Mathematical Structure of Fuzzy Modeling of Medical Diagnoses by Using Clustering Models

Rana Waleed Hndoosh, M. S. Saroa, Sanjeev Kumar

Abstract— An *Adaptive-Network*-based *Fuzzy Inference System ANFIS* with different techniques of *clustering* is successfully developed to solve one of the problems of medical diagnoses, because it has the advantage of powerful modeling ability. In this paper, we propose the generation of an adaptive neuro-Fuzzy Inference System model using different clustering models such as a subtractive fuzzy clustering (*SFC*) model and a fuzzy c-mean clustering (*FCM*) model in the Takagi-Sugeno (*TS*) fuzzy model for selecting the hidden node centers. An experimental result on datasets of medical diagnoses shows the proposed model with two models of clustering (*ANFIS-SFC* & *ANFIS-FCM*) while comparing the same model but both with and without clustering models (*ANFIS*). We obtained better results of *average Training error* of training and checking data with *ANFIS-SFC* when we used a *Back-propagation* model of the Learning Rule, and similarly we obtained the best results with *ANFIS-FCM* when used with a *Hybrid* model. Also we have applied *SFC* & *FCM* models without *ANFIS* to get different matrices of cluster centers on medical diagnoses. Finally, we have displayed the surface of *MF* to each of the ten separate clusters of diseases with values of the objective function.

Index Terms— Clustering, Fuzzy-Cluster System, *FCM*, *SFC*, *ANFIS*, *TS* Fuzzy Model, Learning Rule.

1. INTRODUCTION

Fuzzy models describe systems by establishing relations between the relevant variables in the form of *if–then* rules. Traditionally, a fuzzy model is built by using expert knowledge in the form of linguistic rules. Different approaches have been proposed for this purpose, like fuzzy modeling [1], [2], [10], [17] neural-network training techniques [4], [14], [18], [20] and product-space clustering [3], [6], [12]. Solutions to this problem have been sought for fuzzy neural networks [2], [11], [13], [17], and for fuzzy rule-based models in general [10], [16], [18], [20]. In section II, we present a general model of a fuzzy-cluster system that is explained by some steps. Section 3 shows how a mathematical structure of the applied Takagi–Sugeno (*TS*) rule-based model depends on clustering [3], [9], [16], [19]. In Section 4, we describe a fuzzy clustering of *TS* models that consist of two models, a subtractive fuzzy clustering *SFC* model and a fuzzy c-mean clustering (*FCM*) model that are applied to a modeling problem [6], [7], [9], [10], [17], [18]. After explaining steps of *SFC* and *FCM* models, in Section 5, we have built the structure of *ANFIS* with the *TS* model that depends on *SFC* and *FCM* and we use a *Learning Rule* as the "*back-propagation learning rule*" [8], [5], [11], [13], [18], [20]. In Section 6, we apply all these models on real data of medical diagnosis and we perform a system at *50 epochs* to get the average testing error of training data and the average testing error of checking of *ANFIS-SFC*, *ANFIS-FCM* and *ANFIS* without clustering in part of the experiments and discussion [9], [10], [15]. The performance of the models is measured by the square root of the mean squares prediction error (*RMSE*), where we obtained very small errors, and the smallest value of Average *RMSE* with *ANFIS-FCM*. So we have built a system that has proved its efficiency in the medical diagnoses. Finally, some concluding remarks are given in Section 7.

I. 2. GENERAL FORMULATION of a FUZZY-CLUSTER SYSTEM

Fuzzy clustering is used to detect multidimensional fuzzy sets in the product space of the input variables to identify the antecedent of the fuzzy rules and then assign a linear consequent to each rule. The fuzzy rule base comprises the following fuzzy IF-THEN rules [10], [17]:

$$R^{(l)}: IF\ x_1\ is\ A_1^l\ and \ldots and\ x_n\ is\ A_n^l\ \ THEN\ \hat{y}\ is\ B^l, \qquad (1)$$

where $A_1^l\ and\ B^l$ are fuzzy sets in $U_i \subset R, V \subset R$, respectively, and $\mathrm{x} = (x_1, x_2, \ldots, x_n)^T \in U\ and\ y \in V$ are the input and output (linguistic) variables of the fuzzy system, respectively [4], [7], [11].

Suppose that we are given C input-output pairs (x_0^1, y_0^1), $l = 1,2, \ldots, C$. Our task is to construct a fuzzy system $\hat{y}$ that can match all the C pairs to any given accuracy; that is, for any given $\varepsilon > 0$, we require that $|\hat{y}_0^l - y_0^l| < \varepsilon$, $\forall\ l = 1,2, \ldots, C$.

This optimal fuzzy system may be constructed as [18]:

$$\hat{y} = \frac{\sum_{l=1}^{N} y_0^l . \mu_{A^l}(x)}{\sum_{l=1}^{N} \mu_{A^l}(x)} \qquad (2)$$

Clearly, the fuzzy system (2) is constructed from the C rules in the form of (0) with $\exp\left(-\frac{|x_i - x_{0i}^l|^2}{\delta^2}\right)$ and the center of B^l is equal to y_0^l, and uses the product inference system, singleton fuzzifier, and center average defuzzifier. The optimal fuzzy

- *Rana Waleed Hndoosh is currently pursuing Ph.D degree program in M. M. U, India, and working in department of Software Engineering, College of Computers Sciences & Mathematics, Mosul University, Iraq. rm.mr2100@yahoo.com.*
- *M. S. Saroa is currently professor in department of Mathematics, Maharishi Markandeshawar University, Mullana-133207, India. mssaroa@yahoo.co.in.*
- *Sanjeev Kumar is currently professor in department of Mathematics, Dr. B. R. Ambedkar University, Khandari Campus, Agra-282002, India. sanjeevibs@yahoo.co.in.*

system (2) uses one rule for one input-output pair. Thus it is no longer a practical system if the number of input-output pairs is large [12]. For these large sample problems, various clustering techniques can be used to group the input-output pairs so that a group can be represented by one rule. In this work, we have used [2]:

Step1: Starting with the first input-output pair (x_0^1, y_0^1), establish a cluster center $x_c^1\ at\ x_0^1$, and set $A^1(1) = y_0^1, B^1(1) = 1$. Select a radius *r*.

Step2: Suppose that when we consider the k^{th} input-output pair $(x_0^k, y_0^k), k = 2,3, \ldots$, there are C clusters with centers at $x_c^1, x_c^2, \ldots, x_c^C$. Compute the distances of x_0^k to these C cluster centers, $|x_0^k - x_c^l|, l = 1, 2, \ldots, C$, and let the smallest distances be $\left|x_0^k - x_c^{l_k}\right|$, that is, the nearest cluster to $x_0^k\ is\ \ x_c^{l_k}$, then:

a) If $\left|x_0^k - x_c^{l_k}\right| > r$, establish x_0^k as a new cluster center $x_c^{C+1} = x_0^k$, set $A^{C+1}(k) = y_0^k, B^{C+1}(k) = 1$, and keep $A^l(k) = A^l(k-1)$, $B^l(k) = B^l(k-1)$, for $l = 1, 2, \ldots, C$.

b) If $\left|x_0^k - x_c^{l_k}\right| \leq r$, do the following:

$$A^{l_k}(k) = A^{l_k}(k-1) + y_0^k, \tag{3}$$

$$B^{l_k}(k) = B^{l_k}(k-1) + 1, \text{ and set} \tag{4}$$

$$A^l(k) = A^l(k-1), \tag{5}$$

$$B^l(k) = B^l(k-1), \text{for } l = 1, 2, \ldots, C \text{ with } l \neq l_k\,, \tag{6}$$

