
Rule 14	5	5	3	1	1	1	1	1	1	1	0
Rule 15	5	1–5	2–4	1	3–5	5	1	2–3	1	2	1
Rule 16	5	5	2–3	1	1–4	1–5	1	1–4	1–2	1	0
Rule 17	5	5	2–3	1	1–2	1–5	1	2–3	1–2	1	0
Rule 18	5	5	1–2	1	1–4	1–5	1	1–5	1	1–2	1
Rule 19	5	1–5	2–3	1	3–4	1–5	1	3–4	1	1–2	1
Rule 20	5	1–5	2–3	1	3–4	5	1	3–5	1	1–2	1
Rule 21	5	5	2–3	1	1–4	5	1	1–3	1	1–2	1
Rule 22	5	1–5	1–3	1	2–4	5	1	1–5	1	1–2	1
Rule 23	5	5	3–4	1	1–3	1–5	1	1–4	1–3	1	0
Rule 24	5	1–5	1–4	1	2–4	5	1	3–5	1–2	2	1
Rule 25	5	5	2–3	1	2–5	1–5	1	1–4	1	1–2	1

1 very low, 2 low, 3 medium, 4 high, 5 very high

nonlinear activation function, which enables the MLP to distinguish nonlinear datasets. In the experiment, the learning rate (i.e., the amount of weights are updated) was set to 0.3, and the momentum applied to the weights during updating was set to 0.2.

- Adaboost [7, 8]: Adaboost combines a set of relatively weak and inaccurate classifiers to achieve high accuracy. Adaboost is a powerful classifier that works well on both basic and more complex classification

problems. In this experiment, the weak classifiers used in each round of Adaboost was the Decision Stump.

- Random Forest [9]: Random Forest is an ensemble learning method for classification. It constructs a multitude of decision trees during training and outputs the class that is the mode of the classes output by individual trees. The Random Forest classifier has been shown to be very successful in tackling complex real-world classification problems. The number of tree to be

**Table 4** Interpretation of rule 1

<b>IF</b>	Compensable is <i>very high</i> ArrivalMode is from <i>very low</i> to <i>very high</i> TriageCategory is from <i>low</i> to <i>medium</i> ServiceType is <i>very low</i> AgeInDays is from <i>low</i> to <i>high</i> Sex is from <i>very low</i> to <i>very high</i> IndigStatus is <i>very low</i> Weeknum is from <i>low</i> to <i>high</i> EDQueueDurationMinutes is <i>very low</i> EDServiceDurationMinutes is <i>very low</i>
<b>THEN</b>	Output is “Admission Decline”

**Table 5** Comparison results with state-of-the-art classifiers

Methods	Acc	Sen	Spe
MLP	80.96	82.76	80.61
Adaboost	77.18	86.05	75.43
Random Forest	80.68	77.38	81.34
CART	75.98	76.74	75.84
FMM	75.36	70.81	76.24
FMM-CF	74.09	74.20	74.14
FMM-GA	77.97	77.88	77.58

Acc means accuracy, Sen means sensitivity, and Spe is Specificity

**Table 6** Classification results by FMM and FMM-GA on the three benchmark datasets

Dataset	FMM		FMM-CF		FMM-GA	
	Acc	Num	Acc	Num	Acc	Num
Iris	93.33	53	95.75	13	95.75	4
PID	61.98	187	69.79	35	70.57	14
Sonar	56.09	118	56.09	117	73.17	7

Acc means accuracy, and Num means the number of hyperboxes

generated was set to 10, and the maximum depth of the trees was unlimited.

- **Classification And Regression Tree (CART) [10]:** It learns a tree structure from training data for classification. In the tree structure, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. One advantage of the CART classifier is that the decision tree learned from training data is very useful for interpreting the prediction. In the experiment, the parameters were set as default values of MATLAB (Version R2010a) function *classregtree* for classification.

