

# Extracting the Classification Rules from General Fuzzy Min-Max Neural Network

S. V. Shinde

Professor,

Department of Information Technology,  
PCCOE, Pune, India

U. V. Kulkarni

Professor,

Department of Computer Engineering,  
SGGSIT, Nanded, India

A. N. Chaudhary

Assistant Professor,

Department of Information Technology,  
PCCOE, Pune, India

## ABSTRACT

The general fuzzy min-max neural network (GFMMN) is capable to perform the classification as well as clustering of the data. In addition to this it has the ability of learning in a very few passes with a very short training time. But like other artificial neural networks, GFMMN is also like a black box and expressed in terms of min-max values and associated class label. So the justification of classification results given by GFMMN is required to be obtained to make it more adaptive to the real world applications. This paper proposes the model to extract classification rules from trained GFMMN. These rules justify the classification decision given by GFMMN. For this GFMMN is trained for the appropriate value of  $\theta$ . The min-max values of all the hyperboxes are quantized and these are expressed in the form of rules. Each rule represent the the kind of patters falling in that hyperbox. These rules are readable and represents the trained network. Experiments are conducted on eight different benchmark datasets obtained from UCI machine learning repository. These results prove the applicability of the proposed method.

## Keywords

Neural Network, Bayesian Classifiers, Fuzzy Systems, Support Vector Machines

## 1. INTRODUCTION

Data mining (DM) is the process of extracting the meaningful knowledge from huge databases [1]. Alternatively, it is also known as knowledge discovery in databases (KDD). It is an application area that can provide significant competitive advantages for making the right bussiness decisions. Any specific data mining algorithm is usually an instantiation of the model preference search components. The more common functions in the current data mining process include the following [2].

- (1) Classification: It classifies a data sample into one of the several predefined classes.
- (2) Regression: It maps a data sample to a real-valued prediction variable.
- (3) Clustering: It maps a data sample into a cluster, which are homogenous groupings of data samples based on similarity metrics or probability density models.

- (4) Association rules: It describes association relationship among different data samples.
- (5) Summarization: It gives a compact description for a subset of data.
- (6) Dependency modelling: It describes the important dependencies among variables.
- (7) Sequence analysis: It models the sequential patterns like time-series analysis. In addition to this, the goal is to model the state of the process that generates the sequence and also to extract and report deviations or trends over time.

The proposed work presented in this thesis focuses on classification task of data mining. Classification is one of the most frequent decision making tasks performed by human beings. A classification problem occurs when an object needs to be assigned to a predefined class based on the number of observed attributes related to that object [3]. Classification problems involve assigning a class  $C_i$  from a predefined set of classes  $C = \{C_1, C_2, \dots, C_M\}$  to an object which is described as a point in a certain feature space. The problem of designing a classifier is to find a optimal mapping in the sense of a certain criterion that determines the classifier performance [4]. The final goal is of designing a classifier that assigns a class label with the smallest possible error across the whole feature space.

There are number of techniques that perform the classification like  $k$ -nearest neighbor classifiers, Bayesian networks, decision trees, support vector machines, genetic classifiers, artificial neural networks, fuzzy systems, etc. Following subsections describe each of these techniques in brief.

### 1.1 $k$ -Nearest Neighbor Classifiers

The  $k$ -nearest-neighbor ( $k$ -NN) classification is one of the most fundamental and simple classification method. This classifier is preferred when there is little or no prior knowledge about the distribution of the data [5].

It assumes all the instances are points in  $n$ -dimensional pattern space and a distance measure is needed to measure the closeness of the instances. It classifies an instance by searching its  $k$  nearest neighbors and selecting the most popular class among the neighbors. The neighbors are taken from a set of instances for which the class is known. This set of instances can be thought of as the training set for the algorithm and no explicit training step is required. The drawback of this simple approach is the lack of robustness that characterize the resulting classifiers and also the accuracy of the  $k$ -

NN algorithm can be severely tainted due to the presence of noisy or irrelevant features.

## 1.2 Bayesian Classifiers

Bayesian classifiers store the probabilistic summary for each class and this summary includes the conditional probability of each attribute value to the given class and also the probability of the class i.e. to use Bayesian decision theory a priori probabilities and the conditional probability density of all classes should be known [6, 7]. This data structure describes a single decision boundary through the instance space and when the algorithm encounters a new instance, it updates the probabilities stored with the specified class. This process is not dependent on the order of the training instances. For the given test instance, the classifier uses an evaluation function to rank the alternative classes based on their probabilistic details and assigns the instance to the highest scoring class.

Both of the evaluation function and the summary descriptions used in Bayesian classifiers suppose that all the attributes in the dataset are statistically independent on each other. But this seems unrealistic for many natural domains; many researchers have concluded that the algorithm will behave poorly in comparison to other induction methods.

## 1.3 Decision Trees

The decision tree is an example of a predictive model and it can be viewed as a tree. In decision tree, each branch of the tree is a classification question and the leaves of the tree are partitions of the dataset with their classification. The approach of decision tree is used in many areas because it has several advantages [8, 9].