Step3: 1f x_0^k does not establish a new cluster then the designed fuzzy system based on the k input-output pairs $\left(x_0^j, y_0^j\right), j = 1, 2, \ldots, k$, is:

$$\hat{y} = \frac{\sum_{l=1}^{C} A^l(k).e^{\left(-\frac{\left|x - x_c^l\right|^2}{\delta}\right)}}{\sum_{l=1}^{C} B^l(k).e^{\left(-\frac{\left|x - x_c^l\right|^2}{\delta}\right)}} \tag{7}$$

1f x_0^k establishes a new cluster, then the designed fuzzy system is [3], [7]:

$$\hat{y} = \frac{\sum_{l=1}^{C+1} A^l(k).e^{\left(-\frac{\left|x - x_c^l\right|^2}{\delta}\right)}}{\sum_{l=1}^{C+1} B^l(k).e^{\left(-\frac{\left|x - x_c^l\right|^2}{\delta}\right)}}, \tag{8}$$

Step4: Repeat by going to (Step2) with $k = k + 1$.

From (3) to (6) we see that the variable $B^l(k)$ equals the number of input-output pairs in the l^{th} cluster after k input-output pairs have been used, and $A^l(k)$ equals the summation of the output values of the input-output pairs in the l^{th} cluster. Therefore, if each input-output pair establishes a cluster center, then the designed fuzzy system (8) becomes the optimal fuzzy system (2), because the optimal fuzzy system (2) can be viewed as using one rule to match one input-output pair and the fuzzy system (7) or (8) can be viewed as using one rule to match one cluster of input-output pairs. The number of clusters (or rules) depends on the distribution of the input points in the input-output pairs and the radius *r*. The radius *r* determines the complexity of the designed fuzzy system. For smaller *r*, we have more clusters. For larger *r*, the designed fuzzy system is simpler but less powerful. In practice, a good radius *r* may be obtained by test and errors [4], [10], [18], [20].

3. MATHEMATICAL STRUCTURE of the APPLIED TS MODELS

The unknown nonlinear system of the *TS* fuzzy model is formulated as: $\hat{y}_k = f(\mathrm{x}_k)$, that is based on specified or measured input data $\mathrm{x_k = [x_{1k}, x_{2k}, \ldots, x_{ck}]^T}$ and measured output data $\mathrm{y_k}$ of the system, where $\mathrm{k = 1, 2, \ldots, N}$ denotes the index of the $\mathrm{k^{th}}$ input-output data-pair. In general it may not be easy to find a global nonlinear model that is universally applicable to describe the unknown system $\mathrm{f(.)}$. In that case it would certainly be to build local linear models for specific operating points of the process and combine these into a global model. This can be done by combining a number of local models, where each local model has a predefined operating region. This results in the so-called operating of the *TS* fuzzy model when the operating region is defined by fuzzy rules. This type of operating is formulated as [3], [9], [14]:

$$\hat{y}_k = \sum_{l=1}^{C} w_l(x_k)(a_l^T \phi_k + b_l), \tag{9}$$

where $\mathrm{w_i(x_k)}$ describes the operating of the l^{th} local linear model defined by the parameter vector $\theta_l = [a_l^T b_l]^T$, and $\mathrm{x_k}, \mathrm{\varphi_k}$ subsets of the original input vector $\mathrm{x_k}$. From the general form (0) with the form of the *TS* fuzzy model we derive the model as:

$$R^{(l)}: IF\ x_{1k}\ is\ A_1^l(x_{1k}) and \ldots and\ x_{nk}\ is\ A_n^l(x_{1k})$$

$$THEN\ \hat{y}_k = a_l^T \phi_k + b_l\ \ [\omega_l] \tag{10}$$

$A_j^l(x_{jk})$ represents a multivariate membership function that describes the fuzzy set A_j^l, Where $a_l\ and\ b_l$ are the parameters of the local linear model, and $\omega_l = [0,1]$ is the weight of the rule that represents the desired impact of the rule, (It is often chosen by the designer of the fuzzy system). The degree of performance of a rule is calculated as the product of the degree of performance of the fuzzy sets in the rule that is denoted by [11], [13], [19]:

$$\beta_l(x_k) = \omega_l A^l(x_k) = \omega_l \prod_{j=1}^{n} A_j^l(x_{jk}) \tag{11}$$

The rules of the fuzzy model are aggregated using the normalized fuzzy mean formula:

$$\hat{y}_k = \frac{\sum_{l=1}^{C} \omega_l \beta_l(x_k)(a_l^T \phi_k + b_l)}{\sum_{l=1}^{C} \omega_l \beta_l(x_k)} \tag{12}$$

Gaussian membership functions are used here to represent the fuzzy set $A_j^l(x_{jk})$:

$$A_j^l(x_{jk}) = \exp\left(-\frac{1}{2}\frac{(x_{jk} - c_{lj})^2}{\delta_{lj}^2}\right), \tag{13}$$

where c_{lj} epresents the center and δ_{lj}^2 the variance of the Gaussian curve. The use of Gaussian membership function allows for the compact formulation of (11), [4]:

$$\beta_l(x_k) = \omega_l A^l(x_k) = \omega_l . \exp\left(-\frac{1}{2}\left(x_k - \varsigma_j^x\right)^T (F_l)^{-1}\left(x_k - \varsigma_j^x\right)\right),$$
(14)

where $\varsigma_{\mathrm{j}}^{\mathrm{x}} = \left[\mathrm{c}_{1\mathrm{j}}, \mathrm{c}_{2\mathrm{j}}, \dots, \mathrm{c}_{\mathrm{Cj}}\right]$ denotes the center of the l^{th} multivariate Gaussian and F_l is the inverse of the matrix containing the variances on its diagonal:

$$F_l = \begin{pmatrix} \delta_{l1}^2 & 0 & \dots & 0 \\ 0 & \delta_{l2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \delta_{lC}^2 \end{pmatrix} \tag{15}$$

In the following sections a special type of clustering model is presented.

4. FUZZY CLUSTERING of TS MODELS

In this section, we will review the concepts of two different models, the first is subtractive fuzzy clustering model and the second is Fuzzy c-means model.

4.1 Subtractive Fuzzy Clustering (SFC) Model

Subtractive fuzzy clustering divides the X data into c fuzzy clusters. This means, each observation consists of the input and the output variables, grouped into a row vector $\mathrm{x}_k = [x_k, y_k]$, where k denotes the $\mathrm{k^{th}}$ row of the X matrix. To calculate the parameters of the clusters, the cluster centers should be calculated, along with a standard deviation and estimate θ_l, [16], [6]:

1. Calculate the cluster centers of the Gaussian membership functions:

$$\mu_{ik}^l = \exp\left(-\gamma\left\|x_i^l - c_k^l\right\|^2\right). \tag{16}$$

Each data point $\mathrm{x}_k = (x_k, y_k)$ is assigned a potential P_k, according to its location with reference to all other data points:

$$Pv_i = \sum_{k=1}^{n} \exp(-\gamma\|x_i - c_k\|^2)\,,\ \text{with}\ \gamma = \frac{4}{r_a^2}\ , \tag{17}$$

where Pv_i is the potential-value i_data as a cluster centre, α is the weight between i_data to k_data, x is the data point, and r_a is the cluster radius, The square of the $\|x_i - c_k\|^2 = D_{i,k}^2$ is the distance of the clustering model.

