

Takagi-Sugeno-Kang(zero-order) model for diagnosis hepatitis disease

Dr. Raidah Salim

Computer Science Department, Science College, Basra University

Abstract

The aim of this paper is to use **Takagi-Sugeno-Kang(zero-order)** model as fuzzy neural network for the medical diagnosis of hepatitis diseases which represent a major public health problem all around the world . For further improve the accuracy and the speed of the diagnosis, the **Microarray Attribute Reduction Scheme (MARS)** for reduction features (or attributes) and **Mean Imputation (MI)** method for treatment the missing values were used in this work. The used data source of hepatitis diseases was taken from **UCI machine learning repository**.

After treat the missing values problem by apply MI method, the dataset is partitioned into three training-testing partitions (**30%–70%**, **40–60%** and **20%–80%** respectively) and apply MARS with different values of **thr**(from 0.1-0.9) in order to determine the number attributes (that represent the number of inputs to the fuzzy neural network), the results record in each case of **thr** values and each case of partitions. The high diagnosis accuracy has been achieved for the **40–60%** training-testing, namely, **100%** for training and **95.77%** for testing with **thr** equal to 0.4 and with less training cycle and fuzzy sets number. This work was implemented in MATLAB 7.0 environment.

Keywords: medical diagnosis, Fuzzy logic, fuzzy neural network, *Microarray*

Attribute Reduction Scheme, Mean Imputation.

1. Introduction

Medical Diagnosis can be stated as the process of determining or identifying a possible disease or a disorder. A clinician uses several sources of data and classifies this data in order to find the disorder. A medical diagnosis is made by a physician based on assessment of symptoms and diagnostic tests [1].

Nowadays, the use of computer technology in the field of medicine has highly increased [2]. The use of intelligent systems such as neural network, fuzzy logic, genetic algorithm and fuzzy neural systems has highly helped in complex and uncertain medical tasks such as diagnosis of diseases [3]. Over the last few decades, neural networks and fuzzy systems have established their reputation as alternative approaches to intelligent information processing systems. Both have certain advantages over classical methods, especially when vague data or prior knowledge is involved. However, their applicability suffered from several weaknesses of the individual models. Therefore, combinations of neural networks with fuzzy systems have been proposed, where both models complement each other.

Fuzzy neural hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks [4]. The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a fuzzy system to represent knowledge in an interpretable

manner and the learning ability of a neural network to optimize its parameters [5].

In Section 2 the method that is used for treatments missing values was presented. The schema for attribute reduction was presented in section 3. The structure and learning parameters of the fuzzy neural network was presented in section 4 and 5. Section 6 and 7 describes the dataset used and describe this work. The experimental results are described in Section 8. The conclusion and future work of the work is given in section 9

2. Mean Imputation

Mean imputation (MI) is one of the most frequently used methods for treat the missing values. It consists of replacing the missing data for a given feature (attribute) by the mean of all known values of that attribute in the class where the instance with missing attribute belongs. The value x_{ij} of the k -th class, C_k , is missing then it will be replaced by:

$$\hat{x}_{ij} = \sum_{i: x_{ij} \in C_k} \frac{x_{ij}}{n_k}, \quad \dots\dots\dots 1$$

where n_k represents the number of non-missing values in the j -th feature of the k -th class. According to Little and Rubin [6] the drawbacks of mean imputation are:

- 1) Sample size is overestimated.
- 2) Variance is underestimated.
- 3) Correlation is negatively biased.
- 4) The distribution of new values is an incorrect representation of the population values because the shape of the distribution is distorted by adding values equal to the mean.

Replacing all missing records with a single value will deflate the variance and artificially inflate the significance of any statistical tests based on it. Surprisingly though, mean imputation has given good experimental results in data sets used for supervised classification purposes [7].

