Iranian Journal of Fuzzy Systems

Volume 19, Number 5, (2022), pp. 147-165

Original paper

University of Sistan and Baluchestan

An improvement in integrating clustering method and neural network to extract rules and application in diagnosis support

V. D. Minh[1], T. T. Ngan[2], T. M. Tuan[3], V. T. Duong[4] and N. T. Cuong[5]

[1,4,5] *University of Industry, 298 Cau Dien street, Bac Tu Liem District, Hanoi, Viet Nam*
[2,3] *Faculty of Computer Science and Engineering, Thuyloi University, 175 Tay Son, Dong Da, Hanoi, Vietnam*

minhvd@haui.edu.vn, ngantt@tlu.edu.vn, tmtuan@tlu.edu.vn, duongvt@haui.edu.vn, cuongnt@haui.edu.vn

Abstract

Most of chronic liver diseases without suitable treatment will lead to cirrhosis of the liver, eventually progressing to liver cancer. Thus, early diagnosis is very important in detecting the liver diseases and suggesting the treatment at the right time. A useful model that effectively predicts the patient's liver fibrosis has great importance in reducing the load on doctors, especially in lower-level hospitals. In this paper, a new model combining semi-supervised learning method and fuzzy min max neural network with selective fuzzy rule set rendering is proposed. Cirrhosis level is evaluated by APRI and FIB-4. The proposed method is experimented on data sets from machine learning databases, including UCI and CS. Apart from that, our method is also implemented on the liver data set collected from the hospitals of Thai Nguyen province. The comparison among our proposed method and other related ones is also given. The obtained results show that our proposed model has better performance than compared methods in terms of execution time and the number of rules.

Keywords: Artificial neural network, semi-supervised clustering, chronic liver diseases, liver disease diagnosis, cirrhosis.

1 Introduction

Liver performs many metabolic functions in human body [6], such as filters the blood, and most importantly, detoxifies [10]. Chronic hepatitis is one of the causes of serious illness in the world [14]. Chronic hepatitis has various phenomena with persistent inflammation and necrosis of hepatocytes [7]. Most chronic liver diseases lead to cirrhosis of the liver and progress to cirrhosis and liver cancer. Assessment of cirrhosis is essential in indications for treatment, monitoring and prognosis of chronic hepatitis [12]. This makes an important contribution to reducing the rate of progression to cirrhosis and liver cancer. APRI (Aminotransferase-to-platelet ratio index) [25] and FIB-4 (Fibrosis-4 index) [22] are two simple, inexpensive, and fast techniques that can be done by any medical institution to assess cirrhosis level. Many scientists and doctors are interested in health care and research on intelligent techniques used in diagnosing and treating liver diseases [20, 27]. Aman Singh and his team have conducted a study on the use of intelligent techniques in diagnosing liver diseases [21]. In [1], an expert system based on fuzzy rules to assist in diagnosing liver disease was proposed by Prateek Agrawal et al.. However, the authors have not compared the proposed model with other related models yet. A new method of diagnosis of focal liver damage based on CT (Computerized Tomography) imaging was proposed [4]. A 2-stage model, consisting of (1) exploiting three-dimensional decision rule and (2) CT lives imaging judgment, was introduced. The results of second phase support doctors in liver cancer diagnosing. However, CT imaging could not detect cirrhosis of the liver at early stage. The techniques used in this research are difficult to implement in local hospitals because of the high cost and the need of a qualified specialist.

Artificial neural networks have applied in medicine [2]. One of effective methods is Fuzzy min-max neural network (FMNN) [18, 19]. FMNN was applied into various problems in medical diagnosis and prediction because of the ability

Corresponding Author: T. M. Tuan
Received: August 2021; Revised: March 2022; Accepted: August 2022.
https://doi.org/10.22111/IJFS.2022.7162

to generate very simple "*if ... then*" fuzzy rules [17, 26]. Each of these fuzzy rules was described by quantifying the min-max values of the input properties on the FMNN. Moreover, FMNN is also an approach in disease diagnosing support [9, 16, 26] such as: the combination of Fuzzy C-Means (FCM) with FMNN to support lung cancer detection [5]; FMNN hybrid algorithm in heart disease diagnosis [13]; Optimization of FMNN using genetic algorithm (GA) in the diagnosis of liver disease assistance [23]. Based on the original FMNN model, many studies have suggested mechanisms to improve the performance of FMNN, for example, modifying the size of hyperbox (HB) [8, 15]; cuting off some unimportant HBs, having a low usage index [17]; generating new HBs to manage the overlapping area instead of the overlap adjustment; presenting a method of HB expansion based on the idea of K-Nearest Neighbor algorithm called K-Nearest hyperbox [11]. Moreover, semi-supervised learning method in FMNN was one of the improvements that dramatically improved network performance [23, 24]. This method combined the advantages of supervised and unsupervised learning in FMNN.

As mentioned above, a set of HBs is the result of the training process of FMNN. Based on these HBs, a set of rules is generated. Therefore, the large number of HBs leads to a large number of fuzzy rules. Several matching models have been proposed to overcome this limitation by eliminating HBs with low usage scores [23, 26].

A disadvantage in most of mentioned methods is that training data must include labels of all samples. Thus, these methods are unable to apply into the problems that data set contains unlabeled samples. In fact, the cost of labeling the data is often very high. To overcome this shortcoming, FMM neural network combined to semi-supervised learning in disease diagnosis support systems was introduced. In [23], Tran et al. proposed SCFMN (Semi-supervised Clustering in Fuzzy Minmax Neural network) model with the capability of reducing the number of HBs in order to optimize FMNN. This model used a part of the labeled samples in the first stage to monitor the clustering in the second stage. However, in the second stage of SCFMN, a method similar to stripping algorithm was applied. This increased the complexity of the algorithm.

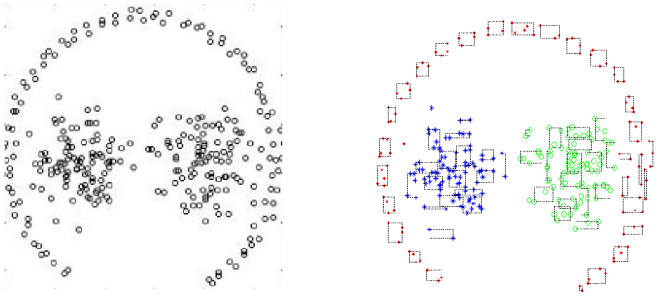
To reduce the time consuming and make the model more stable, in this research, we improve SCFMN model mainly in training process. New model is shorten as MSCFMN (Modified SCFMN). This model is applied in early diagnosis of liver cancer prevention in patients with cirrhosis. The cirrhosis level is defined by using APRI and FIB-4 indices.

The main contributions of this paper include (i) improving learning methods to optimize SCFMN; (ii) integrating the determine of additional information into classification process to make more reliable decisions; (iii) applying proposed model to generate a set of rules without using full labeled data and (iv) evaluating the efficiency of proposed method on UCI data sets and a practical data set consisting of patients' liver enzyme tests at the hospitals in Thai Nguyen province.

The rest of this paper includes the sections as follow. Section 2 presents some related fundamental knowledge. Section 3 introduces the details of proposed algorithm. Section 4 describes the experiments different datasets. The last section states the discussions and conclusions of the paper.

2 The structure of FMNN

FMNN integrates fuzzy min-max theory and artificial neural networks. The training in FMNN will create HBs. The identified data samples depend on the degree of its membership function related to the corresponding metadata. Figure 1 is an example of the partitioning of HBs on Pathbased dataset from the machine learning datasets (CS).