In the late 1970s J. Ross Quinlan introduced a decision tree algorithm named iterative dichotomiser 3 (ID3) [10]. ID3 picks predictors and their splitting values based on the gain in information that the split or splits provide. The information gain is the difference of the amount of information that is needed for making a correct prediction before and after the split has been made.

A main drawback of decision tree is its instability and also the structure of the decision tree is very sensitive to the changes in the datasets. To overcome this problem, some scholars have suggested Fuzzy Decision Tree (e.g. FuzzyID3) [11, 12, 13] by utilizing the fuzzy set theory.

## 1.4 Support Vector Machines

A support vector machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given a labeled training data, the algorithm outputs an optimal hyperplane which categorizes new data samples. In the SVM, a  $m$ -class problem is converted into  $m$  two-class problems in which one class is separated from the remaining classes [14]. For each of these two class problem, the original input space is mapped into the multi dimensional dot product space called feature space and in this feature space, the optimal hyperplanes that maximizes the generalization ability is constructed.

## 1.5 Genetic Classifiers

A genetic algorithm (GA) is a search algorithm based on the mechanics of natural selection and natural genetics. It efficiently uses the historical information to obtain new search points with target enhanced performance. In each generation, a new set of artificial individuals is created by using the information from the best of the old generation. GA combines the survival of the fittest from the old

population with a randomized information exchange that helps to form new individuals with higher fitness than the old ones.

The implementation of GA has some issues for selecting the representation of chromosomes, population size, mutation rate, crossover or mutation operators, termination criteria, etc. In addition to this, GA suffers with the drawback of having limited scalability.

## 1.6 Artificial Neural Networks

Artificial neural networks (ANNs) are considered as simplified mathematical models of human brain and they function as parallel distributed computing networks. In this, computations are performed by a dense mesh of computing nodes and connections. These nodes operate collectively and simultaneously on all the input data and due to this they are considered as a synchronous system that gives possibility to speed up the calculations. The basic processing elements of neural networks are called artificial neurons, or simply neurons. Sometimes they are also called the nodes. These nodes perform as summing and nonlinear mapping junctions.

## 1.7 Fuzzy Systems

Fuzzy set and logic theory [16] is one of the most prominent tools to handle uncertainty in decision-making. The major advantages of fuzzy system models are their robustness and transparency. Fuzzy system modelling achieves robustness by using fuzzy sets which incorporates imprecision to system models. Also unlike some system models like neural networks, the fuzzy system models are highly descriptive.

Fuzzy logic and ANN are considered as complimentary rather than competitive [17, 16]. This is because when the neural networks and fuzzy systems are combined together, they join their advantages and reduce their individual drawbacks. ANNs introduce its learning ability in the fuzzy systems and receive the interpretation and clear way of knowledge representation from fuzzy systems. In this way the disadvantages of the fuzzy systems are compensated by the capabilities of the neural networks and vice versa. Due to this reason the combination of the flexibility of fuzzy sets and the computational efficiency of ANNs has created a great amount of interest in pattern recognition problems [18, 19].

The fuzzy min-max neural network (FMMN) proposed by Simpson [20] is a special type of neuro fuzzy system that has high efficiency compared to the other machine learning methods [21]. FMMN utilizes fuzzy sets as pattern classes and each fuzzy set is an aggregate of fuzzy set hyperboxes. Each fuzzy set hyperbox is an  $n$ -dimensional box defined by min and max point with a corresponding membership function (MF). FMMN is used in wide range of applications [22, 23, 24]. It has the capability of learning only in few passes and it is online-adaptive.

Simpson [25] has proposed fuzzy min-max clustering neural network (FMMCN). The pattern clusters are implemented as fuzzy sets using a MF with a hyperbox core that is constructed from a min point and a max point. The min-max points of hyperboxes are determined by learning. He has presented results of clustering using Fisher Iris data.

Gabrys and Bargiela [26] have proposed the general fuzzy min-max neural network (GFMMN) that can be used for clustering and classification. This method uses supervised and unsupervised learning within a single learning algorithm. They have presented limitations of hyperbox membership function proposed by Simpson [20, 25] and proposed the new MF. Additional features of the algorithm are; (i) input patterns can be fuzzy hyperboxes or crisp points in the pattern space, (ii) hyperbox expansion constraint has been modified

and (iii) the parameter regulating the maximum hyperbox size can be changed adaptively.

In [27], fuzzy hyperline segment neural network (FHLSNN) is proposed and its performance is found superior than the FMMN algorithm. In [28], fuzzy hypersphere neural network (FHSNN) is presented wherein hypersphere instead of hyperbox is used to represent the fuzzy subset in the  $n$ -dimensional pattern space.

An improved version of basic FMMN based on the adaptive resolution classifier (ARC) technique is proposed in [29]. Unlike FMMN, in this method the training result is independent on pattern's presentation order and also the hyperbox expansion is not limited by a fixed maximum size.