3. Microarray Attribute Reduction Scheme

The Microarray Attribute Reduction Scheme (MARS) [8] is a schema for attribute reduction works as follows:

1. Given a training dataset S with n samples, and m is the number of features. In S , all samples associated with a class label $k \in \{1, 2, \dots, c\}$ where c is the number of total classes.

$$S = \{X^j | X^j = (X_1^j, X_2^j, \dots, X_m^j), 1 \leq j \leq n\} \quad \dots\dots\dots 2$$

2. For each feature x_i , the μ_{ik} that is the mean of i -th features for class k is calculated as:

$$\mu_i^k = \left(\sum_{\forall X^j \in S_k} x_i^j \right) \times |S_k|^{-1} \quad \dots\dots\dots 3$$

Where $|S_k|$ represents the number of samples in the k -th class.

3. The value μ_i that is the mean of i -th attributes of all samples is denoted as:

$$\mu_i = \frac{\left(\sum_{j=1}^n x_i^j \right)}{n} \quad \dots\dots\dots 4$$

4. The score(i, k), indicating the ability of the i -th feature to identify the samples associated with the k class label can then be obtained by:

$$score(i, k) = \left| \sum_{\forall X^j \in S_k} v \left(x_i^j, k \right) \right| \times |S_k|^{-1} \quad \dots\dots\dots 5$$

where $v(x_{ij}, k)$ denoted in the following is a weighted voting scheme:

$$V(X_i^j, k) = \begin{cases} 1 & \text{if } |X_i^j - \mu_i^k| \leq |X_i^j - \mu_i^l|, \forall l \in \{1, 2, \dots, c\}, l \neq k, \dots \dots \dots 6 \\ 0 & \text{otherwise} \end{cases}$$

By measuring the distance between class k and class $l \in \{1, 2, \dots, c\}$ under the i -th feature, the $score(i, k)$ metric reflects the fact that the higher a score value, the more ability the corresponding i -th feature can distinguish the k class samples. In those features whose scores exceed a threshold value thr are characterized as the most discriminatory features and recorded in the vector $v_g = (v_1, v_2, \dots, v_i, \dots, v_m)^T$, where the attribute v_i is defined as:

$$v_i = \max_{k \in \{1, 2, \dots, c\}} (select(i, k, thr)) \dots \dots \dots 7$$

Where

$$select(i, k, thr) = \begin{cases} 1 & \text{if } score(i, k) \geq thr \dots \dots \dots 8 \\ 0 & \text{otherwise} \end{cases}$$

Each v_i represents whether the i -th attribute is a selected attribute or not.

4. Structure of Fuzzy neural network

The biggest advantage of neural network methods is that they are general and the disadvantage of neural networks is that they are notoriously slow, and it is very difficult to determine how the net is making its decision.

In the field of artificial intelligence, fuzzy neural network refers to combinations of artificial neural networks and fuzzy logic. Fuzzy neural network hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. The main strength fuzzy

neural system is that they are universal approximates with the ability to solicit interpretable IF-THEN rules[9].

Figure (1) shown the structure of Takagi-Sugeno-Kang fuzzy neural network which consist of four layers describe as follow[10,11]:

Layer 1: Each node in this layer, only transmits input values to the next layer directly.

Layer 2: Each node in this layer corresponds to one fuzzy set (linguistic label)

of one of the input variables in Layer 1. The operation in this layer called fuzzification where each crisp input value convert to fuzzy value by using

membership function (in this work use Gaussian function) as bellow:

$$\mu_i = \frac{1}{\sigma_j} \exp\left(-\frac{(x_i - c_j)^2}{2\sigma_j^2}\right) \quad \begin{matrix} i = 1, 2, \dots, fsetno \text{ and} \\ i = 1, 2, \dots, N \end{matrix} \dots \dots \dots 9$$

Where:

c is the center of membership for input i with it fuzzy set j .

σ is the variance of membership for input i with it fuzzy set j .

N is the number of inputs.

$fsetno$ is the number of fuzzy sets for each input.