(a) Original data points (b) The HBs covering data set points

Figure 1: Illustration of the partitioning of HBs on Pathbased set: (a) is a graphic of the distribution of samples, and (b) is an image of the HBs obtained after FMNN training.

2.1 Fuzzy hyperbox

Each HB in FMNN is determined by the max and min points. The dimension of each dimension of the symbol hyperbox is $\theta(\theta \in [0,1])$. Figure 2 illustrates a HB in a 2-dimensional sample space.

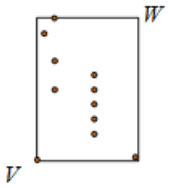


Figure 2: Illustration of HB in 2-dimensional sample space.

Let the j^{th} hyperbox fuzzy set, B_j, is a set of 4 ingredients, be defined by (1):

$$B_j = \{A_h, V_j, W_j, b_{(A_h,B_j)}\}, (j = \overline{1,k}), \tag{1}$$

where:

- $V_j$is the min point and W_j is the max point of the B_j . These values will be calculated during network training;
- $A_h = (a_{h1}, a_{h2}, ..., a_{hn}) \in I^n, (h = (\overline{1,m})$ is the h^{th} sample of training set $D(m = |D|)$;
- $b_{(A_h,B_j)}$ is the membership function or dependency of sample A_h to B_j $(0 \leq b_{(A_h,B_j)} \leq 1)$.

The membership function b_j is used to determine the belonging of a data sample with the corresponding HB, determined by (2):

$$b_{(A_h,B_j)} = \frac{1}{n}\sum_{i=1}^{n}[1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)], \tag{2}$$

where $f(x,y)$ is determined by (3):

$$f(x,y) = \begin{cases} 0, & \text{if } x \times y < 0 \\ x \times y, & \text{if } 0 \leq x \times y \leq 1 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

The sensitivity γ is used to adjust the degree of degradation when the samples are removed away from the HB. Figure 3 presents the effects of γ on b_j.

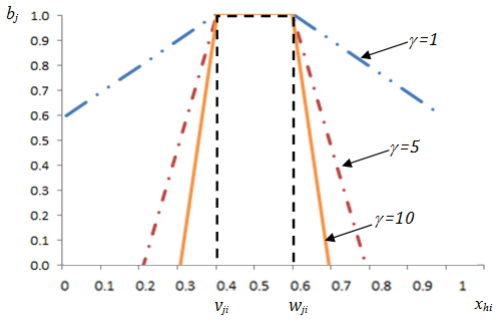


Figure 3: Illustration of the effects of γ on b_j.

2.2 The architecture of the FMNN

Figure 4 and Figure 5 show the FMNN architecture representations for classification (consisting of two layers) [19] and clustering (consisting of three layers) [18] of data, respectively.

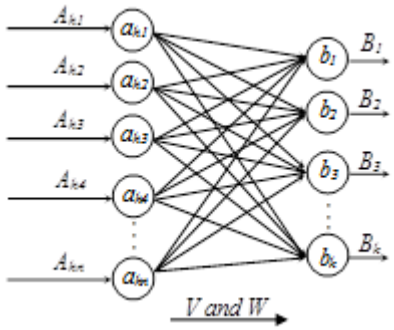


Figure 4: Architecture of FMNN for clustering.

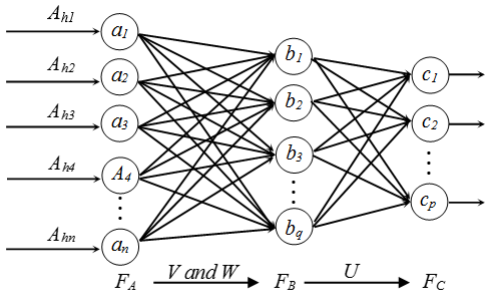


Figure 5: Architecture of FMNN for classification.

2.3 Learning algorithm in FMNN

Learning algorithms in FMNN include the adjustment (expansion, contraction) of HBs in sample space. The process of modifying the HBs depends on the data samples properties in the training set.

Let D be the input training dataset consisting of m data samples. A_h is the h^{th} data sample in D, B is the set of HBs at the output ($k = |B|$) .

The learning algorithm begins by traversing data samples in the training dataset. For each input patterns, the learning process finds and expands the HB that satisfies the expansion condition. If no HB satisfies the expansion condition, this process creates a new HB. The problem is that when extending the HB, there is an overlap among the other HBs that are grouped together. When this happens, the learning algorithm will adjust min and max points of HBs to eliminate the overlap. The process of creating new HB allows to develop new clusters without retraining. The learning algorithm in FMNN includes 3 following steps:

Step 1: Initialize the HB

Step 2: Expand the HB and check for overlap between HBs

Step 3: HB contraction.

Steps 2 is repeated for each input sample in the training set. The algorithm stops when the HBs are stable or all data samples in the training set are trained.

2.4 Generating the *if ... then ...* decision fuzzy rules from hyperboxes

Each HB in FMNN was extracted as a *if ... then ...* fuzzy rule, based on the quantitative value of the min-max points of each HB using fuzzy quantitative method [3]. The assigned inputs correspond to a quantum point according to (4):

$$A_q = \frac{q-1}{Q-1}, \quad q = 1, 2, ..., Q. \tag{4}$$

The *if ... then ...* fuzzy rule is defined by (5):

$$\text{Rule } R_j: \text{ If } x_{p1} \text{ is } A_q \text{ and } \quad \text{and } x_{pn} \text{ is } A_q \text{ then } x_p \text{ is class } C_j \text{ with } CF = CF_j, \tag{5}$$

where:

- m is the total number of HBs;
- $j = \overline{1,m}$;
- $x_p = (x_{p1}, ..., x_{pn})$ is an input vector of n dimensions;
- A_q is the premise value, it is calculated according to formula (4);
- CF_j is value of confidence factor of j^{th} HB;
- C_j is the j^{th} output class.

2.5 Pruning hyperboxes

In the definition of the *if ... then ...* fuzzy rule (5), CF_j is the confidence factor of j^{th} HB. It means that each hyperbox B_j is rated for quality based on confidence factor (CF). The closer to 1 the value of CF_j is, the higher the reliability is. The confidence factor of B_j is calculated by (6):

$$CF_j = (1 - \varphi)U_j + \varphi A_j, \tag{6}$$

where:

- U_j is the proportion of the number of HBs in B_j to the total number of HBs in classes including B_j;
- $j = \overline{1,m}$;
- A_j is the ratio between the number of samples that are classified correctly using B_j and total number of samples that are correctly classified in the same class;
- $\varphi \in [0, 1]$ is the weight.

Pruning HBs has two purposes, including (i) to eliminate the low-confidence HBs, and (ii) to generate the small, high-efficiency rule sets. HBs with the confidence factor greater than a threshold are used to define decision rules. Each of rules has the form of an *if ... then ...* rule.

3 Modified semi-supervised clustering in fuzzy min-max neural network model

In this section, a new model called Modified Semi-supervised Clustering in Fuzzy Min-max Neural network, denoted as MSCFMN, is presented. MSCFMN model inherits the advantages of SCFMN and overcomes the shortcomings of SCFMN in [23] as well.

3.1 The general framework of MSCFMN model

In our previous research [23], we proposed the model namely as SCFMN consisting of 2 phases. In phase 1, SCFMN received labels from users for samples near the center of each cluster. In phase 2, SCFMN propagated the labels that were assigned in phase 1 through the HBs to train the input patterns. These HBs will generate the fuzzy rules that support the medical diagnosis.