The same training and testing data sets were used in the comparison experiment. Table 5 compares the classification results achieved by FMM, FMM-CF and FMM-GA

with those from MLP, Adaboost, Random Forest and CART<sup>3</sup>. We can see that the prediction accuracy (77.97 %) achieved by FMM-GA is lower than those by MLP (80.96 %) and Random Forest (80.68 %), but higher than those by Adaboost (77.18 %) and CART (75.98 %). It is worth mentioning that only CART is able to produce a decision trees that is useful for prediction interpretation. While the other three classifiers (MLP, Adaboost, and Random Forest) perform generally better than FMM-GA and CART, they are only able to produce a prediction for each test sample without being able to explain and justify the prediction. In summary, while the proposed FMM-GA model produces slightly inferior results than black-box classifiers, it is able to yield a parsimonious rule base to provide explanation pertaining to its predictions. This rule extraction capability is very important in practice as it is able to help medical practitioners to understand the underlying reasoning process of a neural network model that leads to the resulting prediction. Therefore, they are able to appreciate and accept the use of neural network as a decision support system in their domain.

## 6 Experiment on benchmark datasets

In this section, we provide further evaluation of the classification performance of original FMM, FMM-CF, and FMM-GA using three benchmark datasets. The following benchmark datasets from the UCI machine-learning repository [31] were used:

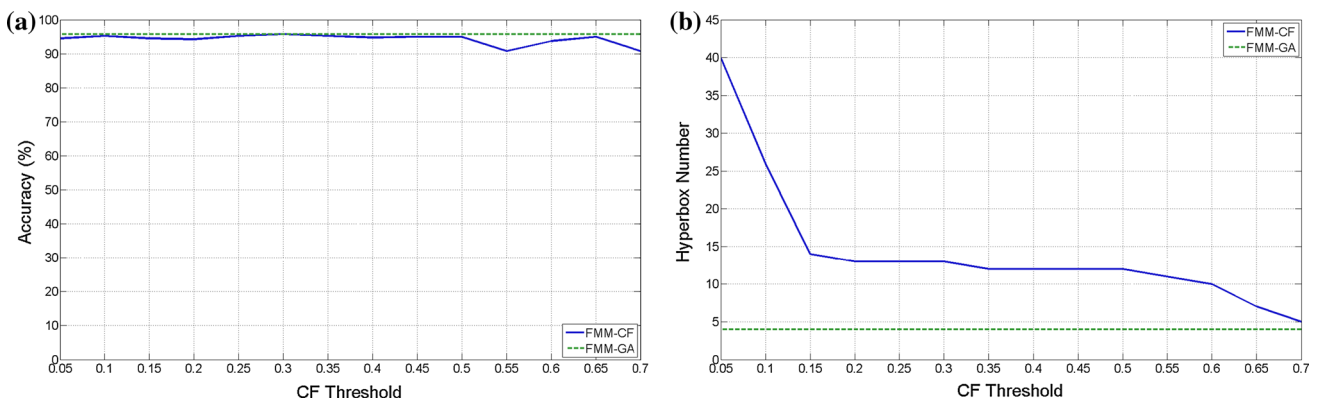
- **The Iris dataset:** This dataset consists of 150 samples from three classes (Iris setosa, Iris versicolor, and Iris virginica, each of which has four continuous features (sepal length, sepal width, petal length, and petal width).
- **The PID dataset:** This dataset contains 768 samples from two classes (diabetic and healthy), each with eight features. 35 % of the samples (i.e., 268 samples) are collected from patients diagnosed as diabetic and the others are as healthy.
- **The Sonar dataset:** This dataset consists of 208 samples from two classes (i.e., sonar signals collected from mine or rocks, respectively). This dataset is a high-dimension dataset and each of the samples has 60 features.

Table 6 gives the classification results by FMM, FMM-CF and FMM-GA with hyperbox size parameter  $\theta = 0.3$ .

<sup>3</sup> We varied the parameters of these four classifiers and found that the accuracies fluctuated slightly.

**Table 7** Classification results with respect to CF threshold by FMM-CF on the three benchmark datasets

Threshold	Iris		PID		Sonar	
	Accuracy	Hyperbox number	Accuracy	Hyperbox number	Accuracy	Hyperbox number
0.05	94.50	40	67.19	101	56.09	118
0.1	95.25	26	66.04	66	56.09	118
0.15	94.50	14	67.11	50	56.09	118
0.2	94.25	13	66.69	45	56.09	118
0.25	95.25	13	67.68	42	56.09	118
0.3	95.75	13	66.35	37	56.09	117
0.35	95.25	12	69.79	35	55.97	117
0.4	94.75	12	68.67	33	56.09	117
0.45	95.00	12	68.07	30	56.09	117
0.5	95.00	12	69.53	26	55.97	117
0.55	90.75	11	68.54	20	51.22	22
0.6	93.75	10	67.05	11	51.22	22
0.65	95.00	7	65.16	9	51.22	22
0.7	90.75	5	64.92	5	51.22	22