The data point with the highest potential, denoted by Pv_i is considered as the first cluster center $c_i = (d_1, e_1)$. The potential is then recalculated for all other points excluding the influence of the first cluster center according to [9], [19]:

$$Pv_i = Pv_i - \overline{Pv_j}\, \boldsymbol{e}^{-\gamma\|x_i - c_j\|^2}\,,\ \text{with}\, r_b = r_a * \eta. \tag{18}$$

$\overline{Pv_j}$ is the potential-value data as cluster centre, c_j is the cluster center of data, γ is the weight of i_data to cluster centre, r_b is the distance between the cluster centre, η is the quash factor.

Again, the data point with the highest potential $\overline{Pv_j}$ is considered to be the next cluster center c_j, if:

$$\frac{d}{r_a} + \frac{\overline{Pv_J}}{\overline{Pv_1}} \geq 1, \tag{19}$$

where d is the distance between c_1 and all previously found cluster centers, and the data point is still accepted as the next cluster center c_1. Further iterations can then be performed to obtain new cluster centers c_2. The data point with the next highest potential $\overline{Pv_k}$ is selected as the new possible cluster center and re-tested. The clustering ends if the following condition is [11], [13], [16]:

$$\overline{Pv_j} < \varepsilon . Pv_i\ , \tag{20}$$

where ε is the reject ratio. Each cluster center is considered as a fuzzy rule that describes the system behavior of the distance to the defined cluster centers:

$$\mu_{ik}^l = \exp\left(-\gamma\left\|x_i^l - c_k^l\right\|^2\right) \tag{21}$$

which is a form of subtractive clustering.

2. Estimate the parameters of the models by weighted least squares. The weighted parameter estimation approach does not estimate all parameters. The parameters of the models are estimated separately, using a set of estimation criteria [18].

$$\min_{\theta_l} = \|y - \Phi_{\mathrm{e}}\theta_l\|^2 \beta_l = \left\|y\sqrt{\beta_l} - \Phi_{\mathrm{e}}\sqrt{\beta_l}\theta_l\right\|^2 \tag{22}$$

where Φ_{e}denotes the extended regression matrix obtained by adding a unitary column to Φ, $\Phi_{\mathrm{e}} = [\Phi\ \ 1]$, where $\Phi = [\Phi_1, \Phi_2, \dots, \Phi_N]^T$, $\mathrm{y} = [y_1, y_2, \dots, y_N]^T$, and β_l denotes a diagonal matrix having membership degrees in diagonal elements. $\beta_l = \begin{pmatrix} \mu_{l,1} & 0 & \dots & 0 \\ 0 & \mu_{l,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_{l,N} \end{pmatrix}$ (23)

The weighted least-squares estimate of the consequent rule parameters is given by:

$$\theta_l = \left(\Phi_{\mathrm{e}}^{\mathrm{T}}\, \beta_l\, \Phi_{\mathrm{e}}\right)^{-1} \Phi_{\mathrm{e}}^{\mathrm{T}}\, \beta_l\, y. \tag{24}$$

3. Calculate the δ_l standard deviation of the modeling error:

$$\delta_l^2 = \frac{\sum_{k=1}^{N}(y_k - f_l(\Phi_k, \theta_l))^T (y_k - f_l(\Phi_k, \theta_l))\mu_{lk}}{\sum_{k=1}^{N}\mu_{lk}}, \tag{25}$$

where $f_l(\Phi_k, \theta_l) = \hat{y}_k$ is the output of model.

4.2 Fuzzy C-Means (FCM) Model

The fuzzy c-means model uses iterative optimization to approximate minima of an objective function which is a member of a family of fuzzy c-means functions using a particular inside product norm metric. We can denote fuzzy c partitions of x by [6], [9], [12]:

$$\mu_{fc} = \{U \in [\mu_{lk}]_{C\times n}\ |0 \leq \mu_{lk} \in [0,1],\ \ \forall\, l,k,\ \ \ \sum_{l=1}^{C}\mu_{lk} = 1,\ \ \forall\, k\, \&\ \ 0 < \sum_{k=1}^{n}\mu_{lk} < n,\ \ \forall\, l\}$$

where $\mu_{i,k}$ are the elements of the matrix and represent the degree of the membership. The fuzzy c-means functional J_q is called objective function defined as [10], [12], [13], [18]:

$$J_q(U,c) = \sum_{k=1}^{n}\sum_{l=1}^{C}\mu_{lk}^{q}\, d^2(x_k, c_l), \tag{26}$$

where $U \in \mu_{fc}$ is a fuzzy c partition of x; $c = (c_1, c_2, \dots, c_C)$ the clusters , $1 \le l \le C$, and $d_{lk}^2 = \|x_k - c_l\|^2$. The basic steps of the model repeat for $q \in [1, \infty)$ (q is the iteration counter) that are given as follows [7]:

Step1: Calculate the parameters of the clusters:

1. Calculate the centers and standard deviation of the Gaussian membership functions:

$$c_l{}^{(m)} = \frac{\sum_{k=1}^{n}\mu_{lk}^{(q-1)}x_k}{\sum_{k=1}^{n}\mu_{lk}^{(q-1)}}, \tag{27}$$

$$\delta_{lj}^{2\,(m)} = \frac{\sum_{k=1}^{n}\mu_{lk}^{(q-1)}(x_{jk}-c_{jk})^2}{\sum_{k=1}^{n}\mu_{lk}^{(q-1)}},\ \ 1 \le l \le C. \tag{28}$$

2. Standard deviation of the modeling error:

$$\delta_l^2 = \frac{\sum_{k=1}^{N}(y_k - f_l(\Phi_k,\theta_l))^T(y_k - f_l(\Phi_k,\theta_l))\mu_{lk}^{(q-1)}}{\sum_{k=1}^{N}\mu_{lk}^{(q-1)}} \tag{29}$$

3. Calculate parameters of the local models:

4. We can calculate θ_l by the same procedure as in the *SFC* model:

$$\theta_l = \left(\Phi_e^T\,\beta_l\,\Phi_e\right)^{-1}\Phi_e^T\,\beta_l\,y. \tag{30}$$

5. Calculate weights of the rules:

$$\omega_l = \prod_{j=1}^{n}\frac{\left(\sum_{k=1}^{n}\mu_{lk}\right)}{n.\left(\sqrt{2\pi\delta_{lj}^2}\right)}, \tag{31}$$

Step2: Compute the distance measure (d_{ik}^2):