Herein, an improvement of SCFMN model is proposed. This model consists of 4 phases:

- **Phase 1.** *Train the input data to find the HB*: Each HB corresponds to a cluster. The HBs identified in this phase are the basis for determining additional information for the next stages of the algorithm. The obtained results of Phase 1 are used as the input of Phase 2.

- **Phase 2.** *Select additional information*: The data samples in the training dataset with full dependence on the HBs will receive the additional information. This additional information is the labels that correspond to the HB indices. Then, the training data set D is split into two subsets D_1 and D_2. D_1 is the input of Phase 3 and D_2 is the input of Phase 4.

- **Phase 3.** *Create the set of HBs for labeled samples*: The HBs are obtained from training progress of the patterns in dataset D_1. In which, the HBs with the same label will belong to the same cluster.

- **Phase 4.** *Create the set of HBs for unlabeled samples*: Create another set of HBs (U) from training the samples in the dataset D_2 base on set G (result of Phase 3).

3.2 Main steps of MSCFMN

The general diagram of the learning algorithm is separated into the different phases of the MSCFMN model. The flowcharts of these phases are presented in Figures 6–9 below.

Phase 1:

Phase 1 (Figure 6) trains the input data samples in dataset D, and creates HBs. In this stage, data samples that have a full membership function with the HBs in set G will be labeled. The label of input sample is the label of the HB corresponding to that sample. The set of this kind of samples is denoted as D_1. Samples without a full membership function to any of the HBs are not labeled, denoted as $D_2(D = D_1 \cup D_2)$.

Phase 1 consists of 3 steps. Each step is described in detail as follows:

Step 1. Hyperbox initialization: Before training, MSCFMN must to know the total number of HBs that can be defined. Therefore, the learning algorithm needs to create a HB set for the new HBs (denoted by U). The number of HBs selected in this step is large enough, possibly up to $\sqrt{n}$, $(n = |D|)$ [28]. HBs in U with max point (W) and min point (V) are defined by (7), (8):

$$V_j = \underline{1},\ \ \underline{1} = (1, 1, ..., 1) \in I^n, \tag{7}$$

$$W_j = \underline{0},\ \ \underline{0} = (0, 0, ..., 0) \in I^n, \tag{8}$$

Step 2. Hyperbox expansion: The HB satisfying conditions (9) and (10) is found for each input sample:

$$b_j = max\{b_j : j = 1, ..., k\}, \tag{9}$$

$$\theta_{(A_h,B_j)} \leq \theta_{\max}, \tag{10}$$

where:

- $\theta_{\max}$ is the size of the HB. This is also an input parameter of the algorithm;
- $\theta_{(A_h,B_j)}$ is the size of hyperbox B_j (after being extended to include the input pattern A_h). $\theta_{(A_h,B_j)}$ is calculated by:

$$\theta_{(A_h,B_j)} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(|\max(a_{hi}, w_{ji})| - |\min(a_{hi}, v_{ji})|)^2}. \tag{11}$$

If existing any B_j satisfies (9) and (10), then this HB is expanded.

If not any B_j satisfies (9) and (10), then a new HB is created. For doing this, a new HB is moved from set U to set B. The max and min points of the new HB (called H_{new}) are calculated according to (12):

$$W_{H_{new}} = V_{H_{new}} = A_h. \tag{12}$$

Step 3. Hyperbox overlap testing and contraction: The learning algorithm performs an overlap testing whenever a HB is expanded in step 2 by comparing each dimension of the HBs. Assume that B_k is a HB that has just been expanded in step 2. A HB B_j ovelaps with B_k on the i^{th} dimension if one of 10 cases mentioned in [19] happens.

If there is the overlap between two HBs, the necessary adjustments are done to remove the overlaps based on 10 situations considered in hyeperbox contraction step in [23].

Step 2 and Step 3 are repeated for each sample in training dataset until clusters are stable.

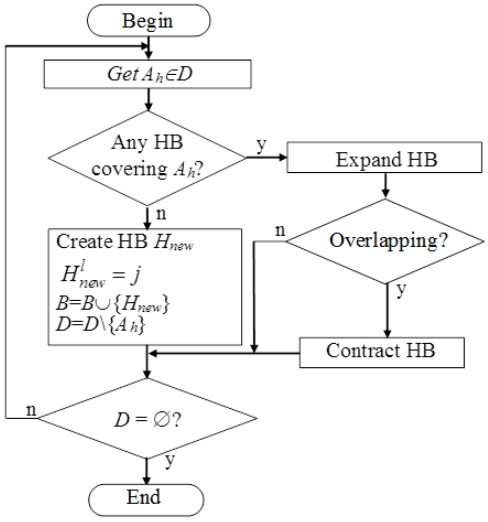


Figure 6: Phase 1. Train the input data to find the HBs.

Phase 2:

Phase 2 of the MSCFMN determines samples that can be labeled from the user in the training dataset D. The set D is split into two subsets D_1 (labeled samples), and D_2 (samples without labeled). The data samples in the training set with a full membership function to the HB will receive additional information. Phase 2 includes 2 following steps:

Step 1. Determine $b_{(A_h,B_j)}$ by (9).

Step 2. Labeled the input patterns $A_h \in D$:

+ If $b_{(A_h,B_j)} = 1$ then $A^l_h = 1$, $D_1 = D_1 \cup A_h$;

+ If $b_{(A_h,B_j)} = 0$ then $A^l_h = 0$, $D_2 = D_2 \cup A_h$ (A^l_h is the label of A_h).

The steps of Phase 2 is given as in Figure 7 below.

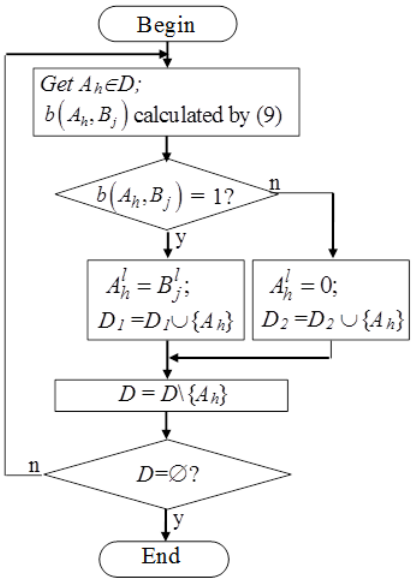


Figure 7: Phase 2. Determining additional information.

Phase 3 Phase 3 (Figure 8) groups the samples in D_1 into clusters. These samples are included to create HB set G. Generated HBs will have labels according to the labels of samples in D_1. The training process is similar to the training process in FMNN [18].

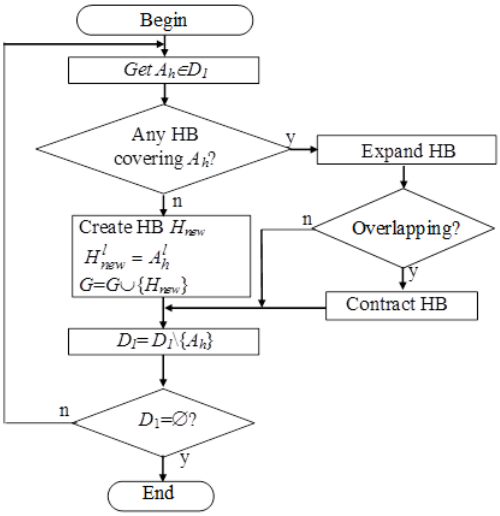


Figure 8: The diagram of Phase 3 in MSCFMN.