**Fig. 5** Classification accuracies on the Iris dataset by FMM-CF and FMM-GA (a) and the corresponding hyperbox numbers extracted by FMM-CF and FMM-GA (b)

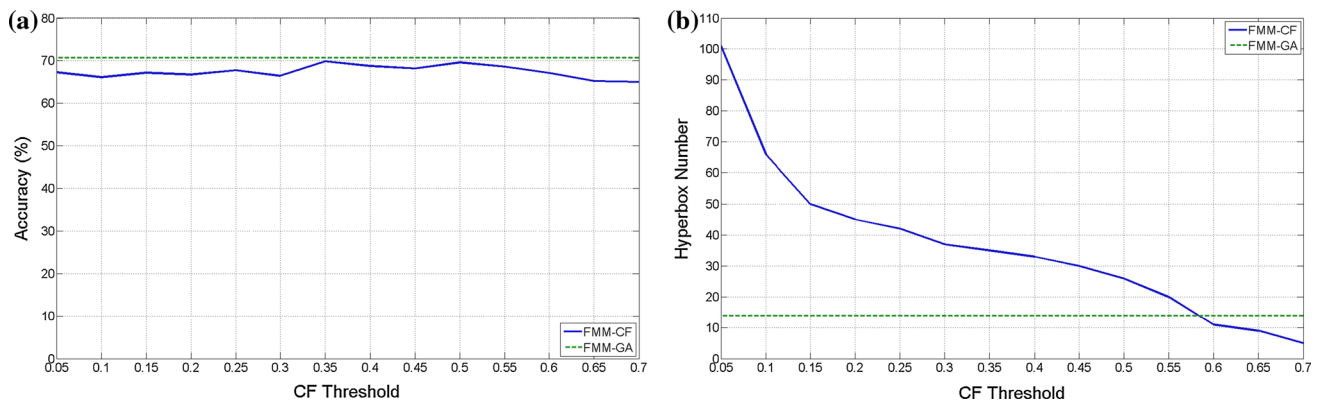
In general, FMM-GA is able to achieve comparable or better classification accuracy with fewer number of hyperboxes, as compared with original FMM and FMM-CF. Notice that both original FMM and FMM-CF only produce around 56 % accuracy with 118 and 117 hyperboxes using the Sonar dataset. FMM-GA significantly improves the classification accuracy to 73.17% with 7 hyperboxes.

Table 7 gives the classification results on the three benchmark datasets by FMM-CF with CF threshold varying from 0.05 to 0.7. It shows that the classification accuracies vary considerably with respect to the CF threshold. This indicates the drawback of FMM-CF, i.e., it is not easy to choose the optimal CF threshold. Figures 5, 6, and 7 show comparison of classification accuracy and the number of hyperboxes of FMM-CF and FMM-GA on the three

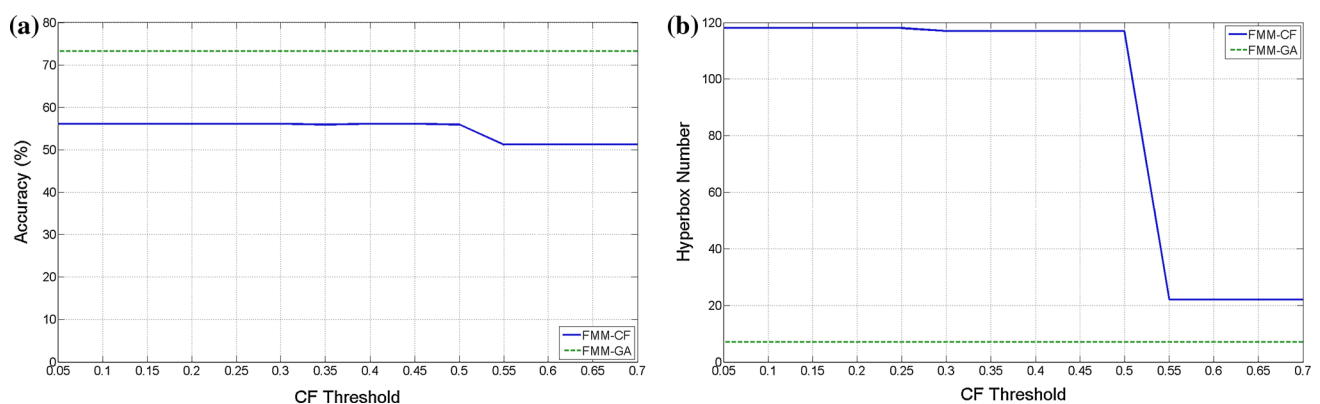
benchmark datasets, respectively. We can see that the accuracies obtained by FMM-GA with optimal CF thresholds (0.3 on Iris, 0.5 on PID and 0.45 on Sonar) are still inferior or comparable to those achieved by FMM-GA. Overall, the proposed FMM-GA model achieved comparable or higher accuracy than original FMM and FMM-CF with a parsimonious number of hyperboxes.