μ_i is membership the input x_i

Layer 3: Nodes in this layer are rule nodes, and constitute the antecedents of

the fuzzy rule base. Every node in this layer is a fixed node labeled as Π ,

whose output is the product of all incoming signals as follow:

$$a_j = \prod_{i=1}^N \mu_i \quad i=1, 2, \dots, N \text{ and } i=1, 2, \dots, ruleno \dots \dots \dots 10$$

Layer 4: This act as the defuzzifier and normalize. This layer are called

“Normalized and summation of firing strengths”. To get output of fuzzy neural

network, every node in this layer is represent the rule's firing strength to the sum of all rules' firing strengths and normalize the firing strength as:

$$y = \frac{\sum_{i=1}^{ruleno} w_i u_i}{\sum_{i=1}^{ruleno} u_i} \quad \dots\dots\dots 11$$

Where w_i strength of rule i.

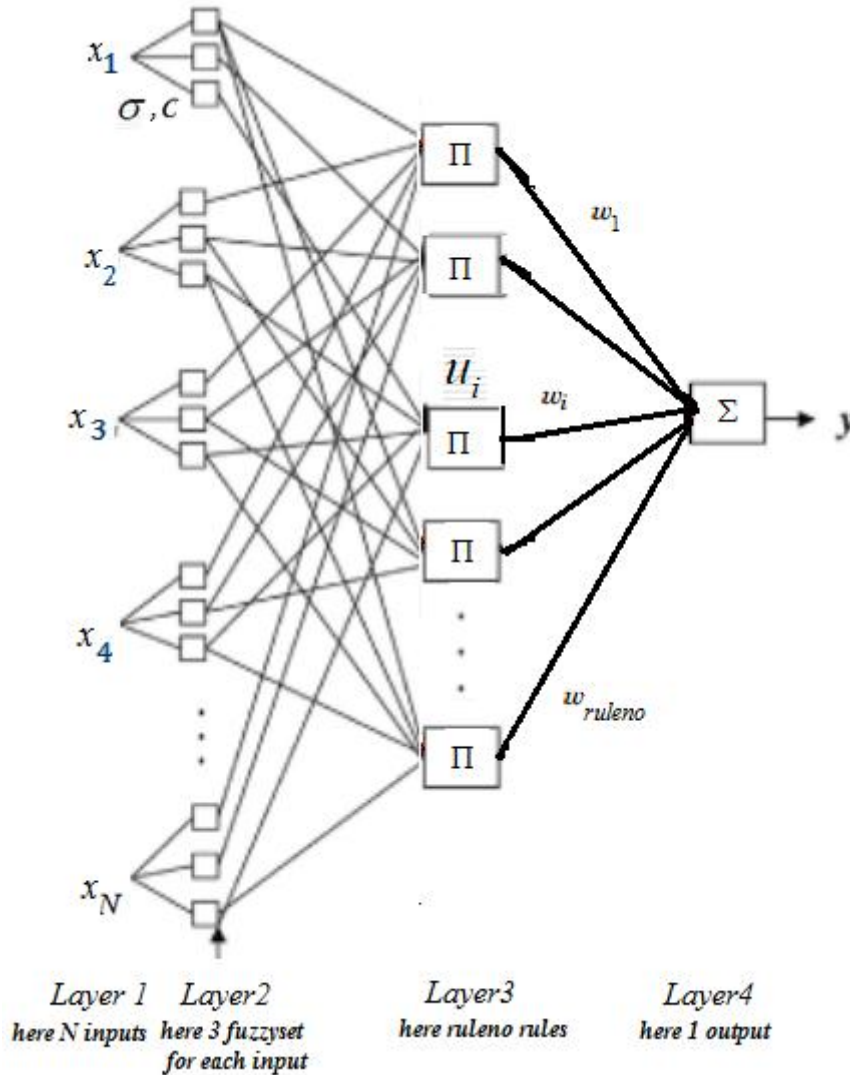


Figure (1) Structure of Takagi-Sugeno-Kang fuzzy neural network.