Phase 4

The general framework of Phase 4 is given as in Figure 9 below.

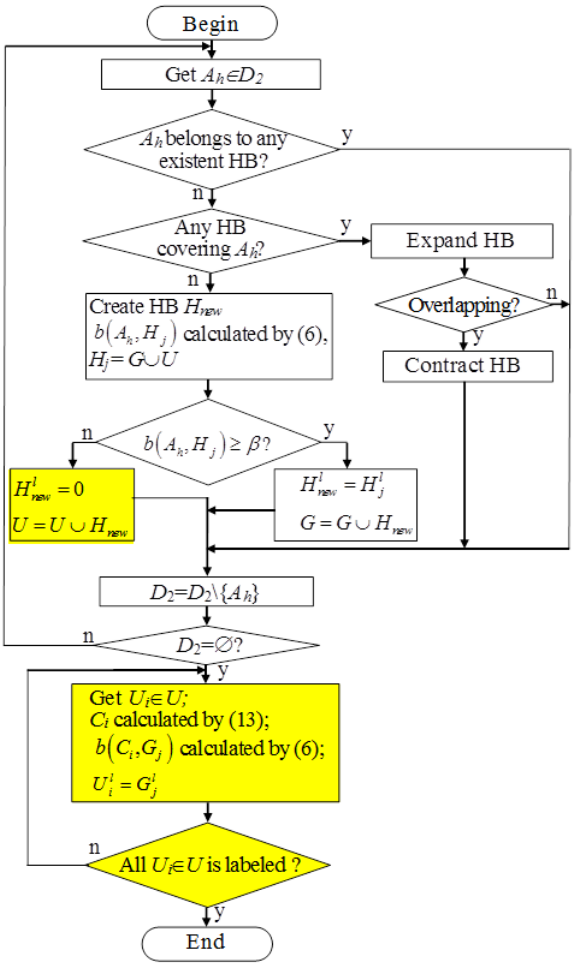


Figure 9: The framework of Phase 4 in MSCFMN.

Phase 4 trains data in D_2 to create the HB set U and propagates the labels from G to U to define the clusters for the HBs in U. The steps are described in detail as below:

Step 1. Expand the HB: For each pattern, the algorithm finds an HB satisfying (9) and (10):

- If there are any B_j satisfying (9) and (10), B_j is expanded.
- If not any B_j satisfy (9) and (10), a new HB (called H_{new}) is created. When creating H_{new}, two following cases may occur:
  + If Hnew satisfies (13), H_{new}) is added into G (H_{new}) participates in the training of set G in the next steps);
  + Otherwise, H_{new} is added into U.

$$b_{(C_{H_{new}},G_j)} \geq \beta, G_j \in G, \tag{13}$$

where:

- $b_{(C_{H_{new}},G_j)}$ is calculated by (2); (the membership degree of to hyperbox G_j);
- $C_{H_{new}}$ is the center of H_{new}. It is calculated by (14);
- β is the threshold parameter. The value of β is intended to prevent the algorithm from labeling H_{new}, with HBs created from unlabeled samples and low membership function. These HBs will be labeled in step 3 of the algorithm.

$$C_{H_{new}} = \frac{V_{H_{new}} + W_{H_{new}}}{2}. \tag{14}$$

Step 2. Check the overlap among HBs and contract HBs (if necessary): The overlap testing is performed whenever a HB is expanded in step 1. This process is done as same as in step 3 of Phase 1. Set of HBs overlapping includes $G \cup U$.

Step 3. Assign labels to HB $U_j \in U$:

1) C_{U_j} defined by (14) is center of U_j for each $U_j \in U$;
2) Find $G_k \in G$ with $b(C_{U_j}, G_k)$ satisfying (9);
3) Label U_j by the label of G_k.

The steps 1, 2, 3 in this phase are performed for each HB in U. The framework of Phase 4 is presented as in Figure 9.

The differences between the MSCFMN and the SCFMN are shown in the highlighted blocks in Figure 9. Moreover, the comparison between SCFCM and MSCFMN is also presented in Table 1 below.

The complexity of MSCFMN algorithm:

The MSCFMN algorithm consists of 4 phases, performed from phase 1 to phase 4, respectively. The time complexity (denoted T) is calculated by 15):

$$T = (T_1 + T_2 + T_3 + T_4), \tag{15}$$

where: T_i is the time complexity of i^{th} phase, i = 1, 2, 3, 4.

Denote that $M = |D|, K = |B|$ and n is the dimension of data samples. The complexity of MSCFMN is presented in Table 2.

Table 1: The comparison between MSCFMN and SCFMN

#Phase	SCFMN	MSCFMN
1	Train the input data to find the HBs. This is the stage to identify additional information for monitoring and guidance in the subsequent phases.	
2	Select additional information. This stage is responsible for selecting data samples under the supervision and guidance of the information identified in phase 1.	
3	Create the set of HBs for labeled samples. This stage performs the training of the dataset that has been added with additional information (D_1) to create set of HBs (called labeled HBs).	
4	Create the set of HBs for unlabeled samples. This stage performs training on the data set in Phase 2 (D_2) to generate new sets of HBs. These sets of HBs combined with HBs obtained from Phase 3 are the training results of the input data set (D).	
	The algorithm uses a threshold value β to decide whether to allow the generation of new HB for each input samples A_h that does not satisfy the HB expansion condition. Samples with a threshold value less than β are ignored and have to wait for the next iteration. This process needs many iterations to go through all the data in the data set D_2.	All input samples $A_h (A_h \in D_2)$ that do not belong to any HB or do not satisfy the HB expansion condition are created new HBs. The algorithm uses a threshold value β to decide which new HBs should be created. There are two cases of creating new HBs: (i) The HB will be created by using the label obtained by propagation from previously labeled HBs; and (ii) The new HB does not receive a label from any previous HB. In case, the label of this new HB is decided by calculating the center of the corresponding HB.

Table 2: The complexity of MSCFMN

#Phase	The complexity
1	$T_1 = M \times n \times K$
2	$T_2 = M \times n \times K$
3	$T_3 = M \times n \times K$
4	$T_4 = M \times n \times K$
Total	$T = O(M \times n \times K)$

4 Experimental results

Herein, our proposed model MSCFMN is implemented on different data sets. MSCFMN is compared to some related models, including SCFMN [23], FMM-GA [26] and FMM-CF [17].

4.1 Data description

The experiments of mentioned methods are performed on various data sets. The details of these data sets are shown in 3.

Table 3: Information on training datasets

ID	Data set	Number of objects	Number of attributes	Number of clusters
1	Thyroid	7200	21	3
2	Iris	150	4	3
3	PID	768	8	2
4	Sonar	208	60	2
5	Wine	178	13	2
6	Flame	240	2	2
7	Jain	373	2	2
8	R15	600	15	2
9	Liver	9000	10	2

The Liver dataset contains the results of 9,000 patients assigned to a subclinical examination to test liver function. This data set contains only information of liver enzyme disorders from selected patients. There are 2 groups in this data set. Group 1 includes 4,000 samples with cirrhosis and group 2 includes 5,000 samples without cirrhosis.