In FMMN, covering of the training data with hyperboxes is constrained to have their boundary surfaces parallel to the coordinate axes of the chosen reference system. In [30], a more precise covering of each data cluster is obtained by rotating the hyperboxes with a suitable local principal component analysis with the intention of arranging the hyperboxes orientation in any direction of the data space.

The drawback of FMMN of having low automation degree is removed in [31] and the two novel learning algorithms for fuzzy min-max neural classifiers are proposed namely the ARC and its pruning version (PARC). ARC/PARC is characterized by a high automation degree and agrees to achieve networks with a remarkable generalization capability.

The algorithm proposed in [32] improves the FMMN performance by putting the thresholds on the dimension of the hyperboxes and sensitivity parameters. Due to this the classification result does not depend on the presentation order of the patterns in the training set, and at each step, the classification error in the training set cannot increase.

In [33], two new algorithms for classification and regression that are based on Simpson's FMMN are proposed.

A weighted FMM neural network presented in [34] assigns a weight value to each of the dimensions of each hyperbox so that membership can be assigned considering not only the occurrence of patterns but also the frequency of occurrences within that dimension.

One of such method is given in [35]. It uses a combination of exclusion and inclusion hyperbox sets to approximate the complex topology of the data. In this, the inclusion hyperbox is the type of hyperboxes that we have considered so far and represents data belonging to a class.

To overcome with the problem of overlapping area, the concept of compensatory neurons (CNs) which are created dynamically is introduced in fuzzy min-max neural network with compensatory neuron (FMCN) [36]. Because of this more number of hyperboxes are created and this leads to a complex network structure and wastage of time.

A fuzzy minmax neural network based on data core (DCFMMN) is given in [37]. A new MF is defined which considers the effect of noise in the data, the geometric centre and the data core of the hyperbox. In DCFMMN, instead of using the contraction process, a new kind of neuron called overlapped neuron with its MF based on the data core is proposed. Each of the overlapped neuron represents an overlapping area of hyperboxes belonging to different classes.

Although the FMCN and DCFMMN makes the use of special neurons to deal with overlapping problem efficiently, these methods cannot classify a high percentage of samples that are located in overlapping regions correctly and also have some structural problems in their learning algorithms. These problems lead to increased complexity and reduced efficiency. To overcome these problems, a multi-level fuzzy min-max (MLF) classifier is proposed in [38].

Although all above mentioned neural networks gives high classification accuracy, the knowledge gained by them is in terms of a large number of numerical values and complex network architectures. This knowledge is in a form that is not easily interpretable by humans beings [39]. To solve this problem, many rule extraction techniques have been developed which gives the rules that portrays the humanly understandable representation for neural networks [40]. In this paper the rule extraction technique from general fuzzy min-max neural network given in [26] is proposed.

The organization of the paper is as follows. Section 2 describes the general fuzzy min-max neural network. Section 3 gives the rule extraction from GFMMN. Experimental results and discussions are given in Section 4. Finally, the concluding remarks are given in Section 5.

## 2. GENERAL FUZZY MIN MAX NEURAL NETWORK

Following Subsections describe the general fuzzy min-max neural network used for classification proposed by Gabrys et al:

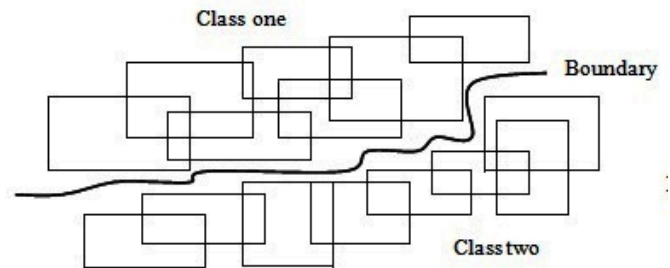


Fig. 1. 2-D hyperboxes created for two class-classification problem

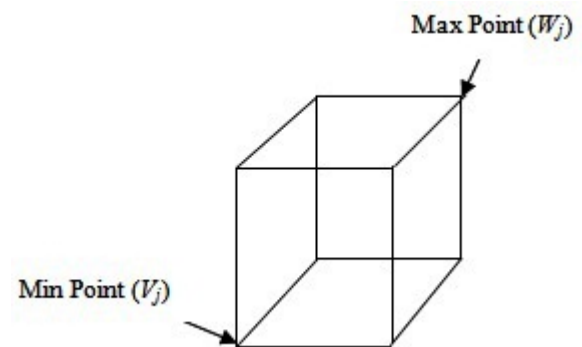


Fig. 2. 3-D hyperbox structure

### 2.1 General Overview

The general fuzzy min-max classification neural network builds the decision boundaries by creating fuzzy subsets of the  $n$ -dimensional pattern space as given in Figure 1. Each of these subset is called the hyperbox which is characterized by a pair of min-max points and the MF. The 3-dimensional hyperbox structure with its min-max