The distance measure consists of two terms. The first term is based on the geometrical distance between the c_k cluster centers and x_i input vector, while the second is based on the performance of the linear models:

$$d_{ik}^2 = \frac{1}{\prod_{j=1}^{n}\frac{\left(\sum_{k=1}^{n}\mu_{lk}\right)}{n.\left(\sqrt{2\pi\delta_{lj}^2}\right)}.\exp\left(-\frac{1}{2}\frac{\left(x_{jk}-c_{jk}\right)^2}{\delta_{lj}^2}\right).\ \frac{1}{\sqrt{2\pi\delta_l^2}}\exp\left(-\frac{\left(y_k-f_l(\Phi_k,\theta_l)\right)^T\left(y_k-f_l(\Phi_k,\theta_l)\right)}{2\delta_l^2}\right)} \tag{32}$$

$$= \frac{1}{\prod_{j=1}^{n}\frac{\left(\sum_{k=1}^{n}\mu_{lk}\right)}{n.\left(\sqrt{\left(2\pi\delta_{lj}^2\right)\left(2\pi\delta_l^2\right)}\right)}.\exp\left(-\frac{\left(x_{jk}-c_{jk}\right)^2}{2\delta_{lj}^2}-\frac{\left(y_k-\hat{y}_k\right)^T\left(y_k-\hat{y}_k\right)}{2\delta_l^2}\right)}$$

$$= \frac{1}{\prod_{j=1}^{n}\frac{\left(\sum_{k=1}^{n}\mu_{lk}\right)}{n.\left(\sqrt{4\pi^2\delta_{lj}^2.\delta_l^2}\right)}.\exp\left(-\frac{1}{2}\left(\frac{\delta_l^2.\left(x_{jk}-c_{jk}\right)^2+\delta_{lj}^2.\left(\left(y_k-\hat{y}_k\right)^T\left(y_k-\hat{y}_k\right)\right)}{\delta_{lj}^2.\delta_l^2}\right)\right)}$$

$$d_{ik}^2 = \frac{2n\pi\sqrt{\delta_{lj}^2.\delta_l^2}}{\prod_{j=1}^{n}\sum_{k=1}^{n}\mu_{lk}.\exp\left(-\frac{1}{2}\left(\frac{\delta_l^2.\left(x_{jk}-c_{jk}\right)^2+\delta_{lj}^2.\left(\left(y_k-\hat{y}_k\right)^T\left(y_k-\hat{y}_k\right)\right)}{\delta_{lj}^2.\delta_l^2}\right)\right)}$$

$$d_{ik}^2 = \prod_{j=1}^{n}\sum_{k=1}^{n}\frac{2n\pi}{\mu_{lk}}\sqrt{\delta_{lj}^2.\delta_l^2}.\exp\left(\frac{\delta_l^2.\left(x_{jk}-c_{jk}\right)^2+\delta_{lj}^2.\left((y_k-\hat{y}_k)^T(y_k-\hat{y}_k)\right)}{2\delta_{lj}^2.\delta_l^2}\right) \tag{33}$$

Step3: Update the partition matrix:

$$\mu_{lk}^{(m)} = \frac{1}{\sum_{j=1}^{C}\left(\frac{d_{lk}}{d_{jk}}\right)^{\frac{2}{(q-1)}}},\ \ 1 \le l \le C, 1 \le k \le n. \tag{34}$$

until $\left\|U^{(m)} - U^{(m-1)}\right\| < \varepsilon$, stop; otherwise.

5. STRUCTURE of ANFIS USING CLUSTERING with TS MODEL

Suppose $x = [x_1\ x_2\ \dots\ x_n]$ is the input vector, $c_k = [c_{1k}\ c_{2k}\ \dots\ c_{rk}]$ is the center cluster of membership functions in the k^{th} neuron, and $\delta_k = [\delta_{1k}\ \delta_{2k}\ \dots\ \delta_{rk}]$ is the width vector of membership functions in the k^{th} neuron, [8], [14].

1. The *Input layer* is the first layer in *ANFIS,* where each neuron in this layer represents an input variable $x = (x_1, x_2, \dots, x_n)^T$,

$$\widehat{y1}_l = \mu_{A_{lk}}(x_k) = \exp\left(\frac{(x_k-c_{lk})^2}{2\delta_{lk}^2}\right) \tag{35}$$

2. The *Hidden layer* consists of a *fuzzifer layer*, a *normalized layer*, and a *weighted layer*. In this layer all the steps that are explained above for *SFC* and *FCM* of *TS* model are applied:

$$\widehat{y2}_l = \prod_{k=1}^{n}\mu_{A_{lk}}(x_k) = \prod_{k=1}^{n}\exp\left(\frac{(x_k-c_{lk})^2}{2\delta_{lk}^2}\right), \text{ then}$$

$$\widehat{y2}_l = \exp\left(\sum_{k=1}^{n}\frac{(x_k-c_{lk})^2}{2\delta_{lk}^2}\right) \tag{36}$$

$$\widehat{y3}_l = \frac{\widehat{y2}_l}{\sum_{l=1}^{C}\widehat{y2}_l} = \frac{exp\left(\sum_{k=1}^{n}\frac{\left(x_k-c_{lk}\right)^2}{2\delta_{lk}^2}\right)}{\sum_{l=1}^{C}exp\left(\sum_{k=1}^{n}\frac{\left(x_k-c_{lk}\right)^2}{2\delta_{lk}^2}\right)} \tag{37}$$

$$\widehat{y4}_l = \widehat{y3}_l.\left(a_l^T\phi_k + b_l\right) \tag{38}$$

3. The *Output layer*. The output of the neuron in this layer is:

$$\widehat{y5}_l = \sum_{l=1}^{C}\widehat{y4}_l \tag{39}$$

where $\hat{y}$ is the value of an output variable.

A feed forward adaptive network is actually a static mapping between its input and output spaces; this mapping may be either a simple linear relationship or a highly nonlinear one, depending on the structure for the network and the function for each node [2], [19], [20]. Here our aim is to construct a network for achieving a desired nonlinear mapping that is regulated by a data set consisting of a number of desired input-output pairs of a target system. This data set is usually called the training data set and the procedure we follow in adjusting the parameters to improve the performance of the network is often referred to as the learning rule. Usually an adaptive network's performance is measured as the discrepancy between the desired output and the network's output under the same input conditions. This discrepancy is called the error measure.

5.1 Mathematical Model of the *Learning Rule*

The central part of a learning rule for an adaptive network is concerned with how to recursively get a slope vector in which each element is defined as the derivative of an error measure with respect to a parameter [9], [11], [13]. The model is generally referred to as the "back-propagation learning rule" because the slope vector is calculated in the direction opposite to the flow of the output of each node.

Suppose a given feed forward adaptive network has L layers and the k^{th} layer has $*(k)$ nodes. We can denote the node in the k^{th} position of the k^{th} layer by (k, i), and its node output by $\hat{y}_i^k$. Since a node output depends on its incoming signals and its parameter set, we have [2], [5], [10]:

$$\hat{y}_i^k = \hat{y}_i^k(\hat{y}_i^{k-1}, \dots, \hat{y}_{*(k-1)}^{k-1}, a\, b, c, \dots), \tag{40}$$

where $a\, b, c,$ ...etc, are the parameters to this node. Assuming the given training data set has P (patios) entries, we can define the error measure for the $p^{th}(1 \le p \le P)$, entry of training data as the sum of squared errors:

$$E_p = \sum_{m=1}^{*(L)} \left(T_{m,p} - \hat{y}_{m,p}^L\right)^2, \tag{41}$$

where $T_{m,p}$, the m^{th}component of p^{th} target is the output vector, and $\hat{y}_{m,p}^L$ is the m^{th}component of the actual output vector produced by the presentation of the p^{th} input vector. Then the overall error measure is:

$$E = \sum_{p=1}^{P} E_p \tag{42}$$

In order to develop a learning procedure that implements slope descent in E over the parameter space, first we must calculate the error rate $\partial E_p / \partial \hat{y}$ for p^{th} training data and for each node output $\hat{y}$. The error rate is $\partial E_p / \partial Y$ for p^{th}. The error rate for the output note at (L, i) can be calculated easily from (41), [18]:

$$\frac{\partial E_p}{\partial \hat{y}_{i,p}^L} = -2(T_{i,p} - \hat{y}_{i,p}^L) \tag{43}$$

For internal node at (k, i), the error rate can be derived by:

$$\frac{\partial E_p}{\partial \hat{y}_{i,p}^L} = \sum_{m=1}^{*(k+1)} \frac{\partial E_p}{\partial \hat{y}_{m,p}^{k+1}} \; \frac{\partial \hat{y}_{m,p}^{k+1}}{\partial \hat{y}_{i,p}^k}, \; 1 \le k \le L-1 \tag{44}$$

That is, the error rate of an internal node can be expressed as a linear combination of the error rates of the nodes in the next layer. Therefore $\forall\, 1 \le k \le L$ and $1 \le i \le *(k)$, we can find $\partial E_p / \partial \hat{y}_{i,p}^k$ by (43) and (44).