The list of attributes in Liver data set includes Age, Gender, Aspartat transaminase (AST), Gamma Glutamyl Transferase (GGT), Albumin, Total Bilirubin (TB), Direct Bilirubin (DB), Upper Level of Normal (ULN), Platelet counts (PLT). These indices were selected based on the APRI index calculation [25], FIB-4 [22] and the evaluation method of the doctors. APRI is calculated using equation (16), and FIB-4 is calculated using equation (17) below.

$$APRI = \frac{AST/ULN}{PLT} \times 100. \tag{16}$$

$$FIB-4 = \frac{Age \times AST}{PLT \times ALT}. \tag{17}$$

Experimental data sets are standardized before conducting the experiments. The missing values are handled similarly to Batista. It means that all missing values are the average of values on the corresponding attribute as in (18).

$$A_{h_j} = \frac{1}{m}\sum_{i}^{m} A_{ij}, \tag{18}$$

where A_{h_j} is the missing value of the j^{th} attribute of the h^{th} sample.

Available models including FMM-CF, FMM-GA required that all samples in training set are labeled. Whereas SCFMN and MSCFMN used a part of unlabeled samples in training set.

4.2 Validity indices

In our experiments on the Benchmark data sets, two evaluation indicators including Accuracy (Acc) and the number of HBs ($\#HB$) are used to compare four related models, including FMM-CF, FMM-GA, SCFMN and MSCFMN.

On Liver datasets, used validity indices are Accuracy (Acc), Sensitivity (Sen), Specificity ($Spec$), Negative predictive value (NPV), Positive Predictive Value (PPV), time consuming ($Time$) and number of HB ($\#HB$). The values of these indices are determined by the formula from (19) to (23) respectively:

$$Acc = \frac{a+d}{a+b+c+d}. \tag{19}$$

$$Sen = \frac{a}{a+c}. \tag{20}$$

$$Spec = \frac{d}{d+b}. \tag{21}$$

$$PPV = \frac{a}{a+b}. \tag{22}$$

$$NPV = \frac{d}{c+d}, \tag{23}$$

where a, b, c, d are values determined as in Table 4.

Table 4: Information about predictive values

Result of clustering	**Positive**	**Negative**
Positive	a	b
Negative	c	d

4.3 Evaluation results of MSCFCM on liver dataset

The experiments are performed three times. In each time, 1,300 samples were randomly selected from each group (cirrhosis or without cirrhosis) to generate a data set consisting of 2,600 samples with complete information. In order to remove HBs, 50% of the samples in the training set was used. The training and testing progresses are run 5 times and swapped out after each run. Then, the average value of 15 runs is calculated. As mentioned above, FMM based methods (FMM-CF, FMM-GA) use all labeled data while semi-supervised based methods (SCFMN, MSCFMN) use both labeled and unlabeled data. The parameters are set as $CF = 0.45, \gamma = 20, \theta_{max} = 0.3, \beta = 0.8$.

Table 5 shows the values of validity indices of MSFMN on Liver data set by applying FIB-4 and APRI techniques when changing θ_{max}.

Table 5: Values of validity indices when using MSCFMN when changing θ_{max}

θ_{max}	Technique	Acc	Sen	Spec	PPV	NPV	Time	#HB
0.1	FIB-4	**0.97**	**0.98**	**0.95**	**0.98**	**0.95**	1.02	67
	APRI	**0.94**	**0.96**	**0.9**	**0.96**	**0.9**	0.91	68
0.2	FIB-4	0.95	0.97	0.92	0.97	0.92	0.87	39
	APRI	0.93	0.94	0.91	0.96	0.88	0.79	38
0.3	FIB-4	0.94	0.96	0.9	0.96	0.9	0.73	27
	APRI	0.92	0.93	0.89	0.95	0.85	0.65	27
0.4	FIB-4	0.92	0.95	0.86	0.94	0.87	0.59	**14**
	APRI	0.91	0.93	0.86	0.94	0.84	0.53	**14**
0.5	FIB-4	0.88	0.91	0.79	0.91	0.8	0.48	21
	APRI	0.89	0.92	0.83	0.93	0.82	0.44	21
0.6	FIB-4	0.85	0.9	0.75	0.89	0.75	0.39	19
	APRI	0.87	0.91	0.79	0.91	0.78	0.35	19
0.7	FIB-4	0.83	0.88	0.7	0.88	0.72	0.31	17
	APRI	0.83	0.88	0.7	0.88	0.72	0.31	17
0.8	FIB-4	0.82	0.87	0.7	0.87	0.7	**0.26**	15
	APRI	0.82	0.88	0.7	0.87	0.71	**0.18**	15
0.9	FIB-4	0.81	0.87	0.68	0.87	0.69	0.27	16
	APRI	0.81	0.87	0.67	0.86	0.69	**0.18**	15

Based on the results in Table 3, the validity indices, including $Acc, Sen, Spec, NPV, PPV$ get the best values in the case of $\theta_{max} = 0.1$. The time consuming is lowest (0.26 seconds) when using *FIB*-4 at $\theta_{max} = 0.8$. When using *APRI*, the best value is 0.18 seconds obtained at $\theta_{max} \in \{0.8, 0.9\}$.

Figure 10 presents the comparison of values among validity indices when using FIB-4 and APRI techniques when $\theta_{max} = 0.1$.

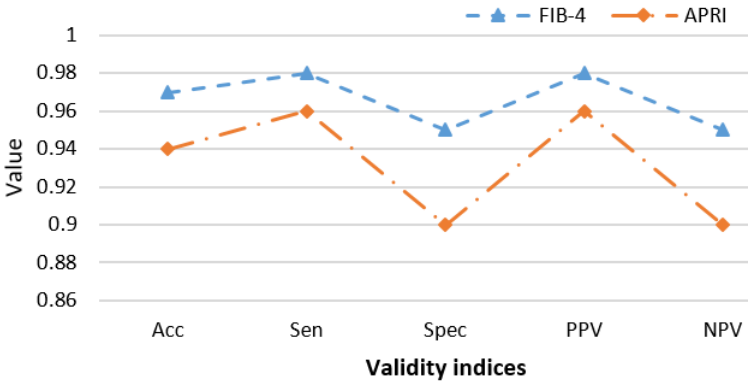


Figure 10: The validity indices of MSCFMN on Liver dataset with $\theta_{max} = 0.1$ using FIB4 and APRI.

As shown in Figure 10, when $\theta_{max} = 0.1$, the values of validity indices obtained by using *FIB*-4 are higher than using *APRI*. The comparison of time consuming when using two these techniques is also illustrated as in Figure 11 below.

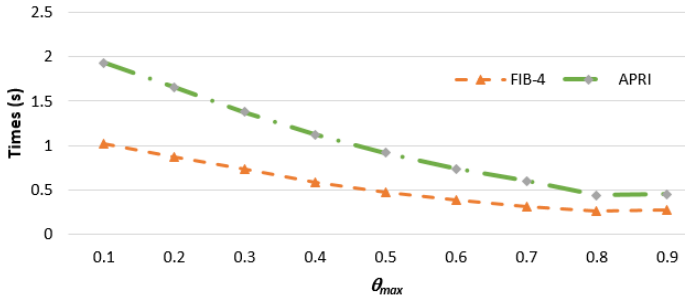


Figure 11: The training time of MSCFMN on Liver dataset using FIB4 and APRI when θ_{max} is changed

As shown in Figure 10 and Figure 11, Table 6 below describes information of the *if ... then ...* fuzzy rules of MSCFMN. There are 16 rules generated from a set of 16 HBs. These rules are used to support the diagnostics. Where A_1 is AST, A_2 is ULN, A_3 is PLT, and CF is the confidence factor. If the conclusion is $C = 1$, the diagnosis is positive (diseased). Otherwise ($C = 0$), the patient is diagnosed as negative (non-diseased). Quantitative values are set as 5 levels by 1 *very low*, 2 - *low*, 3 - *medium*, 4 - *high*, 5 *very high*.