### 7 Conclusion and future works

In this paper, the FMM neural network has been enhanced for tackling real-world problems related to patient admission prediction. Decision rules that provide an explicit explanation of the decision process have been extracted from the hyperboxes learned by FMM. In order to simplify



**Fig. 6** Classification accuracy on the PID dataset by FMM-CF and FMM-GA (a) and the corresponding numbers extracted by FMM-CF and FMM-GA (b)



**Fig. 7** Classification accuracy on the Sonar dataset by FMM-CF and FMM-GA (a) and the corresponding numbers extracted by FMM-CF and FMM-GA (b)

the decision rules and the network structure of FMM, a GA-based pruning method has been formulated to find the optimal solution (i.e., hyperboxes pruning) that maximizes prediction accuracy and simultaneously minimizes the number of hyperboxes. Unlike the CF-based pruning method (FMM-CF) that requires a user-defined CF threshold to remove the hyperboxes, the proposed FMM-GA model is able to automatically determine the optimal configuration of the hyperboxes in terms of prediction accuracy and the number of hyperboxes.

The proposed FMM-GA model has been evaluated using a very large patient admission dataset that consists of 450740 data samples collected from an emergency department of a hospital. The experimental results demonstrate that the proposed method achieves good results with the ability to establish a parsimonious rule base. In addition, the proposed model has been tested on three benchmark datasets from the UCI machine-learning repository, which reveals that the proposed model performs very well relatively to the other state-of-the-art methods.

For further work, it is useful to investigate patient-specific information in the dataset to improve prediction

accuracy. As the same patient could visit the emergency department several times, it is beneficial to correlate all the information from the same patient in order to reach a more accurate prediction. This patient-specific information needs to be investigated in order to extract more useful representative features. Another information to be considered is the date of each visit, which is ignored in this study. This temporal information will be investigated in future work. Another suggestion is to investigate how the prediction results of the proposed model can be further improved, e.g., by using an ensemble approach, so that it is more useful in real clinical environments.

**Acknowledgments** The authors would like to thank Ballarat Health Services, Australia, and Victorian Department of Health, Australia, for giving us access to the patient admission dataset.

## References

- Boyle J, Lind J, Jessup M, Wallis M, Crilly J, Miller P, Green D, Fitzgerald G (2012) Predicting emergency department admissions. *Emerg Med J* 29:358–365