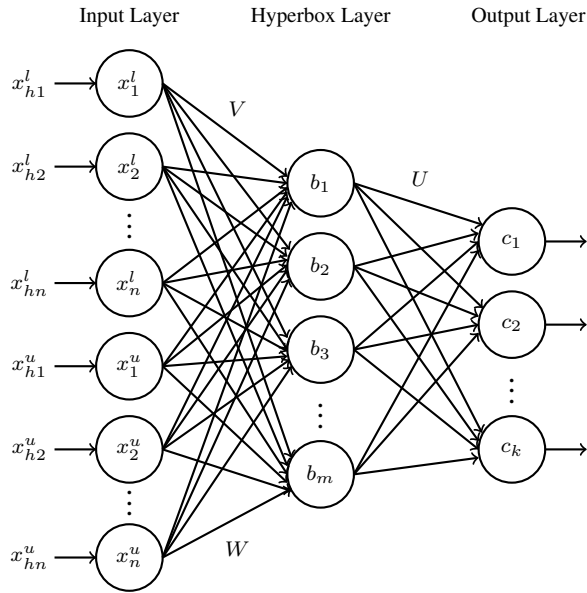


Fig. 3. GFMMN architecture

points is shown in Fig. 2.

Mathematically each hyperbox  $B_j$  is defined by

$$B_j = \{X_h, V_j, W_j, f(X_h, V_j, W_j)\} \quad (1)$$

where  $X_h = \{x_1, x_2, \dots, x_n\}$  is the  $h^{th}$  input pattern,  $V_j = \{v_{j1}, v_{j2}, \dots, v_{jn}\}$  is the min point,  $W_j = \{w_{j1}, w_{j2}, \dots, w_{jn}\}$  is the max point and  $f(X_h, V_j, W_j)$  is the MF.

The MF  $f(X_h, V_j, W_j)$  for hyperbox  $B_j$  is represented as  $b_j(X_h)$  and its value is in the range  $0 \leq b_j(X_h) \leq 1$ , measures the degree to which  $X_h$  falls outside of hyperbox  $B_j$ . Its value is always one when pattern  $X_h$  falls inside or on boundaries of  $B_j$  and decreases as the pattern goes away from that hyperbox.

Due to the fact that the fuzzy membership function proposed by Simpson and used in FMMN algorithms can assign a relatively high membership value to an input pattern which is quite far from the class prototype, Gabrys *et al.* have proposed new membership function that is defined as

$$b_j(X_h) = \min_{i=1 \dots n} (\min([1 - f(x_{hi}^u - w_{ji}, \gamma_i)], [1 - f(v_{ji} - x_{hi}^l, \gamma_i)])) \quad (2)$$

where

$$f(r, \gamma) = \begin{cases} 1 & \text{if } r\gamma > 1 \\ r\gamma & \text{if } 0 \leq r\gamma \leq 1 \\ 0 & \text{if } r\gamma < 0 \end{cases}$$

where  $X_h = [X_h^l, X_h^u]$  is the  $h^{th}$  input pattern in a form of lower,  $X_h^l$ , and upper,  $X_h^u$ , limits vectors contained within the  $n$ -dimensional unit cube,  $V_j = \{v_{j1}, v_{j2}, \dots, v_{jn}\}$  is the min point for  $B_j$ ,  $W_j = \{w_{j1}, w_{j2}, \dots, w_{jn}\}$  is the max point for  $B_j$ , and  $\gamma_i$  is the sensitivity parameter that controls how fast the membership values decreases as the distance between  $X_h$  and  $B_j$  increases.

## 2.2 GFMMN Architecture

The architecture of GFMMN consists of three layers of nodes, as shown in Fig. 3. This architecture grows adaptively to meet the

demands of the problem. It consists of an input layer ( $F_A$ ), hyperbox layer ( $F_B$ ) and the output layer ( $F_C$ ). The input layer has  $2*n$ -input nodes, two for each of the  $n$ -dimensions of input pattern. The output layer contains nodes equal in number to the number of classes. The hyperbox layer which is also called the hyperbox layer wherein, each node represents a hyperbox fuzzy set. All  $F_A$  to  $F_B$  connections are the min-max points. The  $F_B$  layer transfer function is the hyperbox MF defined in (2). The min and max points are stored in matrices  $V$  and  $W$  respectively. The min point matrix  $V$  is applied to the first  $n$ -input nodes representing the vector of lower bounds of the input pattern and the max point matrix  $W$  is applied to the other  $n$ -input nodes representing the vector of upper bounds of the input pattern. The connections between the  $F_B$  and  $F_C$  nodes are binary-valued and are stored in matrix  $U$ . The equation for assigning the values from  $F_B$  to  $F_C$  connections is