Table 6: Prediction rules obtained from hyperboxes by using MSCFMN based on *APRI*

Rule	**If**			**Then**	**CF**
	A1	**A2**	**A3**		
R1	1	1	2-3	1	0.300
R2	1-3	1	2-3	1	0.114
R3	3	1	1-3	1	0.010
R4	1	1	1-2	1	0.016
R5	1-2	1	3-4	1	0.075
R6	1-2	1	4-5	1	0.023
R7	3	4	1	1	0.013
R8	1-2	1-2	1	1	0.010
R9	1	1	4	0	0.008
R10	3	1	3	1	0.003
R11	1	1	1	0	0.008
R12	3-4	1-2	1	1	0.039
R13	4	1	2	1	0.010
R14	4	5	1	1	0.003
R15	1-3	1-4	1-2	1	0.834
R16	1	1	1-4	0	0.43

Table 7 presents the *if ... then ...* fuzzy rules of MSCFMN after pruning all HBs with CF confidence less than 0.039 from Table 6.

Table 7: Prediction rules obtained from hyperboxes by using MSCFMN based on APRI with $CF \geq 0.039$

Rule	**If**			**Then**	**CF**
	A1	**A2**	**A3**		
R1	1	1	2-3	1	0.300
R2	1-3	1	2-3	1	0.114
R3	3	1	1-3	1	0.010
R12	3-4	1-2	1	1	0.039
R15	1-3	1-4	1-2	1	0.834
R16	1	1	1-4	0	0.43

Table 8 synthesizes the results predicted by MSCFMN. In this table, value 1 in the last column means that the patients have liver damage due to cirrhosis or hepatitis. The others are not disease infected (value 0 in the last column).

As given in Table 8, six out of nine selected cases are diagnosed as disease infected.

Table 8: The diagnosis results on 9 samples

If										Then
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
79	0	98.3	104.1	3.1	154.4	36.7	27.3	10.1	37	1
50	0	84.1	100.9	3.1	266.4	25.2	37.6	11.7	29	1
52	0	89.9	94.3	3.1	249.0	24.5	34.1	10.0	28	1
76	0	96.2	92.3	3.1	136.9	33.5	24.2	9.0	37	1
65	1	580.1	200.6	3.0	195.6	38.3	359.5	139.3	39	1
34	0	568.6	208.7	2.7	82.6	27.5	65.3	15.3	23	1
35	1	60.3	57.0	1.1	87.8	37.4	19.0	3.5	18	0
37	0	58.5	45.4	1.3	196.2	39.2	12.1	3.5	29	0
47	0	59.4	45.4	1.3	196.4	39.2	12.1	3.5	29	0

4.4 Comparison of MSCFCM with some other methods on Liver dataset

Using APRI technique, in this subsection, the comparison of experimental results obtained by applying MSCFMN and some other related models are presented. The methods used in this comparison include SCFMN [23], FMM-GA [26], FMM-CF [17]. All selected methods are implemented on Liver dataset and mentioned Benchmark datasets. On Liver dataset, all the properties on Table 2 are used in training process. The experiments use the "*k-fold cross-validation*" method, with $k = 10$ for evaluation. The results in following tables are the average of 10 runs.

Table 9 shows the values of accuracy by applying 4 methods on Liver data. The results showed that FMM-GA, SCFMN, MSCFMN have quite similar values. All three methods are better than FMM-CF. However, the total numbers of HBs when using FMM-CF and FMM-GA is higher than that of SCFMN and MSCFMN (shown in Table 10). This leads to the number of rules in diagnostic support using FMM-CF and FMM-GA is more than using SCFMN and MSCFMN.

Table 9: Accuracy (%) of four methods on Liver dataset when changing θ_{max}.

θ_{max}	**FMN-CF**	**FMN-GA**	**SCFMN**	**MSCFMN**
0.1	71.02%	88.77%	88.10%	88.45%
0.2	74.04%	89.14%	91.27%	90.35%
0.3	73.94%	94.74%	91.24%	90.76%
0.4	74.21%	90.44%	89.92%	90.37%
0.5	72.06%	87.68%	89.60%	88.41%
0.6	67.38%	87.78%	84.85%	85.62%
0.7	65.67%	83.30%	82.76%	81.01%
0.8	60.08%	77.94%	79.94%	77.92%
0.9	59.89%	75.34%	76.45%	76.56%

From the results in Table 9, accuracy of FMN-GA gets the highest values in five cases. The average value of accuracy on these 9 cases is 14.40%, 0.25%, 0.84% higher than FMN-CF, SCFMN and MSCFCM respectively. The values in Table 9 are visually presented as in Figure 12.

Apart from accuracy, the number of HBs created when applying 4 methods on Liver data is evaluated and presented in Table 10.

In fact, the higher values of θ_{max}, the larger the size of HBs is. Consequently, the number of HBs needing in the model is smaller. As shown in above table, the number of HBs in FMN-CF and FMN-GA get the lowest values at $\theta_{max} = 0.9$ while SCFCM and MSCFCM get the lowest number of HBs at both $\theta_{max} = 0.8$ and $\theta_{max} = 0.9$. Moreover, the number of HBs created by SCFCM and MSCFCM almost are the same. The number of HBs is compared in graphical form as in Figure 13.

To compare the computational time between two semi-supervised clustering approaches, SCFMN and MSCFMN are implemented on Liver data set. Figure 14 shows the run time of two these methods when using a same input set in training process.

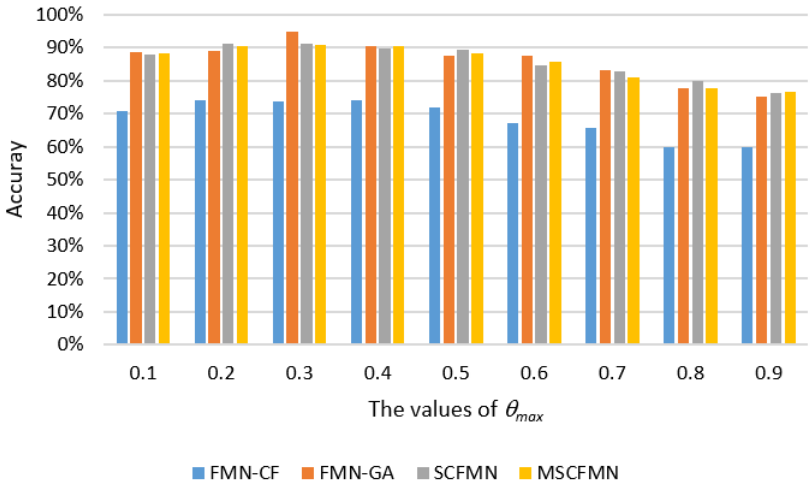


Figure 12: Comparison of accuracy among 4 models when θ_{max} is changed.

Table 10: The number of HBs created by 4 models.

