Chapter 5

Comparative Risk Analysis on Prediction of Diabetes Dieses

5.1 Introduction

Machine learning techniques [28] proved their effectiveness in different fields including medicine [80], business [82], marketing [83], education [28], and many others. During the last decade, there has been an explosion of interest in medical machine learning research, due to the vast amount of data collected from individual patients and stored in databases. In addition, hospital errors are becoming area concern, where errors and carelessness so hospital staff were the reason behind 87% of hospital deaths in India [74]. By applying machine learning techniques to patient's databases, errors can be discovered and avoided. Inadiabetes diagnosis, machine learning models are used to discriminate between diabetic and non- diabetic individuals. Classification is used extensively in medicine as a pre-screening technique.

ANFIS is hybrid technique where the fusion is made between the neural network and the Fuzzy Inference System (FIS). This ANFIS methodology comprises of a hybrid system of fuzzy logic and neural network technique. The fuzzy logic takes into account the imprecision and uncertainty of the system that is being modeled while the neural network gives it a sense of adaptability. Using this hybrid method, at first an initial fuzzy model along with its input variables are derived with the help of the rules extracted from the input-output data of the system that is being modeled. Next the neural network is used to fine tune the rules of the initial fuzzy model to produce the final ANFIS model of the system.

ANN is non-linear, predictive models that learn through training. An ANN, usually called Neural Network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. An input is presented.

To the neural network and a corresponding desired or target response is set at the output using supervised training. An error is composed from the difference between the desired response and the system output. This error information is feedback to the system and adjusts the system parameters in a systematic fashion using the learning rule. The process is repeated until the performance is acceptable.

5.2 Literature Survey

Sl no.	Methodology	Authors, Year	Description
1	ANFIS	V. Anuja Kumari, R. Chitra , 2013 [66]	Makes fuzzy model as per the requirements of the problem. Then model is optimized by a training algorithm. The historical database used is of known inputs and outputs and are used for training. Supervised learning techniques are employed like Back Propagation Algorithm and Hybrid Learning
2	Female Diabetic Fuzzy Risk Classifier (FDFRC)	T.Jayalakshmi and Dr. A.Santhakumaran. ,2010 [56]	Employs a Mamdani type FIS technique and classifies the risk level of Diabetes by dividing the defuzzifier into low and high risk category.
3	Modular Neural Network (MNN)	Md. Osman Goni Nayeem and Muang Ning Wan ,2015 [54]	Exploits the modularity in the problem by dividing the problem into set of modules. All the modules individually solve the problem and then the results are supplied to central integrator. The integrator combines the results of various modules and gives final output.
4	Evolutionary Artificial Neural Network	Rahul Kala, Anupam Shukla, Ritu Tiwari,2009 [55]	It is a hybrid approach in which Artificial Neural Network is trained by using GA. It is a conventionalist approach. ANN's weights are evolved as fully connected model where every neuron of every layer is connected to all neurons of the other layer.
5	ANFIS-Group Method of Data Handling (GMDH) /Hierarchical TSK Fuzzy System	Arash Sharifi, Asiyesh Vosolipour,2008 [65]	GMDH uses 2 input ANFIS structure and Back Propagation algorithm is chosen for learning network structure.Type-3 ANFIS topology is used. Neural Network and Fuzzy Logic are model free estimators and share common abilities to deal with noise and uncertainties. Both of them encode information in parallel and distributed architecture in numerical framework.

5.3 Methodology

5.3.1 Dataset

Data from two hundred individuals were collected from Bhubaneswar, Odisha (India). The data consisted of 6 variables and the individuals comprised of both male and female between the age group of 21-71 years. The data was trained to identify the diabetes pattern using MATLAB13. They were asked to fill up the Questionnaire (See AppendixI) to predict the occurrence of diabetes with the given symptoms. (Appendix II) consists of the entire dataset.

5.3.2 Adaptive Neuro-Fuzzy Inference System

ANFIS are a class of adaptive networks that are functionally equivalent to FIS. It is used for solving nonlinear problems [57]. ANFIS is implemented based on Sugeno FIS. It allows the user to choose or modify the parameters of the membership functions according to the user input [57].The parameters are adjusted automatically by them with adaptive learning techniques like back propagation algorithm or hybrid method. The learning techniques allow the FIS to learn the information about the dataset. Sugeno FIS are computationally more efficient and accurate than Mamdani FIS as the complex defuzzification process of the Mamdani FIS is replaced with weighted average. It uses mathematical function of inputs as the rules consequent instead of fuzzy set employed in Mamdani FIS.