5. Parameters learning of fuzzy neural network

Based on the above structure, the learning algorithm was used to determine the proper centers and variance of each fuzzy sets in the system. In this paper the backpropagation algorithm to tune the parameters of the fuzzy neural network.

- The weights (w) in layer 4 are updated by [10,11]

$$w_i(t+1) = w_i(t) + \eta \times [y(t) - y_a(t)] \times \frac{u_i}{\sum_{i=1}^{ruleno} u_i} \quad \dots\dots\dots 12$$

- The centers and variances of the fuzzy sets in layer 2 are updated by[10,11]

$$c_y(t+1) = c_y(t) + \eta \times [y(t) - y_d(t)] \times [w_i - y(t)] \times [u_i / \sum u_i] \times \frac{(x_i - c_y(t))^2}{(\sigma_y(t))^2} \quad \text{----- 13}$$

$$\sigma_y(t+1) = \sigma_y(t) + \eta \times [y(t) - y_d(t)] \times [w_i - y(t)] \times [u_i / \sum u_i] \times \frac{(x_i - c_y(t))^2}{(\sigma_y(t))^3} \quad \text{----- 14}$$

6. Dataset description

The hepatitis database taken from UCI machine learning repository[10] The dataset contains 155 samples. Each sample in the dataset has 20 attribute

including the class attribute DIE and LIVE (as output) distribute as DIE: 32 and LIVE: 123. The output shows whether patients with hepatitis are alive or dead.

Table 1 shows the details of the hepatitis data. There are number of missing values in each attribute. Table 2 shows the description of these missing attribute values: (indicated by "?").

Table 1: Description of attributes from UCI machine learning dataset

Attribute number	Attribute Name	Attribute value
1	Class	DIE, LIVE (output attribute)
2	AGE	10, 20, 30, 40, 50, 60, 70, 80
3	SEX	male, female
4	STEROID	no, yes
5	ANTIVIRALS	no, yes
6	FATIGUE	no, yes
7	MALAISE	no, yes
8	ANOREXIA	no, yes
9	. LIVER BIG	no, yes
10	LIVER FIRM	no, yes
11	. SPLEEN PALPABLE	no, yes
12	. SPIDERS	no, yes
13	ASCITES	no, yes
14	VARICES	no, yes
15	BILIRUBIN	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
16	ALK PHOSPHATE	33, 80, 120, 160, 200, 250
17	SGOT	13, 100, 200, 300, 400, 500
18	ALBUMIN	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
19	PROTIME	10, 20, 30, 40, 50, 60, 70, 80, 90
20	HISTOLOGY	no, yes

Table 2: Description of missing attributes values

Attribute Number	Number of Missing Values	Attribute Number	Number of Missing Values
1	0	11	5
2	0	12	5
3	0	13	5
4	1	14	5
5	0	15	6
6	1	16	29
7	1	17	4
8	1	18	16
9	10	19	67
10	11	20	0

7. Describe the Work

The diagram in figure (2) and the following points describe the work of this paper:

- Apply the MI method for treatment the missing values in the original dataset, where in each value "?" in each attribute replace by **round** the mean of all known values of that attribute in the class.
- By apply the MARS with different values for **thr** from 0.1-0.9. The number of attributes was reduced according to value of **thr**. Table 3 shown the number of attributes after reduce and the select attributes number for **thr** from 0.1-0.9
- In order test the ability of the Takagi-Sugeno-Kang(zero-order) system that is used in this work with MI and MARS in the training and in the generalization, The dataset is

partitioned into three training-testing partitions (30%–70%, 40–60% and 20%–80% respectively). Table 4 shown the number of patterns (or samples) for training and for testing in each training-testing partitions.

- For the hepatitis problem the attributes represent the number of Sugeno fuzzy model inputs and the number of outputs is one that represents class Die or Live (first attribute in table 1).
- Gaussian function used as a membership function in fuzzification layer and constant function in defuzzification layer.