$$u_{ji} = \begin{cases} 1 & \text{If } B_j \text{ is a hyperbox of class } c_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $B_j$  is the  $j^{th}$  node in the hyperbox layer with  $j = (1, 2, \dots, m)$  and  $c_i$  is the  $i^{th}$  node in the output layer with  $i = (1, 2, \dots, k)$ . The output of the  $F_C$  node represents the degree to which input pattern  $X_h$  fits for class  $c_i$ . The transfer function of each  $F_C$  node performs the fuzzy union operation on the appropriate hyperbox fuzzy values which is defined as

$$c_i = \max_{j=1}^m b_j u_{ji} \quad (4)$$

where  $b_j$  is the membership value of  $X_h$  to the  $j^{th}$  hyperbox and  $u_{ji}$  is the binary value as defined in (3). Equation (4) gives the membership value of pattern  $X_h$  to each of the  $i^{th}$  class. This is the soft classification decision given by GFMMN and can be used wherever required. If hard decision is required then the maximum value among all values of  $c_i$  is set to unity.

## 2.3 Learning in GFMMN

Learning in GFMMN creates the collection of hyperboxes. Prior to learning only input and output layer exists and the hyperbox layer is constructed during learning by adding hyperboxes one by one. When each input pattern  $X_h$  is presented, it is checked for the possible expansion of the existing hyperbox of the same class as that of pattern  $X_h$ . If expansion is not possible then new hyperbox is created. This continues till all patterns are finished. So the learning process consists of four steps: initialization, expansion, overlaps test and contraction. These are summarized as below

- (1) Initialization: Before training starts  $V_j$  and  $W_j$  are initialized with

$$V_j = 0 \quad \text{and} \quad W_j = 0. \quad (5)$$

When  $j^{th}$  hyperbox  $B_j$  is adjusted for the first time using the input pattern, the min and max points of this hyperbox would be

$$V_j = X_h^l \quad \text{and} \quad W_j = X_h^u. \quad (6)$$

- (2) Expansion: In this step the membership values of input pattern  $X_h$  to all the hyperboxes of same class as that of  $X_h$  are calculated. The hyperbox with the highest membership value,  $B_j$ , is selected and tested for possible expansion. The expansion test is given by

$$\forall_{i=1 \dots n} (\max(w_{ji}, x_{hi}^u) - \min(v_{ji}, x_{hi}^l)) \leq \theta \quad (7)$$

where  $0 \leq \theta \leq 1$  is the user defined threshold value that determines the maximum size of the hyperbox. The larger value of  $\theta$  gives the small number of hyperboxes and small value gives the large number of hyperboxes. If the expansion condition for hyperbox  $B_j$  is satisfied then it is expanded by adjusting min and max points in each dimension  $i$  by using the equations

$$v_{ji}^{new} = \min(v_{ji}^{old}, x_{hi}^l) \quad \forall i = 1, 2, \dots, n, \quad (8)$$

$$w_{ji}^{new} = \max(w_{ji}^{old}, x_{hi}^u) \quad \forall i = 1, 2, \dots, n. \quad (9)$$

If neither of the existing hyperboxes can expand to include the input pattern, then a new hyperbox is created.

- (3) Overlap Test: After expansion of the hyperbox  $B_j$ , its overlap is tested with each different class hyperbox  $B_k$ . For every dimension  $i$ , if any one of the following four case is satisfied then overlap exists between two hyperboxes. With  $\delta^{old} = 1$  initially, the four test cases and the corresponding minimum overlap value for the  $i^{th}$  dimension are as follows.

$$\begin{aligned} \text{Case 1 : } & v_{ji} < v_{ki} < w_{ji} < w_{ki}, \\ & \delta^{new} = \min(w_{ji} - v_{ki}, \delta^{old}). \end{aligned} \quad (10)$$

$$\begin{aligned} \text{Case 2 : } & v_{ki} < v_{ji} < w_{ki} < w_{ji}, \\ & \delta^{new} = \min(w_{ki} - v_{ji}, \delta^{old}). \end{aligned} \quad (11)$$

$$\begin{aligned} \text{Case 3 : } & v_{ji} < v_{ki} \leq w_{ki} < w_{ji}, \\ & \delta^{new} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old}). \end{aligned} \quad (12)$$

$$\begin{aligned} \text{Case 4 : } & v_{ki} < v_{ji} \leq w_{ji} < w_{ki}, \\ & \delta^{new} = \min(\min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{old}). \end{aligned} \quad (13)$$

If  $\delta^{old} - \delta^{new} > 0$ , then  $\Delta = i$  and  $\delta^{old} = \delta^{new}$ . If not, the testing stops and the minimum overlap index variable is set to indicate that the next contraction step is not necessary, i.e.  $\Delta = -1$ .