2. Jessup M, Wallis M, Boyle J, Crilly J, Lind J, Green D, Miller P, FitzGerald G (2010) Implementing an emergency department patient admission predictive tool: insights from practice. *J Health Organ Manag* 24:306–318
3. Sun Y, Heng BH, Tay SY, Seo E (2011) Predicting hospital admissions at emergency department triage using routine administrative data. *Acad Emerg Med* 18:844–850
4. Peck JS, Benneyan JC, Nightingale DJ, Gaehde DA (2012) Predicting emergency department inpatient admissions to improve same-day patient flow. *Acad Emerg Med* 19:1045–1054
5. Leegon J, Jones I, Lanaghan K, Aronsky D (2005) Predicting hospital admission for emergency department patients using a bayesian network. In *AMIA Annu Symp Proc*, p 1022
6. Haykin S (1998) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice Hall PTR, Upper Saddle River
7. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
8. Bishop CM (2006) *Pattern recognition and machine learning*. Springer-Verlag New York Inc, Secaucus
9. Breiman L (2001) Random forests. *Mach Learn* 45(1): 5–32. ISSN 0885–6125.
10. Breiman L (1984) *Classification and regression trees*. The Wadsworth and Brooks-Cole statistics-probability series. Chapman & Hall, London
11. Simpson PK (1992) Fuzzy min-max neural networks—part 1: classification. *IEEE Trans Neural Netw* 3(5):776–786
12. Simpson PK (1993) Fuzzy min-max neural networks—part 2: clustering. *IEEE Trans Fuzzy Syst* 1(1):32–45
13. Quteishat A, Lim CP (2008) Application of the fuzzy min-max neural networks to medical diagnosis. In Ignac Lovrek, RobertJ. Howlett, and LakhmiC. Jain, editors, *Knowledge-based intelligent information and engineering systems*, volume 5179 of *Lecture Notes in Computer Science*. Springer, Berlin, pp 548–555
14. Nandedkar AV, Biswas PK (2009) A granular reflex fuzzy min-max neural network for classification. *IEEE Trans Neural Netw* 20(7):1117–1134. doi:10.1109/TNN.2009.2016419
15. Seera M, Lim CP, Ishak D, Singh H (2012) Fault detection and diagnosis of induction motors using motor current signature analysis and a hybrid FMM–CART model. *IEEE Trans Neural Netw Learn Syst* 23(1):97–108
16. Seera M, Lim CP (2013) Online motor fault detection and diagnosis using a hybrid fmm-cart model. *IEEE Trans Neural Netw Learn Syst* 99:1–7. doi:10.1109/TNNLS.2013.2280280
17. Seera M, Lim CP, Ishak D, Singh H (2013) Offline and online fault detection and diagnosis of induction motors using a hybrid soft computing model. *Appl Soft Comput* 13 (12): 4493–4507. ISSN 1568–4946. doi:10.1016/j.asoc.2013.08.002
18. Zhang H, Liu J, Ma D, Wang Z (2011) Data-core-based fuzzy min–max neural network for pattern classification. *IEEE Trans Neural Netw* 22(12):2339–2352. doi:10.1109/TNN.2011.2175748
19. Davtalab R, Dezfoulian MH, Mansoorizadeh M (2014) Multi-level fuzzy min-max neural network classifier. *IEEE Trans Neural Netw Learn Syst* 25 (3): 470–482. ISSN 2162–237X. doi:10.1109/TNNLS.2013.2275937
20. Benitez JM, Castro JL, Requena I (1997) Are artificial neural networks black boxes? *IEEE Trans Neural Netw* 8(5):1156–1164
21. Taha IA, Ghosh J (1999) Symbolic interpretation of artificial neural networks. *IEEE Trans Knowl Data Eng* 11(3):448–463
22. Carpenter GA, Tan A (1995) Rule extraction: from neural architecture to symbolic representation. *Connect Sci* 7:3–27
23. Setiono R, Leow WK, Zurada JM (2002) Extraction of rules from artificial neural networks for nonlinear regression. *IEEE Trans Neural Netw* 13(3):564–577
24. Ishibuchi H, Nozaki K, Yamamoto N, Tanaka H (1994) Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets Syst* 65(2–3): 237–253. ISSN 0165–0114
25. Ishibuchi H, Nii M, Murata T (1997a) Linguistic rule extraction from neural networks and genetic-algorithm-based rule selection. In: *International conference on neural networks vol 4*, pp 2390–2395
26. Ishibuchi H, Murata T, Turksen IBIB (1997b) Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets Syst* 89(2):135–150
27. Quteishat A, Lim CP (2008b) A modified fuzzy min-max neural network with rule extraction and its application to fault detection and classification. *Appl Soft Comput* 8(2):985–995
28. Quteishat A, Lim CP, Tan KS (2010) A modified fuzzy min-max neural network with a genetic-algorithm-based rule extractor for pattern classification. *IEEE Trans Syst Man Cybern Part A Syst Hum* 40(3):641–650
29. Mitchell M (1998) *An introduction to genetic algorithms*. MIT Press, Cambridge
30. Zoubir AM, Boashash B (1998) The bootstrap and its application in signal processing. *IEEE Signal Process Mag* 15(1):56–76
31. Murphy PM, Aha DW (1995) *UCI Repository of Machine Learning Databases*, (Machine-Readable Data Repository). Dept. Inf. Comput. Sci., Univ. California, Irvine, CA