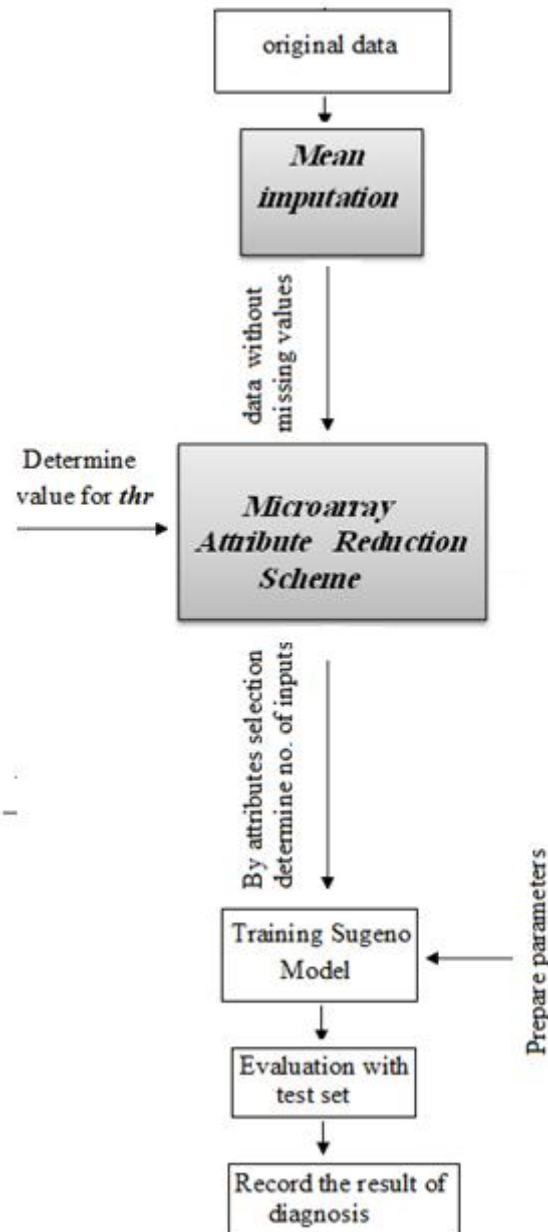


Figure (2) Diagram show this work

Table 3 Summarize of apply MARS in different values of *thr*

<i>thr</i> value	Number of attributes	Selected attributes
0.1	19	All attributes(1..19)
0.2	16	1,3-6,9-19
0.3	16	1,3-6,9-19
0.4	15	1,3,5,6,9-19
0.5	13	1,3,6,9-15,17-19
0.6	13	1,3,6,9-15,17-19
0.7	11	1,6,10-15,17-19
0.8	7	10-15,17
0.9	4	10,12-14

- Prepare parameters of Takagi-Sugeno-Kang fuzzy model:
1. To find the best number of fuzzy sets we experiment the vales (6-11) as a fuzzy sets for each input.
 2. The best learning rate is 0.01 after many experiments.
 3. The maximum learning cycles is 1000 cycles.
 4. To start train the fuzzy model, the initial weights are chosen as follow:
 - The following steps shown how the weights of membership function parameters (c is center and b (σ) is variance of membership function) are initialization:

```

value = 0.3;
step = 1/ruleno;
for i=1:fsetno
    for j=1:ruleno
        c(i,j) = value;
        b(i,j) = 1;
        value = value + step;
    end
end

```

- The following steps shown how the weights that represent firing strength of rules are initialization:

```

k=1;
for j=1:ruleno

```

```

    if (j>patterno)    w(j) = 0.1;
    else               w(j) = desiredout(j);
end

```

where:

- w is weights matrix for consequent part.
- rulehno represent number of rule in layer 3.
- Desireout(y_a) represent the desired output in training-test partition that the number of its patterns for training is patternno (N).
- fsetno represent the number of fuzzy sets for each input

- After prepare the training patterns and testing by apply MI algorithm and MARS algorithm on the original patterns or samples, run the fuzzy neural model many times, in each times input different number of fuzzy sets 6,7,8,9,10,11. the training and testing rate, the error in training stage and testing stage , number of fuzzy sets and number of training cycles for two best run in each training-testing partition. Tables 5-9 shown accuracy for three different training and testing sets in different values of *thr*.