ANFIS is a hybrid method and uses the connectionist, parallel distributed processing model and learning ability feature of ANN. ANFIS consists of two parts. The first part is the antecedent part and the second part is the conclusion part, which are connected by rules. ANFIS network is feed-forward type which makes this adaptive network employ in wide variety of applications of modeling, decision-making, signal processing and control. ANFIS architecture comprises of 5 layers [64] as shown in Fig.6.1.

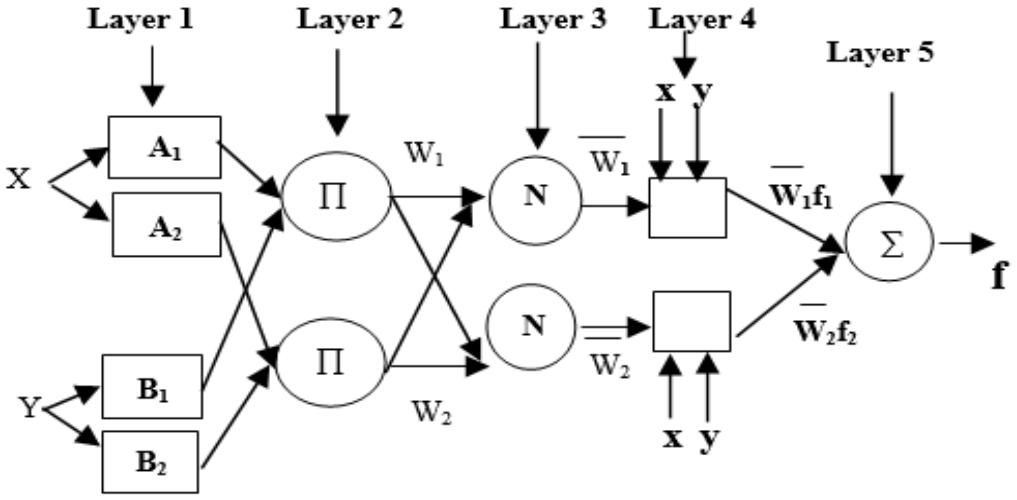

Fig.6.1. ANFIS architecture

A. Layer1

Each node in the first layer of ANFIS architecture processes the inputs. X and Yare the input values of the nodes and Ai and Bi represents fuzzy sets. Fuzzy set shave membership function. In our study, triangular MF is used. The output of the layer are defined by

$$O^1_i = \mu_{Ai}(X), \mu_{Bi}(Y) \qquad i = 1,2.....n \qquad (5.1)$$

B. Layer2

Aggregation of membership function is performed due to an appropriate t-norm applied in premise part. It is a layer of rules. The output of the first layer constitutes the input to this layer. W_i value is the multiplication of A_i node and that of B_i node. W_1, W_2 are the ignition strength of the rule [57]

$$O^2_i = W_i \; AND \; rule(O^1_i = \mu_{Ai}(X), \mu_{Bi}(Y)) \qquad (5.2)$$

C. Layer3

This is the layer of normalization. Each node in the layer calculates the ratio of i^{th} rules firing strength to the total of all firing strength.

$$O^3_i = W'_i = W_i / W_1 + W_2 + \ldots W_n \qquad (5.3)$$

D. Layer4

Each node in the fourth layer is an adaptive node which maps to the output membership function [57]

$$O^4_i = Y_i = W'_i f = W'_i (p_i X + q_i Y + r_i) \quad (5.4)$$

Where pi, qi, ri are consequent parameters

E. Layer5

There is a single node in this layer. The node sums the output values of each node in the layer 4.The summation is the output value of the ANFIS system.

ANFIS system performance is measured by Root Mean Squared Error (RMSE). RMSE is the square root of the sum of the squares of the difference between the actual system output and the model output. Our goal is to minimize the error. A total of 100 data were considered in which 60% of the data was used for training, 30% was used for testing and rest 10% was used for checking .FIS was generated by using Grid partitioning method. FIS was trained in 400 Epochs. Error tolerance was kept zero for the process. Grid partitioning models can run accurately with a less number of membership functions. Gridis only suitable for applications with small number of inputs variables (less than 6) [63] .A total of 5 inputs and one output were given to the system. The description of the input and their linguistic term is shown in Table 2.1.