Now, if β is a parameter of the given adaptive network [6], [8], we have:

$$\frac{\partial E_p}{\partial \beta} = \sum_{\hat{y}^* \in S} \frac{\partial E_p}{\partial \hat{y}^*} \; \frac{\partial \hat{y}^*}{\partial \beta}, \tag{45}$$

where S is the set of nodes whose outputs depend on β. Then the derivative of the overall error measure E with respect to β is:

$$\frac{\partial E}{\partial \beta} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \beta} \tag{46}$$

The update formula for the generic parameter β is:

$$\Delta \beta = -\eta \frac{\partial E}{\partial \beta} \tag{47}$$

In which, η is a learning rate which can be further expressed as:

$$\eta = \frac{k}{\sqrt{\sum_\beta \left(\frac{\partial E}{\partial \beta}\right)^2}}, \tag{48}$$

where k is the step size, the length of each slope transition in the parameter space.

Table 1. The inputs with range selected respectively.

Inputs	Symptoms (Symbol)	Range
X_1	Fever (F.)	[99-107]
X_2	Malaise (ma.)	[0-1]
X_3	Anorexia (An.)	[0.10-0.90]
X_4	Headache (He.)	[0.25-0.75]
X_5	Cough (Co.)	[0.05-0.95]
X_6	Abdominal Pain (Ab.)	[0.10-0.90]
X_7	Diarrhea (Di.)	(5-8) times P. 12 h.
X_8	Lethargy (Le.)	[0.05-0.4]
X_9	Skin Rash (SR.)	[0.10-0.80]
X_{10}	Myalgia (My.)	[0-1]
X_{11}	Delirium (De.)	[0.20-0.80]
X_{12}	Difficulty concentrating Dc.	[0-1]
X_{13}	Mild coryza (Mc.)	[0-1]
X_{14}	Enlarged lymph & pain(Ep.)	[0.10-0.90]
X_{15}	Moderate Throat pain (Mp.)	[0.10-0.80]
X_{16}	Rhinitis (Rh.)	[0.20-1]
X_{17}	Hoarseness of voice (Hv.)	[0.25-0.85]
X_{18}	Rigor (Ri.)	(2-10) times P. 12 h.
X_{19}	Swelling of salivary glands Ss.	[0.10-0.90]
X_{20}	Difficulty in Swallowing DS.	[0-1]
X_{21}	Skin Rash with pruritus SR.	[0.10-0.90]
X_{22}	Vomiting (v.)	(2-10) times P. 12 h.
X_{23}	Difficulty breathing (Db.)	[0-1]
X_{24}	Paleness (pl.)	[0.20-0.80]
X_{25}	Muscle cramp (Mcr.)	[0-1]
X_{26}	Sickness (S.)	[0.25-0.75]
X_{27}	Heart beat (Hb.)	(85-250) time P. m.
X_{28}	Paralysis (pr.)	[0.10-0.70]
X_{29}	Thyroid swelling (Ts.)	[0.10-0.90]
X_{30}	Pain in Thyroid gland (PT.)	[0.10-0.80]
X_{31}	Blood with spit & phlegm Bsp	[0-1]
X_{32}	Night sweats (Ns.)	[0.25-0.75]

6. The PROPOSED MODEL

In this section, we will apply all previous concepts on application of medical diagnosis and the comparison the results of *SFC* and *FCM* without using *ANFIS* as following:

6.1 Description of Problem

In this application of medical diagnosis, we chose acceptable and real data because the system has to deal with the patient's health and even life, and this real data depends on a continuous range of numbers according to a range of symptoms [15]. The data base has given us a group of doctors and staff specialists from a combination of 3 hospitals (“*Mosul General Hospital*”, “*Al Kindi General Teaching Hospital*”, and “*Specialized Hospital Eben Al Athir for Children*”) merged into a single dataset for research purposes. Table (1) illustrates the symptoms (inputs from x_1 *to* x_{32}) selected respectively with a range for all symptoms.

We selected *100* patients from the information of a real dataset for many visitors of a hospital with ten patients per disease for a total of 10 different diseases (*Typhoid Fever, Rubella, Acute Laryngitis, Mumps, Chicken pox, Malaria, Anemia, Polio, Goiter, and Tuberculosis*). Table (2) shows the list of diseases and their assigned labels from *1 to 10* respectively. This dataset provided *32* symptoms of ten different diseases. We have processed the real data to conduct the process of scaling to all data based on linear equations and obtained scaled data limited between 0 and 1. From these scaled data we obtained a matrix of dimensions (*100x32*) for a hundred patients (in rows), where the *32 column* indicate the symptoms (*inputs*) for ten different diseases (*output*). Each disease depends on some of the symptoms: that is the relationship between inputs and output. The data matrix has been identified depending on this relationship between symptoms and disease that correlate with these symptoms as shown in table (3). Representing the real data we have available for the application of medical diagnostics enables us to draw the clusters in separate groups and in points. Each group has a different color, so that each disease is represented by a different color. See Fig. (1).

Table (2): Disease names and their labels.

Assigned label	Disease name (output)	Assigned label	Disease name (output)
1	Typhoid Fever	6	Malaria
2	Rubella	7	Anemia
3	Acute Laryngitis	8	Polio
4	Mumps	9	Goiter
5	Chicken pox	10	Tuberculosis

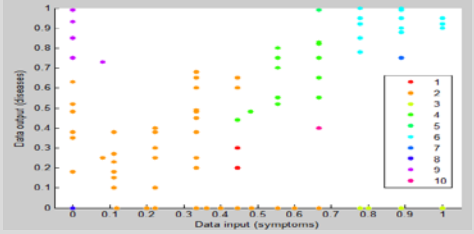

Fig.1: The real data medical diagnostics.

6.2 Experiments and Discussion

We will now load the inputs and output variables into the workspace. Data inputs have 32 columns (symptoms) and data output has 1 column (diseases) representing the output variable, as shown in Fig.(2). The number of rows is 100 and represents the number of patients. The columns from 1 to 32 represent the values of variables for the symptoms, and the last column (*33th column*) represents the output values of the variable of the diseases. We modeled the relationship between the input and the output variables by using the concept of clustering of the data.

Table 3. The relationship between inputs and output.