Table (4) Training-Testing partitions

Training – Testing partition(%)	No. of patterns for training	No. of patterns for testing
30-70	31	124
40-60	62	93
20-80	47	108

Table (5) Accuracy for three different training and testing sets in case $thr = 0.1$

Partition set	Best runs	Training Rate	Training Error	Test Rate	Test Error	No. of Cycle	No. of fuzzy sets
(30%-70%)	1	100%	0.008	88.89%	0.008	786	6
	2	100%	0.001	87.04%	0.06	747	8
(40%-60%)	1	100%	0.01	92.5%	0.05	490	8
	2	100%	0.01	91.4%	0.05	587	7
(20%-80%)	1	100%	0.02	87.9%	0.06	444	7
	2	100%	0.009	86%	0.02	541	6

Table (6) Accuracy for three different training and testing sets in case $thr = 0.2$ and $thr=0.3$

Partition set	Best run	Training Rate	Training Error	Test Rate	Test Error	No. of Cycle	No. of fuzzy sets
(30%-70%)	1	100%	0.01	88.89%	0.05	778	10
	2	100%	0.009	87.96%	0.06	580	8
(40%-60%)	1	100%	0.002	94.6%	0.03	502	8
	2	100%	0.01	91.4%	0.05	727	10
(20%-80%)	1	100%	0.02	88.7%	0.06	568	9
	2	100%	0.009	87.9%	0.05	506	8

Table (7) Accuracy for three different training and testing sets in case **thr = 0.4**

Partition set	Best run	Training Rate	Training Error	Test Rate	Test Error	No. of Cycle	No. of fuzzy sets
(30%-70%)	1	100%	0.007	88.89%	0.04	863	6
	2	100%	0.002	87.03%	0.06	888	10
(40%-60%)	1	100%	0.003	95.7%	0.02	500	8
	2	100%	0.01	90.3%	0.06	530	9
(20%-80%)	1	100%	0.007	88.7%	0.05	756	9
	2	100%	0.009	87.92%	0.05	501	8

Table (8) Accuracy for three different training and testing sets in case **thr = 0.5 and thr=0.6**

Partition set	Best run	Training Rate	Training Error	Test Rate	Test Error	No. of Cycle	No. of fuzzy sets
(30%-70%)	1	100%	0.002	89.8%	0.01	516	7
	2	100%	0.005	87.96%	0.02	493	6
(40%-60%)	1	100%	0.002	92.5%	0.01	920	10
	2	100%	0.007	88%	0.02	738	11
(20%-80%)	1	100%	0.009	88%	0.01	418	8
	2	100%	0.001	87%	0.03	531	6

Table (9) Accuracy for three different training and testing sets in case $thr = 0.7$

Partition set	Best run	Training Rate	Training Error	Test Rate	Test Error	No. of Cycle	No. of fuzzy sets
(30%-70%)	1	100%	0.01	87.03%	0.05	640	8
	2	100%	0.008	86.1%	0.06	666	9
(40%-60%)	1	100%	0.009	91.4%	0.02	847	9
	2	100%	0.01	87%	0.06	888	8
(20%-80%)	1	100%	0.006	89.5%	0.05	906	8
	2	100%	0.02	88.7%	0.04	531	10

8. Experimental results

This section evaluates the performance of this work on diagnosis of the hepatitis disease. The data sets representing this problem were obtained from the UCI machine learning benchmark repository and were real world data. Table 1 and Table 2 shows the descriptions of the data sets. In this work as shown in Table 3 the number of patterns for training less than the number of patterns for testing.