Table 5.1 Description of input variables and linguistic terms

Input	***Linguistic Term***
Age	*Adult, Middle Aged, Elderly, Old Aged*
Body Mass Index	*Normal, Overweight, Obese, Extreme Obese*
Diastolic Blood Pressure(mm/Hg)	*Low, Normal, High, Very High*
Diabetes Pedigree Function	*Yes, No*
Plasma Glucose Concentration(mg/dl)	*Low, Intermediate , High , Very High*
Suffering Diabetes	*Yes, No*

The model was trained till the results were obtained with minimum error. Error tolerance was used as stopping criterion of training. The error tolerance was set to 0.The training stopped after training data error remained within this tolerance. For generating FIS structure we used triangular membership function and output membership function as linear

type. The input/output space was given to the ANFIS EDITOR to construct FIS whose membership function parameter was adjusted by using back propagation algorithm.

5.3.3 Artificial Neural Network using Levenberg Marquardt Back propagation Algorithm

Levenberg-Marquardt **(**LM) algorithm was considered to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated and the gradient can be computed as

$$H=J^TJ \qquad (5.5)$$

$$g=J^Te \qquad (5.6)$$

Where J is a Jacobian matrix, which contains first order derivatives of the network errors with respect to the weight and biases, e is a vector of network errors. The Jacobian matrix can be computed through a standard back propagation technique that is much less complex than the Hessian matrix. In order to make sure that the approximated Hessian matrix J^TJ is invertible, Levenberg–Marquardt algorithm introduces another approximation to Hessian matrix

$$H=J^TJ+\mu I \qquad (5.7)$$

Where μ is always positive, called combination coefficient, I is the identity matrix.

Update rule of Levenberg–Marquardt algorithm can be presented as

$$W_{k+1}=W_k-(J_k^TJ_k+\mu I)^{-1}J_ke_k \qquad (5.8)$$

Where k is the index of iterations, e is the training error and w is the weight vector.

If neuron j is in the first layer, all its inputs would be connected to the inputs of the network, otherwise, its inputs can be connected to outputs of other neurons or to networks inputs if connections across layers are allowed. The output node of neuron j is calculated using

$$Y_j=f_j(net)_j \qquad (5.9)$$

Where f j is the activation function of neuron j and net value net j is the sum of weighted input nodes of neuron j.

Derivative of net $_j$

$$\frac{\partial \text{netj}}{\partial \text{wj,i}} = \text{Yj,i} \tag{5.11}$$

Slope Sj of activation function

$$\text{Sj} = \frac{\partial \text{Yj}}{\partial \text{netj}} = \frac{\partial \text{fj(net j)}}{\partial \text{net j}} \tag{5.12}$$

$$O_m = F_{m,j}\left(Y_j\right) \tag{5.13}$$

Where $O_n$is the output of the network

For all patterns

% Forward computation

for all layers

for all neurons in the layer

calculate net; % Equation (5.10)
calculate output; % Equation (5.9)
calculate slope; % Equation (5.12)

end;
end;

% Backward computation

initial delta as slope;

for all outputs

calculate error;

for all layers for all neurons in the previous layer

for all neurons in the current layer

multiply delta through weights

sum the back propagated delta at proper nodes

end;

multiply delta by slope;

end;

end;

Fig.5.2 Pseudo code of forward computation and backward computation Implementing Levenberg– Marquardt algorithm.

A multilayer perception (MLP) is a feed forward artificial neural network model that maps a set of input data onto a set of appropriate outputs [77]. Since single neurons are not able to solve complex tasks so a generic model is developed which can adapt some training data. A multilayer feed-forward network is formed by the interconnection of several layers. The input layer receives the input and buffers the input signal. Output layer is responsible for generating the output of the network[56] . Layers that are formed between the input and output layer are called the hidden layers. Hidden layer is internal to the network and has no direct contact with the external environment [60]. As the number of hidden layer increases the complexity of the network also increases.