- (4) Contraction: If overlap between the hyperboxes exists and  $\Delta$  be the selected dimension for contraction, then by using the minimal disturbance principle these hyperboxes are adjusted by the following four cases corresponding to the overlap occurred.

$$\begin{aligned} \text{Case 1 : } & v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}, \\ & w_{j\Delta}^{new} = v_{k\Delta}^{new} = \frac{w_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}. \end{aligned} \quad (14)$$

$$\begin{aligned} \text{Case 2 : } & v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}, \\ & w_{k\Delta}^{new} = v_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}. \end{aligned} \quad (15)$$

$$\begin{aligned} \text{Case 3a : } & v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta} \\ & \text{and } (w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta}), \\ & v_{j\Delta}^{new} = w_{k\Delta}^{old}. \end{aligned} \quad (16)$$

$$\begin{aligned} \text{Case 3b : } & v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta} \\ & \text{and } (w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta}), \\ & w_{j\Delta}^{new} = v_{k\Delta}^{old}. \end{aligned} \quad (17)$$

$$\begin{aligned} \text{Case 4a : } & v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta} \\ & \text{and } (w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta}), \\ & w_{k\Delta}^{new} = v_{j\Delta}^{old}. \end{aligned} \quad (18)$$

$$\begin{aligned} \text{Case 4b : } & v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta} \\ & \text{and } (w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta}), \\ & v_{k\Delta}^{new} = w_{j\Delta}^{old}. \end{aligned} \quad (19)$$

### 3. RULE EXTRACTION FROM GFMMN

All the techniques of rule extraction from ANN are grouped into three approaches namely decompositional, pedagogical and eclectic. In the decompositional approach, the activation values and weights of the hidden layers of the neural network are analyzed. The pedagogical approach treats the ANN as a black box and extract the rules only in terms of input and output relationships that are understood by the ANN. Not all of the rule extraction techniques fit in exactly one of these two categories, so there is a third category called the eclectic approach which is the hybrid category and it is based on internal architecture and/or weight vectors in a trained ANN and input-output relationships.

The proposed classification model aims to extract the rules from the GFMMN using eclectic category where each hyperbox layer node is analyzed and the rules stating the global relationships between input and output of the GFMMN are extracted. As the GFMMN is trained using continuous and discrete attributes, the rule antecedents are formed by using the conditions for both continuous and discrete attributes.

For each hyperbox  $B_j$ , the min-max values are used to form the rule conditions. Min-max values of all the hyperboxes are quantized and due to quantization of the min-max values, the rules can be described in words rather than real numbers and such rules are more readable. The rule conditions of all attributes are connected by logical 'and' operator to form an antecedent of *if-then* rules.

A quantization level  $Q$  is defined as the number of feature values in the quantized rules. For example, with  $Q = 3$ , feature values are described as low, medium or high in the fuzzy rules. Quantization by round-off distributes,  $Q$  quantization points evenly, with one at each end point given by

$$V_q = \frac{q-1}{Q-1} \quad (20)$$

where  $q = 1, 2 \dots Q$ . This method then rounds off a min-max values to the nearest  $V_q$  value.

### 4. EXPERIMENTATION RESULTS AND DISCUSSION

The experiments are conducted on eight different datasets found in the UCI machine learning repository. The details of these datasets regarding the number and type of attributes and number of patterns can be obtained from the website of UCI machine learning repository [41]. All these datasets are the good mixture of small number of attributes, large number of attributes, continuous attributes and discrete attributes etc. These datasets includes credit, iris, bupa liver

disorder, heart disease, wine, thyroid, ionosphere and glass dataset. Experiments are conducted on each of these datasets for  $\theta = 0.1$  to 1 with the step size of 0.01. Initially to observe the learning capability, percentage accuracy values for all of these datasets are recorded by training them with 50% data and testing with the same 50% data.

The minimum, maximum and average values of percentage accuracy and number of hyperboxes are recorded in Table 4 and Table 4 respectively.

Table 1. Minimum, maximum and average percentage accuracy for 50% train-100%test data

Dataset	GFMMN		
	Min	Max	Avg
Credit	78.26	78.84	78.72
Iris	94	97.33	95.58
Liver	55.94	81.74	62.23
Heart	80	81.85	81.14
Wine	86.52	91.57	89.58
Thyroid	82.79	85.58	83.64
Ionosphere	58.40	96.29	82.25
Glass	71.49	56.07	61.34

Table 2. Minimum, maximum and average number of hyperboxes for 50% train-100%test data

Dataset	GFMMN		
	Min	Max	Avg
Credit	262	334	276.9
Iris	3	39	10.3
Liver	2	106	21.4
Heart	94	135	107.6
Wine	3	89	228.4
Thyroid	3	38	10.2
Ionosphere	42	160	85.7
Glass	6	59	19

The rules are extracted for the maximum accuracy of 97.33% of the iris dataset. This accuracy is obtained for the  $\theta = 0.06$  with total 34 hyperboxes. After quantization of min-max values of these hyperboxes, 16 duplicate rows were found. Rules for the remaining 18 hyperboxes are defined as given in Table 4. These rules are pruned and it gives only four rules as given in Table 4.