θ_{max}	**FMN-CF**	**FMN-GA**	**SCFMN**	**MSCFMN**
0.1	90	62	40	41
0.2	70	48	30	31
0.3	57	35	21	22
0.4	41	29	16	16
0.5	26	15	11	11
0.6	18	12	7	7
0.7	15	9	5	5
0.8	11	8	4	4
0.9	8	7	4	4

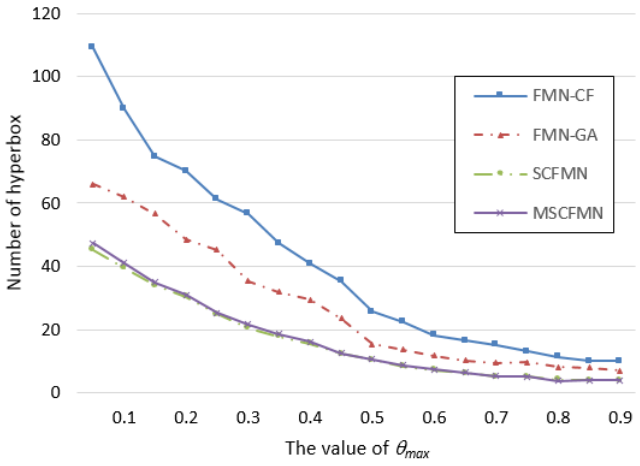


Figure 13: The comparison of FMM-CF, FMM-GA, SCFMN and MSCFMN in term of the number of HBs.

As in Figure 14, SCFMN takes more computational time than MSCFCM because SCFCM puts the unsatisfied samples back to training set. Then, these samples are retrained while each sample is trained once in MSCFCM. This increases the run time of SCFCM comparing with MSCFCM. Two these models are also implemented on 10 runs with different sets of input samples. The results of time consuming in every runtime are shown in Figure 15 below.

Figure 15 shows that MSCFMN is faster and more stable than SCFMN. It means that input samples have significantly affected to the SCFCM methods. As mentioned above, the unsatisfied samples have to been trained again until

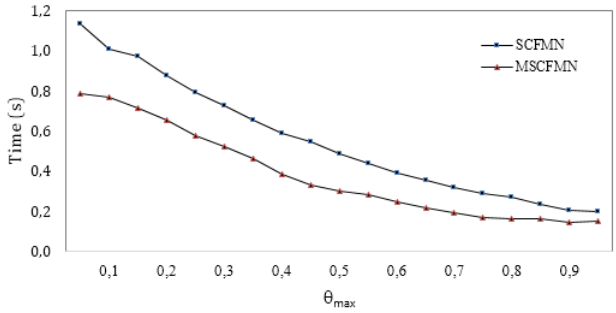


Figure 14: Training time of SCFMN and MSCFMN on Liver dataset when θ_{max} is changed.

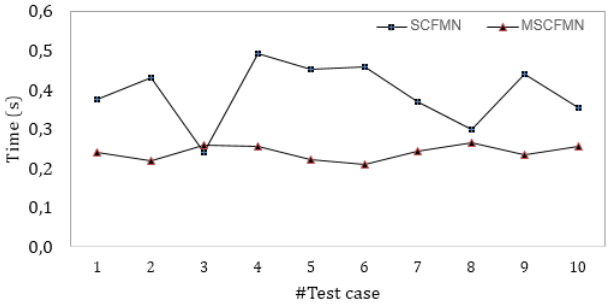


Figure 15: Training time of SCFMN and MSCFMN on Liver dataset with $\theta_{max} = 0.5$ on 10 different input sets.

satisfying. It leads to the time consuming depends on characteristics of the input samples, especially depending on the number of samples that are not satisfied. MSCFCM illuminates the retraining data. Each input data is trained once in MSCFCM. Hence, the number of unsatisfied samples in input set does not affect to the time consuming of training process of MSCFCM. This makes MSCFCM be more stable than SCFCM.

MSCFCM improves mainly in training process of SCFCM. Thus, the run time of MSCFCM and SCFCM in testing are the same. The results of testing on 10 different input sets from Liver data set are presented as in Fig 16 below.

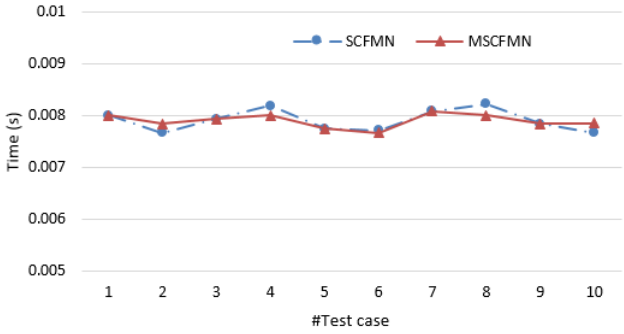


Figure 16: Testing time of SCFMN and MSCFMN on Liver dataset with $\theta_{max} = 0.5$ using 10 different input sets.

4.5 Evaluation four models on UCI data sets

On UCI data sets, the values of accuracy are presented in Table 11 below.

The results show that SCFCM achieves the highest accuracy on PID, Sonar, and Thyroid datasets. MSCFMN and FMM-GA get the best accuracy on Wine and Iris respectively. In Iris data set, the difference of accuracy obtained by MSCFMN and FMM-GA is quite small. However, the MSCFMN does not require labeled accompanying templates. This is a big advantage of MSCFMN. Apart from that, the numbers of HBs created by each of 4 models on these datasets are given in Table 12.

Table 11: Accuracy (%) of 4 models on UCI data sets.

Datasets	θ	**FMM-CF**	**FMM-GA**	**SCFMN**	**MSCFMN**
PID	0.6	62.11	70.44	**70.57**	70.12
Sonar	0.6	65.22	73.43	**73.91**	73.54
Thyroid	0.7	87.76	92.63	**92.65**	92.63
Wine	0.7	91.11	93.33	93.33	**93.86**
Iris	0.6	92.16	**95.42**	94.12	**94.67**
Flame	0.01	93.17	96.32	97.93	**97.95**
Jain	0.02	92.58	96.54	**98.72**	98.54
R15	0.015	93.89	97.83	99.50	**99.64**

Table 12: Experimental results of $\#HB$.

Datasets	θ	**FMM-CF**	**FMM-GA**	**SCFMN**	**MSCFMN**
PID	0.6	35	14	**10**	**10**
Sonar	0.6	35	10	**7**	**7**
Thyroid	0.7	33	19	**12**	13
Wine	0.7	18	10	**6**	**6**
Iris	0.6	12	**4**	**4**	**4**
Flame	0.01	80	40	31	**30**
Jain	0.02	90	45	**32**	**32**
R15	0.015	48	35	**30**	31

From the results in Table 12, the number of HBs generated by applying MSCFMN is equivalent to SCFMN method. These numbers are much less than those of other methods. Set the parameters as $CF = 0.45, \gamma = 20, \beta = 0.8$, the comparison of the number of HBs among 4 methods is given as in Figure 17 below.

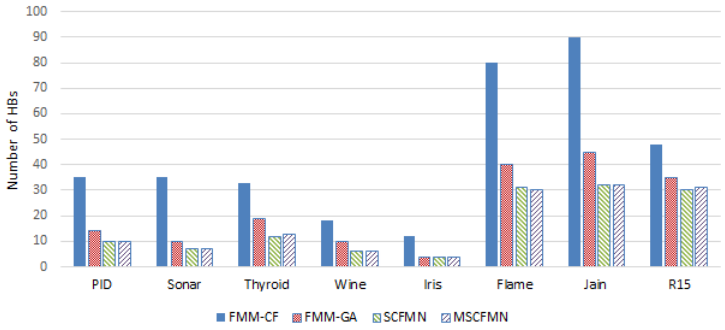


Figure 17: Comparison of the number of HBs among 4 models on UCI data sets $CF = 0.45, \gamma = 20, \beta = 0.8$.

5 Discussions and conclusions

Based on the previous research, in this paper, an improvement of the SCFCM model in training phase was proposed. The details of novel model (MSCFMN) are also given. Based on the detail description and experimental results, there are some main advantages of MSCFCM, such as:

- MSCFCM is able to define additional information in training process by itself.
- The results of clustering and run time in training process of MSCFCM are not affected by samples in input set.
- The number of rules obtained by MSCFCM is small with the similar clustering quality comparing with avaiable methods. These rules support to doctors in disease diagnosis.