Disease	Y_1 : Typhoid Fever = $(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12})$
	Y_2 : Rubella = $(x_1 + x_9 + x_{13} + x_{14})$
	Y_3 : Acute Laryngitis = $(x_1 + x_2 + x_3 + x_{15} + x_{16} + x_{17})$
	Y_4 : Mumps = $(x_1 + x_2 + x_3 + x_4 + x_{18} + x_{19} + x_{20})$
	Y_5 : Chicken pox = $(x_1 + x_2 + x_{21})$
	Y_6 : Malaria = $(x_1 + x_2 + x_3 + x_4 + x_7 + x_8 + x_{10} + x_{12} + x_{18} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27})$
	Y_7 : Anemia = $(x_2 + x_3 + x_4 + x_8 + x_{12} + x_{23} + x_{24} + x_{27})$
	Y_8 : Polio = $(x_1 + x_4 + x_{20} + x_{22} + x_{25} + x_{26} + x_{28})$
	Y_9 : Goiter = $(x_1 + x_{20} + x_{23} + x_{29} + x_{30})$
	Y_{10} : Tuberculosis = $(x_1 + x_2 + x_3 + x_5 + x_8 + x_{23} + x_{31} + x_{32})$

Now we will build an adaptive *neuro-fuzzy inference system* (*ANFIS*) based on the clustering process, and the first thing we do is to load data in the workspace. Of the original *100* patients, we use *75* patients as training data and *25* patients as checking data. After loading data, we generate a model from data using clustering, to determine which method we will use in the process of clustering. We can generate an *FIS* structure using subtractive fuzzy clustering (*SFC*) and fuzzy c-main (*FCM*) models. To use the clustering technique, we should select the clustering type of the *ANFIS* program before the *FIS* is generated.

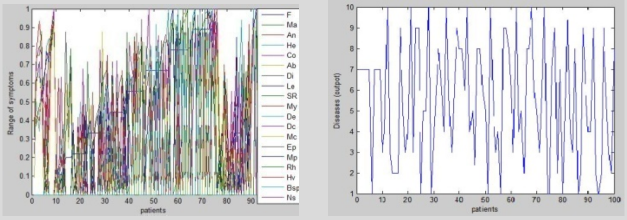

Fig.2: Representation the input and output variables.

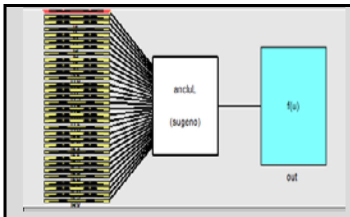

Fig.3: The structure for building *FIS*.

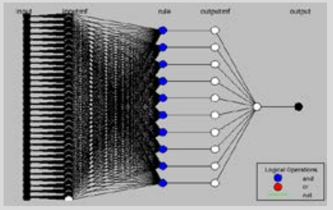

Fig.5: The model structure.

The *SFC* and *FCM* models partition the data into groups called clusters, and generate an *FIS* with the minimum number of rules required to distinguish the fuzzy qualities associated with each of the clusters. The *FIS* is composed of inputs, output and rules. See Fig.(3). The number of rules equals the number of clusters and hence *10* rules are created.

Notice, all the inputs and outputs have exactly *10* membership functions for each input variable. The *10* membership functions represent the*10* clusters that were identified by *SFC & FCM*. Notice that the membership function type is *"gaussmf"* (*Gaussian MF*) and that the parameters of the membership function are linear, that is denoted by the mathematical formula:

$$\mu(x;a,b) = exp\left(-\frac{(x-a)^2}{2b^2}\right), \text{where } a,b \text{ parameters} \qquad (49)$$

$$R_1 : IF\ x_1\ is\ A_1^{(1)}\ and\ x_2\ is\ A_2^{(1)} and\ \ \dots\ \ and\ x_{32}\ is\ A_{32}^{(1)}\ \ THEN$$

$$f_1 = \ a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,32}x_{32} + a_{1,33}$$

$$R_2 : IF\ x_1\ is\ A_1^{(2)}\ and\ x_2\ is\ A_2^{(2)} and\ \ \dots\ \ and\ x_{32}\ is\ A_{32}^{(2)}\ \ THEN$$

$$f_2 = \ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,32}x_{32} + a_{2,33}$$

:

$$R_{10} : \underbrace{IF\ x_1\ is\ A_1^{(10)} and\ x_2\ is\ A_2^{(10)} and \dots and\ x_{32}\ is\ A_{32}^{(10)}}_{Antecedent\ part}\ THEN$$

$$\underbrace{f_{10} = \ a_{10,1}x_1 + a_{10,1}x_1 + \cdots + a_{10,32}x_{32} + a_{10,33}}_{Concequent\ part} \qquad (50)$$

where $x_1, x_2, \dots, x_{32}$ are the symptoms variable. See table (1). $A_j^{(i)}$ *MF* is according to variables ($i = 1\ to\ 10$ and $j = 1\ to\ 32$), 🐦🐦 are the rules, 🐦🐦 as the linear function of output (diseases), {🐦🐦,🐦} are the parameters. Let's explore how the fuzzy rules are constructed. There are exactly *10* rules. Each rule attempts to map a cluster in the input space to a cluster in the output space. The first rule can be explained simply as follows: If the inputs to the *FIS* are *High Fever, Malaise, Anorexia, Headache, Cough, Abdominal Pain, Diarrhea, Lethargy, Skin Rash, Myalgia, Delirium, and Difficulty concentrating* (all of which strongly belong to their respective cluster1 *MFs*) then the output "*Typhoid Fever*" must strongly belong to its *cluster1 MF*. The other rules in the output space the function similarly. The output of the *FIS*, diseases, has *10* linear *MFs* (*ST* model) representing the *10* clusters identified by the *SFC & FCM* models. Parameters are estimated from the dataset using the least squares estimation technique by (24), (30).We can now use the *FIS* that has been constructed by the *ST* model with the *SFC & FCM* models to understand the relationship being modeled. We can use command windows, "*surfview*" the surface viewer that helps view the input-output surface of the fuzzy system. This conception is very helpful to understand how the system is going to behave for the entire range of values in the input space, as shown in Fig. (4).

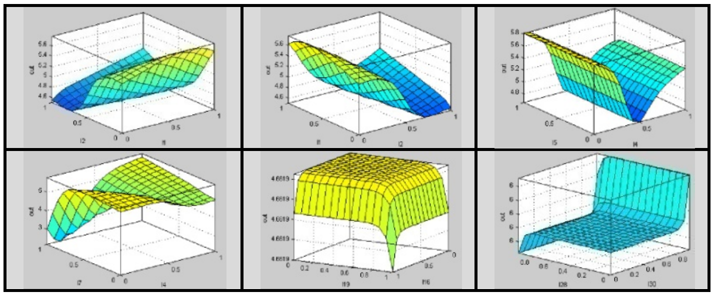

Fig.4: Some surface viewers b/w inputs and output.

Now, having built the fuzzy system, if we want to understand which disease will occur for particular symptoms, than this tool will help us to simulate the *FIS* response for the input of our choice. After completing the system, we save it, and will go back again to the *ANFIS* and load *FIS* from file. Now, an *FIS* structure is returned; the model type for the *FIS* structure is a first order *TS* model with *10* rules that are applied the first time with the *SFC* model and the second time with the *FCM* model. From the program of *ANFIS* the structure of the system appears as shown in Fig.(5). We have selected a type of the *Learning Rule* (*back-propagation learning rule*) as an *ANFIS* parameter optimization method for *FIS* training. This reduces the problem of too much propagation of rules when the input data has a high dimension (as in our application). In the next step, we perform a system at *50 epochs* and get the average testing error of training data and the average testing error of checking of *ANFIS-SFC*, *ANFIS-FCM* and *ANFIS* without clustering. Additionally, the number of nodes are a number of linear parameters and a number of nonlinear parameters, as shown in table(4) is obtained where the better results of *Training error, ATE* of training data and *ATE* of checking data was obtained with *ANFIS-SFC* (*0.013637*) when the *Back-propagation* model of the Learning Rule was used. The best results of *Training error, ATE* of training data and *ATE* of checking data was obtained with *ANFIS-FCM* (*0.00002172*) when the *Hybrid* model of the Learning Rule was used. See Fig.(6a, 6b, 6c). We tried to compare the real output of medical diagnoses with the results that we got from the systems for three cases namely with *ANFIS-SFC*, *ANFIS-FCM* and *ANFIS* without clustering used. The performance of the models is measured by the square root of the mean squares prediction error (*RMSE*):

$$\text{🐦🐦🐦🐦} = \sqrt{\frac{(\text{🐦}_{\text{🐦}} - \hat{\text{🐦}}_{\text{🐦}})^2}{\text{🐦}}}$$

(51)

Using (51) we obtained a very small error, and the smallest value of Average *RMSE* was obtained with *ANFIS-FCM,* as the following table (5) illustrates. So we have built a system that has proved its efficiency in the medical diagnoses.