## 5. CONCLUSION

General fuzzy min-max neural network has the capability to learn the input data in a very few passes and when it is combined with rule extraction then it can be easily adapted in sensitive applications. In this paper, we have proposed the rule extraction method from general fuzzy min-max neural network. These rules justify the classification decision given by GFMMN. For this, GFMMN is trained for the input data and resulting min-max values are quantized. Due to the quantization of the numeric min-max values, rules are expressed in terms of words and these rules are more understandable.

Table 3. Unpruned rules extracted from GFMMN for iris dataset with 100% train-100% test data size and  $\theta = 0.1$

Rules
1. If a1=Low and a2= Medium and a3=Low and a4=Low then Class=1
2. If a1=Low and a2=Low and a3=Low and a4=Low then Class=1
3. If a1=Low and a2= Low and a3=Medium and a4=Medium then Class=2
4. If a1=Low to Medium and a2=Medium and a3=Low and a4=Low then Class=1
5. If a1=Medium to High and a2=High and a3=Low and a4=Low then Class=1
6. If a1=Medium and a2=Medium and a3=Low and a4=Low then Class=1
7. If a1=Medium and a2= High and a3=Low and a4=Low then Class=1
8. If a1=Medium and a2= Medium and a3=Medium and a4=Medium then Class=2
9. If a1=Medium and a2=Low and a3=Medium and a4=Medium then class=2
10. If a1=Medium and a2=Low to Medium and a3=Medium and a4=Medium then class=2
11. If a1=Medium and a2=Medium and a3=Medium and a4=Medium to High then class=3
12. If a1=Medium and a2=Low and a3=Medium and a4=High then class=3
13. If a1=Medium and a2=Medium and a3=Medium and a4=High then class=3
14. If a1=Medium and a2=Medium and a3=Medium and a4=Medium3
15. If a1=Medium and a2=Low and a3=Medium and a4=Medium then class=3
16. If a1=Medium and a2=Medium and a3=High and a4=Medium then class=3
17. If a1=Medium and a2= Medium and a3=High and a4=High then class=3
18. If a1=Medium and a2=Medium and a3=High and a4=High to Medium then Class=3

Table 4. Pruned rules extracted from GFMMN for iris dataset with 100% train-100% test data size and  $\theta = 0.1$

Rules
1. If a3=Low then Class=1
2. If a4=Medium and a2=Medium then Class=3
3. If a2=Medium then Class=3
otherwise class=2

## 6. REFERENCES

- [1] J. Han, M. Y. Kamber, and S. C. Lee, *Data mining: concepts and techniques*, San Francisco, CA, USA: Morgan Kaufmann, 2001.
- [2] W. Chen and Y. K. Du, Using neural networks and data mining techniques for the financial distress prediction model, *Expert Systems with Applications*, vol. 36, pp. 40754086, 2009.
- [3] C. Lin, I. Chung and C. Chen, "An entropy-based quantum neuro-fuzzy inference system for classification applications," *Neurocomputing*, vol. 70, pp. 2502-2516, 2007.
- [4] O. Cordon, M. Jesus and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, pp. 21-45, 1999.
- [5] N. K. Patil, V. S. Malemath, and R. M. Yadahalli, "Color and Texture Based Identification and Classification of food Grains using different Color Models and Haralick features," *Int. J. Computer Science & Engineering*, ol. 2, no. 12, pp. 3669-3680, 2011.
- [6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, second ed., Wiley, 2001.
- [7] P. Langely, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," *AAAI-92 Proceedings*, pp. 223-228, 1992.
- [8] *Decision Tree*, Wikipedia, the free encyclopaedia 2005: <http://en.wikipedia.org/wiki/Decisiontree>.
- [9] *The federal budget execution process decision tree*, 2005: <http://www.knownet.hhs.gov>.