Feed-forward neural networks are widely used models for classification. A network is said to be a feed-forward network if no neuron in the output layer is an input to a node in the same layer or in the preceding layer. In this study a neural network toolbox of MATLAB 13 was used. The network consists of three layers:

I. Input Layer

II .One hidden layer

III. Output Layer

In case of diabetes prediction, one hidden layer with10 hidden layer neurons was created and trained. One hidden layer generally produces excellent results. Increasing the number of neurons in the hidden layer increases the power of the network, but requires more computation and is more likely to produce over fitting. The training function used was TRAINLM The default transfer function for hidden layers was TANSIG and the transfer function for the output layer was purelin.75 % of the data was used for training, 15 % was used for testing and 10% was used for validating the model. Training set is used to teach the network. The training process continued as long as the validation set improved [54]. The information moves in only in forward direction, from the input nodes to that of the output nodes through the hidden nodes as shown in Fig.5.3.

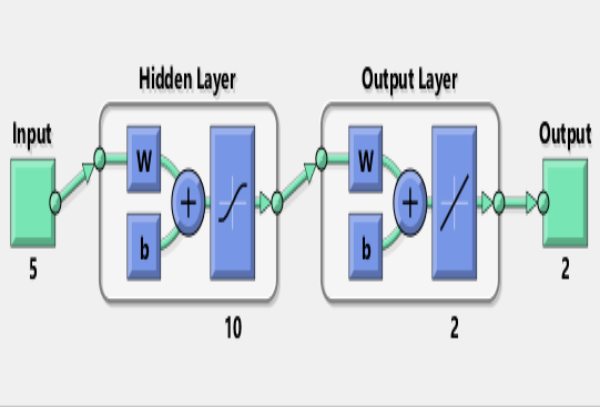

Fig. 5.3. Network model with different layers

Error calculation is performed in backward direction. This is why the network model is termed as feed-forward back propagation neural network model. The hidden layers learn the pattern in data during the training phase [64]. Each neuron in the hidden layer uses a transfer function to process data it receives from the input layer and then transfers the processed information to the output neuron for further processing using a transfer function in each neuron. Fig.6.4. shows back propagation neural network with one hidden layer.

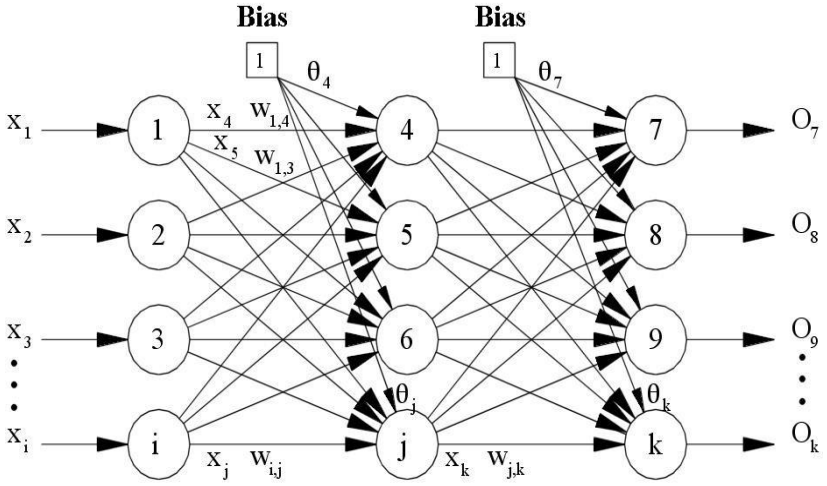

Fig.5.4. Back propagation Neural Network

The output of the hidden layer can be represented by equation (5.14), (5.15), and (5.16)

$$V_K = \sum_{j=0}^{m} W_{kj} X_j \qquad (5.14)$$

Error Calculation:

$$E_0(k) = Y_{d,0}(k) - Y_0(k) \qquad (5.15)$$

Final hidden layer output:

$$Yk = \Phi(Vk) \qquad (5.16)$$

Where,

Wki.....Wkm are the value of synaptic weight Wk0 is the value of bias.

E0 (k) is the value of error.

Yd,0(k) is the desired value of output Y0 is the actual value of output.

Vk is the output value before the process of activation function. Φ(.) is the activation function.

Yk is the final output for hidden layer.