The diagnosis accuracy on the testing data for the reduced feature subset is shown in tables (5-9) in different values of thr with three partition sets. In general the high diagnosis accuracy has been achieved for the 40–60% training–testing partition in each values of the thr and the highest accuracy, namely, 95.77% has been achieved with thr equal to 0.4 with less training cycle and fuzzy sets number.

9. Conclusion and Future work

- The hepatitis treatment is very expensive and severe side effects, therefore, it is important to identify those patients who most probably can react to the treatment, so that the others

can be protected from a treatment with no benefits.

- This work present an efficient way for medical diagnosis by using two algorithms (Reduction algorithm and treat the missing values) as preprocessing the data before training the fuzzy neural network with these data. Where reducing algorithm are suitable for hepatitis diagnosis which will help in reducing extra time consumption in treatment.
- In order to keep the whole real world data as it, we must treat any missing values in this work use MI as algorithm for this purpose.
- The number of fuzzy sets of fuzzy neural network which select by trail and error plays an important role for give higher diagnosis accuracy as shown in tables 5-9. For a particular input, any feature(s) for hepatitis problem not be effective to the number of fuzzy sets. By extracting these features by using MARS we can minimize the training time. As a future work, we will try to extend the algorithm for improving backpropagation using another feature

selection algorithm, another algorithm for treatments missing values and use genetic algorithm to predict a suitable number of fuzzy sets..

- This paper focusing at the using algorithm for reduces the attributes and treatments the missing values and ignored all the factors that improve the performance of backpropagation algorithm.
- From the tables 5-9 the above existing methods are effective in diagnosis; their accuracy may be reduced when handling the datasets with a large amount of attributes and very few attributes.

Reference

1. Peter K. Sharpe..et.al, "**Artificial Neural Networks in Diagnosis of Thyroid Function**", clinical chemistry, 1993.
2. Ali A. and Mehdi N., "**A Fuzzy Expert System for Heart Disease Diagnosis**". A Proceeding of the International MultiConference of Engineers and Computer Scientists, vol. 1, 2010.
3. ZAPTRON Systems, Inc. Neurofuzzy, "**A Different Type of Neural Nets**" Zaptron's High-Order Nonlinear Neural Networks.Fuzzy logic, 1999.
4. Mehdi N. and Mehdi Y. , "**Designing a Fuzzy Expert System of Diagnosing the Hepatitis B Intensity Rate and Comparing it with Adaptive Neural Network Fuzzy System**" A Proceeding of the World Congress on Engineering and Computer Science, vol. II, 2009.
5. Vipul A.S., "**Adaptive Neuro-Fuzzy Inference System for Effect of Wall Capacitance in a Batch Reactor**" Advances in Fuzzy Mathematics ISSN 0973-533X 4;69-75, 69-70 , 2009.
6. Little, R. J. and Rubin, D.B. "**Statistical Analysis with Missing Data**", Second Edition. John Wiley and Sons, New York, 2002.
7. Mundfrom, D.J and Whitcomb, A. , "**Imputing missing values: The effect on the accuracy of classification**". Multiple Linear Regression Viewpoints. 25(1), 13-19, 1998..
8. Jain-Shing Wu, Chuan-Wen Chiang, Chung-Nan Lee and Kuo-Yi Wu1, "**MARS: A Microarray Attribute Reduction Scheme for Gene Expression Classification Problems**"
9. The Wikipedia website [online]. Available at: <http://www.wikipedia.org/>
10. Ajoy Kumar Palit, and Gerhard Doeding, "**Backpropagation Based Training Algorithm for Takagi - Sugeno Type MIMO Neuro-Fuzzy Network to Forecast Electrical Load Time Series** " . IEEE, 2002.
11. Stefka Stoeva, Alexander Nikov, "**A fuzzy backpropagation algorithm** ", Fuzzy Sets and Systems Vol. 112 P. 27-39, 2002.
10. <http://www.ics.uci.edu/pub/ml-repos/machine-learning-databases/>, 2003.