To compare the performance of proposed model with other related models, some UCI data sets and a collected data set of Liver disease are used in experiments. The results show that MSCFCM creates the number of HBs the same as SCFCM but time-consuming of MSCFCM is less than SCFCM. Moreover, the runtime of MSCFCM in training process is more stable than that of SCFCM. Finally, the proposed model has ability to create smaller number of rules to support the diagnosis process of doctors with the same accuracy.

Our proposed method still has the limitation that is the quality and performance of the clustering process depends on parameters θ and the threshold β, even though the value of β is self-adapting in the algorithm.

For further researches, we are going to develop this model in determining the size of the HBs based on the input properties. Moreover, more information related to the patients' disease history is used to get the accuracy higher. Apart from that, the results of subclinical testing and image analysis will be involved to determine the final disease.

References

[1] P. Agrawal, V. Madaan, V. Kumar, *Fuzzy rule-based medical expert system to identify the disorders of eyes, ENT and liver*, International Journal of Advanced Intelligence Paradigms, **7**(3-4) (2015), 352-367.

[2] Z. A. Bulaghi, A. H. Navin, M. Hosseinzadeh, A. Rezaee, *World competitive contest-based artificial neural network: A new class-specific method for classification of clinical and biological datasets*, Genomics, **113**(1) (2021), 541-552.

[3] G. A. Carpenter, A. H. Tan, *Rule extraction: From neural architecture to symbolic representation*, Connection Science, **7**(1) (1995), 3-27.

[4] Y. Chen, X. Yue, H. Fujita, S. Fu, *Three-way decision support for diagnosis on focal liver lesions*, Knowledge-based Systems, **127** (2017), 85-99.

[5] K. S. Darne, S. S. Panicker, *Use of fuzzy C-mean and fuzzy min-max neural network in lung cancer detection*, International Journal of Soft Computing and Engineering (IJSCE), **3**(3) (2013), 265-269.

[6] M. Eslam, S. K. Sarin, V. W. S. Wong, J. G. Fan, T. Kawaguchi, S. H. Ahn, J. George, *The Asian pacific association for the study of the liver clinical practice guidelines for the diagnosis and management of metabolic associated fatty liver disease*, Hepatology International, (2020), 1-31.

[7] L. D. Jules, *Chronic Hepatitis*, Harrisons Gastroenterology and Hepatology, 17 th edit: Mc Graw Hill Medical, (2012), 390-414.

[8] T. T. Khuat, B. Gabrys, *A comparative study of general fuzzy min-max neural networks for pattern classification problems*, Neurocomputing, **386** (2020), 110-125.

[9] A. S. Kumar, A. Kumar, V. Bajaj, G. K. Singh, *Class label altering fuzzy min-max network and its application to histopathology image database*, Expert Systems with Applications, **176** (2021), 114880.

[10] B. N. Li, C. K. Chui, S. Chang, S. H. Ong, *A new unified level set method for semi-automatic liver tumor segmentation on contrast-enhanced CT images*, Expert Systems with Applications, **39**(10) (2012), 9661-9668.

[11] M. F. Mohammed, C. P. Lim, *Improving the fuzzy min-max neural network with a K-nearest hyperbox expansion rule for pattern classification*, Applied Soft Computing, **52** (2017), 135-145.

[12] M. Ney, S. Li, B. Vandermeer, L. Gramlich, K. P. Ismond, M. Raman, P. Tandon, *Systematic review with meta-analysis: Nutritional screening and assessment tools in cirrhosis*, Liver International, **40**(3) (2020), 664-673.

[13] B. R. Rajakumar, A. George, *On hybridizing fuzzy min-max neural network and firefly algorithm for automated heart disease diagnosis*, In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), (2013), 1-5.

[14] A. B. Ryerson, S. Schillie, L. K. Barker, B. A. Kupronis, C. Wester, *Vital signs: Newly reported acute and chronic hepatitis C cases-United States, 2009-2018*, Morbidity and Mortality Weekly Report, **69**(14) (2020), 399.

[15] M. Seera, C. P. Lim, C. K. Loo, H. Singh, *A modified fuzzy min-max neural network for data clustering and its application to power quality monitoring*, Applied Soft Computing, **28** (2015), 19-29.

[16] T. Sharma, G. Kumawat, P. Chakrabarti, S. Poddar, T. Chakrabarti, A. M. Kamali, M. Nami, et.al., *Using artificial neural network and machine learning algorithms to scrutinize liver diseases*, (2021). DOI: 10.21203/rs.3.rs-324049/v1.

[17] S. Shinde, S. D. Waghole, M. M. Bare, P. P. Patil, P. M. Humnabade, *Diabetes diagnosis using fuzzy min-max neural network with rule extraction and apriori algorithm*, The International Journal of Science and Technoledge, **2**(4) (2014), 369.

[18] P. K. Simpson, *Fuzzy min-max neural networks-Part 1: Classication*, IEEE Transactions on Neural Networks, **3**(5) (1992), 776-786.

[19] P. K. Simpson, *Fuzzy min-max neural networks-Part 2: Clustering*, IEEE Transactions on Fuzzy Systems, **1**(1) (1993), 32-45.

[20] A. Singh, J. C. Mehta, D. Anand, P. Nath, B. Pandey, A. Khamparia, *An intelligent hybrid approach for hepatitis disease diagnosis: Combining enhanced kmeans clustering and improved ensemble learning*, Expert Systems, **38**(1) (2021), e12526.

[21] A. Singh, B. Pandey, *Intelligent techniques and applications in liver disorders: A survey*, International Journal of Biomedical Engineering and Technology, **16**(1) (2014), 27-70.

[22] R. K. Sterling, E. Lissen, N. Clumeck, R. Sola, M. C. Correa, J. Montaner, D. Messinger, *Development of a simple noninvasive index to predict significant fibrosis in patients with HIV/HCV coinfection*, Hepatology, **43**(6) (2006), 1317-1325.

[23] T. N. Tran, D. M. Vu, M. T. Tran, B. D. Le, *The combination of fuzzy min-max neural network and semi-supervised learning in solving liver disease diagnosis support problem*, Arabian Journal for Science and Engineering, **44**(4) (2019), 2933-2944.

[24] D. M. Vu, V. H. Nguyen, B. D. Le, *Semi-supervised clustering in fuzzy min-max neural network*, In International Conference on Advances in Information and Communication Technology. Springer International Publishing, (2016), 541-550.

[25] C. T. Wai, J. K. Greenson, R. J. Fontana, J. D. Kalbfleisch, J. A. Marrero, H. S. Conjeevaram, A. S. F. Lok, *A simple noninvasive index can predict both significant fibrosis and cirrhosis in patients with chronic hepatitis C*, Hepatology, **38**(2) (2003), 518-526 .

[26] J. Wang, C. P. Lim, D. Creighton, A. Khorsavi, S. Nahavandi, et. al., *Patient admission prediction using a pruned fuzzy min-max neural network with rule extraction*, Neural Computing and Applications, **26**(2) (2015), 277-289.

[27] Z. Yao, J. Li, Z. Guan, Y. Ye, Y. Chen, *Liver disease screening based on densely connected deep neural networks*, Neural Networks, **123** (2020), 299-304.

[28] C. Zhong, M. Malinen, M. Miao, P. Fränti, *A fast minimum spanning tree algorithm based on K-means*, Information Sciences, **295** (2015), 1-17.