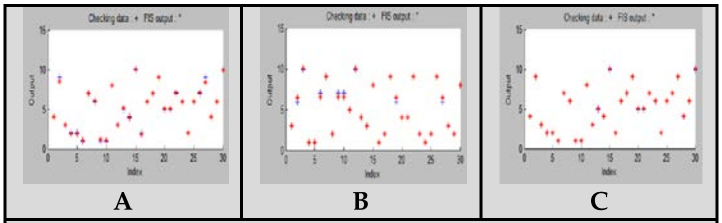

Fig.6. Representation of the average testing error of *ANFIS-SFC*, *ANFIS-FCM* and *ANFIS* respectively.

In the second part of our application, we have applied the clustering without a neural network on the data

{🐦$_{1}$, 🐦$_{2}$,.., 🐦$_{32}$} that is represented by 100 cases. The first time, we applied the *FCM* model depending on (26) to (34) with 10 clusters. The second time, we applied the *SFC* model depending on (16) to (25) with the same number of clusters. When these steps were applied to the *SFC* model by using *MATLAB*, we obtained the matrix of optimal *cluster centers C*. See Fig.(7a). Each row of *C* contains the position of a cluster center.

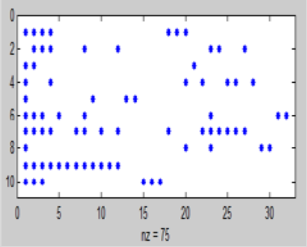

Fig.7a. Cluster centers matrix of *SFC.*

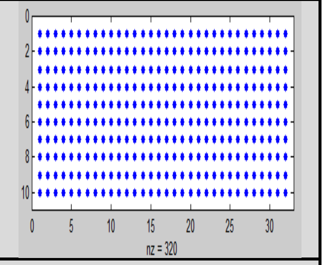

Fig.7b. Cluster centers matrix of *FCM.*

Table 4. Comparison of the *ATE* of training & checking data of three cases with no. of parameters.

Type of Learning Rule	Type modeling	Training error	*ATE* of training data	*ATE* of checking data	No. of nodes	No. of L.Ps	No. of non-L.Ps
Back-propagation	***ANFIS-SFC***	***0.013637***	***0.01363***	***0.0041308***	695	330	640
Hybrid		0.00015227	0.00014509	0.13004			
Back-propagation	***ANFIS-FCM***	0.321	0.32017	0.35873	695	330	640
Hybrid		***0.00002172***	***0.00002174***	***0.07341***			
Back-propagation	***ANFIS***	0.2341	0.23271	0.22721	119	10	96
Hybrid		0.14078	0.14078	0.19845			

We also obtained the *vector S* that contains the sigma values that specify the range of effect of a cluster center in each of the data dimensions. All cluster centers share the same set of sigma values. We represented the *cluster centers* with clusters as in Fig.(8), where the symbols 'o' represent data points for each cluster component and a big circle in the center of each cluster represents the cluster center. Each cluster is represented by a different color. However, when these steps are applied to the *FCM* model, we have obtained a matrix of final *cluster centers* where each row provides the center coordinates. See Fig.(7b). We also obtained final membership function matrix *U*. The plot in Fig.(9) displays the surface of *MF* to each of the ten separate clusters, and values of the objective function during iterations (*obj-fcn*). This value started from *132.22* and went up to *10.20* at iteration *22*, as shown in Fig.(10).

Finally, when we have selected the number of clusters (*N=10*), we got to *10 clusters* after applying clustering programs, where each cluster includes 10 patients suffering from the same disease, where:

The *First cluster* contains all patients who have symptoms of *Typhoid Fever*.

The *Second cluster* contains all patients who have symptoms of *Mumps*.

The *Third cluster* contains all patients who have symptoms of *Acute Laryngitis*.

The *Fourth cluster* contains all patients who have symptoms of *Malaria*.

The *Fifth cluster* contains all patients who have symptoms of *Chicken pox*.

The *Sixth cluster* contains all patients who have symptoms of *Goiter*.

The *Seventh cluster* contains all patients who have symptoms of *Rubella*.

The *Eighth cluster* contains all patients who have symptoms of *Polio*.

The *Ninth cluster* contains all patients who have symptoms of *Anemia*.

The *Tenth cluster* contains all patients who have symptoms of *Tuberculosis*.

Table 5. Comparison of the results of *ANFISs* with real outputs.

No	*Real output*	*ANFIS output*	*Error RMSE*	*ANFIS-SFC output*	*Error RMSE*	*ANFIS-FCM output*	*Error RMSE*
1	7	6.8	0.0365	7	0	7.01	0.0018
2	7	6.99	0.0018	6.9	0.0183	6.92	0.0146
3	7	6.8	0.0365	6.74	0.0475	6.81	0.0347
4	1	0.99	0.0018	0.933	0.0122	0.99	0.0018
5	1	0.98	0.0037	1.05	0.0091	0.96	0.0073
6	1	0.99	0.0018	0.826	0.0318	0.95	0.0091
7	3	3	0	3.2	0.0365	3	0
8	3	3	0	2.94	0.0110	3	0
9	3	3	0	3.23	0.0420	3	0
10	4	4	0	3.99	0.0018	4	0
11	4	4	0	3.97	0.0055	4	0
12	4	4	0	3.99	0.0018	4	0
13	10	10	0	10.02	0.0037	10	0
14	10	10	0	9.93	0.0128	10	0

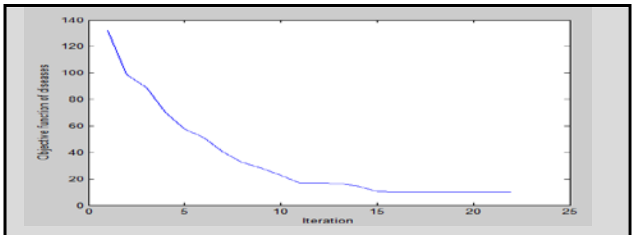

Fig.10. Objective function of ten diseases.