- [10] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann, San Mateo, CA., 1993.
- [11] M. Umamo, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita, "Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems," *3rd IEEE conf on Fuzzy Systems*, pp. 2113-2118, 1994.
- [12] R. L. Chang and T. Pavlidis, "Fuzzy decision tree algorithms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, no. 1, pp. 28-35, 1977.
- [13] X. Z. Wang, D. S. Yeung, and E.C.C. Tsang, "A comparative study on heuristic algorithms for generating fuzzy decision trees," *IEEE Trans. Systems, Man and Cybernetics*, vol. 31, no. 2, pp. 215-226, 2001.
- [14] S. Abe, *Pattern classification: Neuro-fuzzy methods and their comparison*, Springer Verlag, 2001.
- [15] Zak, "Neural model of underwater vehicle dynamics," *J. of Mathematics and Computers in Simulation*, vol. 1, no. 2, pp. 189-195, 2007.
- [16] L. A. Zadeh, *Fuzzy logic, neural network and soft computing*, Commun. ACM, vol. 37, pp. 77-84, 1994.
- [17] J. Zupan, *Introduction to artificial neural network (ANN) methods: What they are and how to use them*, Department of Chemistry, University Rovira Virgili, Tarragona, Spain Acta Chimica Slovenica, pp. 327-352, 1994.
- [18] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Trans. Neural Networks*, vol. 3, pp. 698-713, 1992.
- [19] S. C. Newton, S. Pemmaraju, and S. Mitra, Adaptive fuzzy leader clustering of complex data sets in pattern recognition, *IEEE Trans. Neural Networks*, vol. 3, pp. 794-800, Sept. 1992.
- [20] P. K. Simpson, "Fuzzy min-max neural network - Part I: classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776-786, 1992.
- [21] A. Joshi, N. Ramakrishnan, E. N. Houstis, and J. R. Rice, "On neurobiological, neuro-fuzzy, machine learning, and statistical pattern recognition techniques," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 18-31, Jan. 1997.
- [22] A. V. Nandedkar, "An interactive shadow detection and removal tool using granular reflex fuzzy min-max neural network," in *Proc. World Congr. Eng.*, vol. 2, 2012.
- [23] M. Seera, C. P. Lim, D. Ishak, and H. Singh, "Fault detection and diagnosis of induction motors using motor current signature analysis and a hybrid FMM CART model," *IEEE Trans. Neural Networks and Learn. Syst.*, vol. 23, no. 1, pp. 97-108, Jan. 2012.
- [24] N. P. Jawarkar, R. S. Holambe, and T. K. Basu, "Use of fuzzy min-max neural network for speaker identification," in *Proc. ICRTIT*, pp. 18-182, 2011.
- [25] P. K. Simpson, "Fuzzy min-max neural network - Part II: clustering," *IEEE Trans. Fuzzy System*, vol. 1, pp. 32-45, 1993.
- [26] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. Neural Networks*, vol. 11, pp. 769-783, 2000.
- [27] U. V. Kulkarni, T. R. Sontakke, and G. D. Randale, "Fuzzy hyperline segment neural network for rotation invariant handwritten character recognition," in *Proc. Int. Joint Conf. on Neural Networks: IJCNN'01*, Washington DC, USA, pp. 2918-2923, July 2001.
- [28] U. V. Kulkarni and T. R. Sontakke, "Fuzzy hypersphere neural network classifier," in *Proc. IEEE Conference on Fuzzy Systems held at University of Melbourne*, Australia, December 2001.
- [29] A. Rizzi, M. Panella, and F. M. F. Mascioli, A recursive algorithm for fuzzy min-max networks, in *Proc. IEEE/INNS/ENNS Int. Joint Conf. Neural Netw.*, vol. 6. Como, Italy, Jul. 2000, pp. 541-546.
- [30] A. Rizzi, F. M. F. Mascioli, and G. Martinelli, Generalized min-max classifier, in *Proc. 9th IEEE Int. Conf. Fuzzy Syst.*, vol. 1. San Antonio, TX, May 2000, pp. 3641.
- [31] A. Rizzi, M. Panella, and F. M. F. Mascioli, Adaptive resolution min-max classifiers, *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 402-414, Mar. 2002.
- [32] M. Meneganti, F. S. Saviello, and R. Tagliaferri, Fuzzy neural networks for classification and detection of anomalies, *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 846-861, Sep. 1998.
- [33] R. Tagliaferri, A. Eleuteri, M. Meneganti, and F. Barone, Fuzzy min-max neural networks: From classification to regression, *Soft Comput.*, vol. 5, no. 6, pp. 697-706, Feb. 2001.
- [34] H. J. Kim, J. Lee, and H. S. Yang, A weighted FMM neural network and its application to face detection, in *Neural Information Processing* New York, NY, USA: Springer-Verlag, pp. 1771-1786, 2006.
- [35] A. Bargiela, W. Pedrycz, and M. Tanaka, "Exclusion/inclusion fuzzy classification network," *J. Knowledge-Based Intelligent Information and Engineering Systems*, vol. 2773, pp. 1236-1241, 2003.
- [36] A. V. Nandedkar and P. K. Biswas, "A fuzzy min-max neural network classifier with compensatory neuron architecture," *IEEE Trans. Neural Networks*, vol. 18, pp. 42-54, 2007.
- [37] H. Zhang, J. Liu, D. Ma, and Z. Wang, "Data-core-based fuzzy minmax neural network for pattern classification," *IEEE Trans. Neural Networks*, vol. 22, no. 12, pp. 2339-2352, Dec. 2011.
- [38] Reza Davtalab, Mir Hossein Dezfoulian, and Muharram Mansoorizadeh, "Multi-level fuzzy min-max neural network classifier," *IEEE Trans. Neural Networks*, vol. 25, no. 3, pp. 470-482, March 2014.
- [39] W. Duch, R. Adamczak, and K. Grabczowski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules," *IEEE Trans. Neural Networks*, vol. 12, pp. 277-306, 2001.
- [40] S. H. Huang and H. Xing, "Extract intelligible and concise fuzzy rules from neural networks," *Fuzzy Sets and Systems*, vol. 132, pp. 233-243, 2002.
- [41] UCI repository of machine learning databases, University of California, Irvine, <http://www.ics.uci.edu/mllearn/MLRepository.html>, 1998.