15	10	10	0	10.2	0.0365	10	0
16	2	2	0	2.01	0.0018	2	0
17	2	2	0	2	0	2	0
18	2	2	0	2	0	2	0
19	9	9	0	8.99	0.0018	9	0
20	9	9	0	9.03	0.0055	9	0
21	9	9	0	9.02	0.0037	9	0
22	5	5	0	5	0	5	0
23	5	5	0	5	0	5	0
24	5	5	0	5	0	5	0
25	6	5.57	0.0785	6.26	0.0475	6.26	0.0475
26	6	5.65	0.0639	5.96	0.0073	5.89	0.0201
27	6	6.13	0.0237	6.17	0.0310	6.17	0.0310
28	8	8	0	8	0	8	0
29	8	8	0	8.01	0.0018	8	0
30	8	8	0	8	0	8	0
Average RMSE		***0.0083***		***0.0124***		***0.0056***	

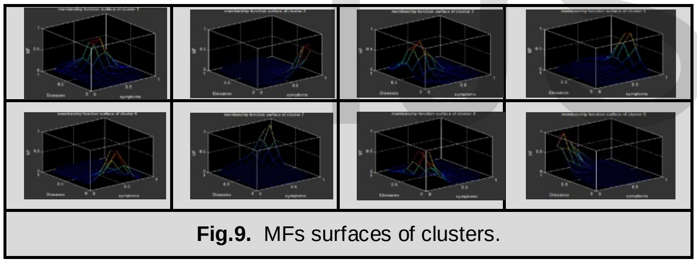

Fig.9. MFs surfaces of clusters.

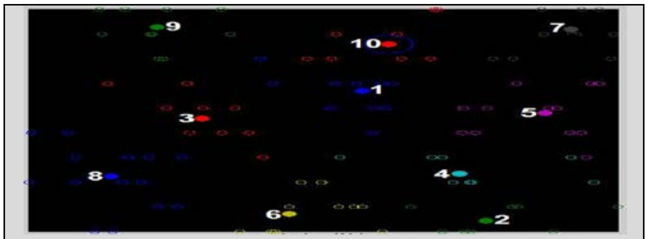

Fig.8. The clustering of tenth diseases.

7. CONCLUSION

This application has attempted to convey how clustering and fuzzy systems can be employed as effective techniques for data modeling with many different types of diseases. The main variables in medical diagnosis and prediction use intelligence techniques to be more interactive. Expert systems and fuzzy systems will be suitable techniques for dealing with uncertainty regarding the effects of the proposed models. For the prediction tasks, Neural Networks with fuzzy systems have been proven to produce better results compared to other techniques. In this work we compared three different techniques, first *ANFIS-SFC*, second *ANFIS-FCM* and third *ANFIS* without clustering, where each one of these techniques depends on different models of clustering. From that we noted the best results of the *Training error*, the *ATE* of the training data and the *ATE* of checking data with *ANFIS-SF* (*0.013637*) when the *Back-propagation* model of the Learning Rule was used. We have obtained the best results of the *Training error*, the *ATE* of the training data and the *ATE* of the checking data with *ANFIS-FCM* (*0.00002172*) when the *Hybrid* model of the Learning Rule was used. The smallest value of the average *RMSE* was obtained with the *ANFIS-FCM* (*0.0056*). Also we have compared the results of *SFC* and *FCM* without using *ANFIS*, where results for different matrices of cluster centers are denoted by fig.(9a) and (9b). However, in both models we got *10 clusters* and each cluster includes 10 patients suffering from the same disease. We note that these techniques are very effective with mathematical modeling of medical diagnosis. We can improve this work by using other models of clustering like G-K clustering, that depends on the concept of fuzzy modeling or improves this system to find treatment of each case.

ACKNOWLEDGEMENTS

The author wishes to thank the reviewers for their excellent suggestions that have been incorporated into this paper. I would like to thank and acknowledge Prof. Deepak Gupta, Head department of mathematics, M.M. University, Mullana, Ambala, India, who advised me to improve my work.

REFERENCE

[1] A. Abdennour (2005), "A long horizon neuro-fuzzy predictor for mpeg video traffic". J. King Saud University, Vol. 18, pp. 161-180.

[2] A. Abraham (2005), "Measuring System Design", Oklahoma State University, Stillwater, OK, USA, edited by Peter H. Sydenham and Richard Thorn. John Wiley & Sons, Ltd. ISBN: 0-470-02143-8.

[3] E. Al-Gallaf (2005), "Clustered based takagi-sugeno neuro-fuzzy modeling of a multivariable nonlinear dynamic system", Asian Journal of Control, Vol. 7, No. 2, pp. 163-176.

[4] A. Baraldi, E. Binaghi, P. Blonda, A. Brivio, and A. Rampini, (2001), "Comparison of the multilayer perceptron with neuro-fuzzy techniques in the estimation of cover class mixture in remotely sensed data", IEEE Transactions on Geoscience and Remote Sensing, Vol. 39, No. 5, pp. 994-1005.

[5] S. Chopra, R. Mitra, and V. Kumar (2007), "A neuro-fuzzy learning and its application to control system", World Academy of Science, Engineering and Technology, Vol. 34, pp. 475-481.

[6] A. Elmzabi, M. Bellafkih, and M. Ramdani, (2001),"An adaptive fuzzy clustering approach for the network management", International Journal of Information Technology, Vol. 3, No. 1, pp. 12-17.

[7] S. Guillaume and B. Charnomordic (2004), "Generating an interpretable family of fuzzy partitions from data", IEEE Transactions on Fuzzy Systems, Vol. 12, No. 3, pp. 324-335.

[8] R. Hndoosh, M. S. Saroa and S. Kumar (2012), "Fuzzy and adaptive neuro-fuzzy inference system of washing machine", European Journal of Scientific Research, Vol. 86 No 3, pp.443-459.

[9] U. Inyaem, C. Haruechaiyasak, P. Meesad, and D. Tran, (2010), "Terrorism event classification using fuzzy inference systems", International Journal of Computer Science and Information Security, Vol. 7, No. 3, pp. 247- 256.

[10] C. Jose, R. Neil and W. Curt (1999), "Neural and Adaptive Systems". John Wiley and SONS, INC, New York.

[11] C. Ji-lin, H. Yuan-long, X. Zong-yi, J. Li-min, and T. Zhong-zhi (2006), "A multi objective genetic-based method for design fuzzy classification systems", International Journal of Computer Science and Network Security, Vol. 6, No. 8A, pp. 10-17.

[12] J. Jiang, R. Liou, and S. Lee (2011), "A fuzzy self constructing feature clustering algorithm for text classification", IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 3, pp.335-349.

[13] T. Kamel M. Hassan (2009), "Adaptive Neuro Fuzzy Inference System (ANFIS) For Fault Classification in the Transmission Lines", Journal on Electronics and Electrical Engineering, Vol. 2, No. 1, pp. 164-169.

[14] J. Lin, C. Cheng, Y. Sun and K. Chau, (2005), "Long-term prediction of discharges in man-wan hydropower using adaptive-network-based fuzzy inference systems models", Lecture Notes in Computer Science, Vol. 3612, pp. 1152-1161.

[15] S. Moein, S. Monadjemi, and P. Moallem (2008), "A novel fuzzy-neural based medical diagnosis system". World Academy of Science, Engineering and Technology, Vol. 37, pp. 443-459.

[16] A. Priyono, M. Ridwan, A. Alias, O. Rahmat, A. Hassan and M. Alauddin, (2005), "Generation of fuzzy rules with subtractive clustering", Journal Technology, Vol. 43, pp.143–153.

[17] D. Ruan and P. P. Wang (1997),"Intelligent Hybrid System: Fuzzy Logic, Neural Network and Genetic Algorithm", Kluwer Academic Publishers.

[18] S. Sivanandam and S. Deepa (2008), "Principal of Soft Computing", Wiley India (P) Ltd.

[19] P. Yang, Q. Zhu and X. Zhong, (2009), "Subtractive clustering based RBF neural network model for outlier detection", Journal of Computers, Vol. 4, No. 8, pp. 755-762.

[20] G. Zhang (2000),"Neural networks for classification: A survey", IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 30, No. 4, pp. 451-462.