The neural network is trained by TRAINLM that is Levenberg-Marquardt Algorithm (LMA) [61]. LM algorithm is most widely used optimization algorithm [77]. LMA generally provides faster convergence and better estimation than other training algorithm. Although the LMA converges very fast but it can cause the memorization effect when over training occurs. If neural network starts to memorize the training set, its generalization starts to decrease and its performance may not be improved for constrained sets [58]. It outperforms simple gradient descent and other conjugate gradient methods. LMA is also known as Damped Least Squares (DLS) method and is used to solve non-linear least squares problems. LMA finds the local minimum, which is not necessarily the global minimum.

5.4 Results and Discussion

The performance of the each method as discussed below.

5.4.1 Results of Adaptive Neuro-Fuzzy Inference System

TheANFISmodelcreatedatotalof512rulesasshownin Fig.6.5.Thenumberofmembership functions for the ANFIS affects the number of rules. Structure of the generated ANFIS is shown in Fig6.6.

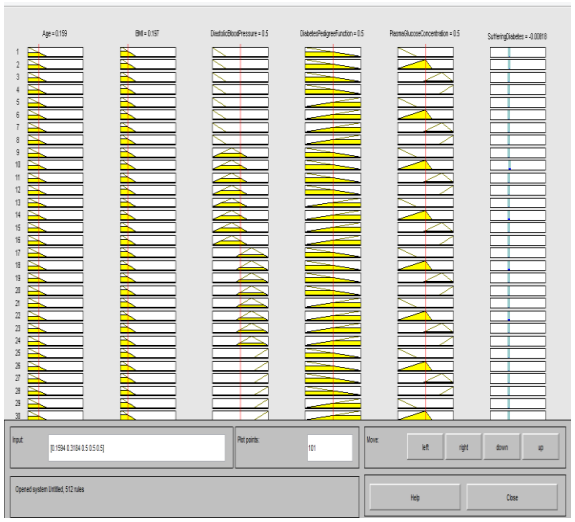

Fig. 5.5.Fuzzyruleviewer

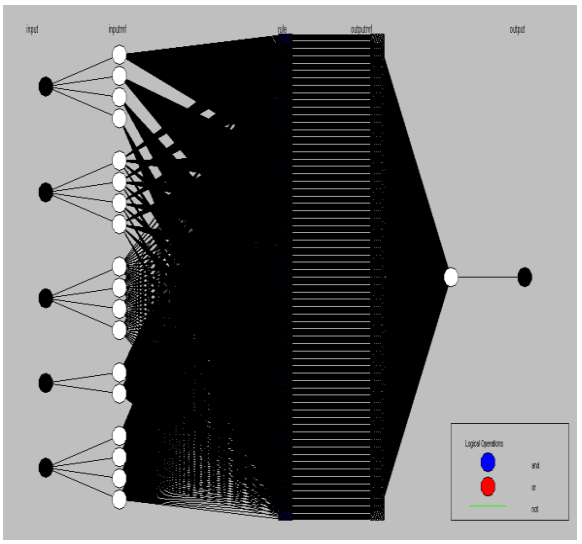

Fig. 5.6. ANFIS model structure

Training the ANFI system with the training data set is shown in Fig.6.7.Atrainingerror of 0.069932 was obtained. Testing the trained FIS is shown in Fig. 6.8. The average testing error for the training data set obtained was0.069817.

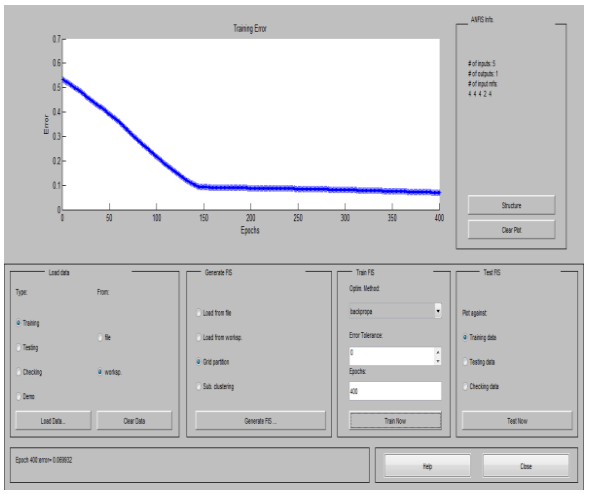

Fig. 5.7.Training error

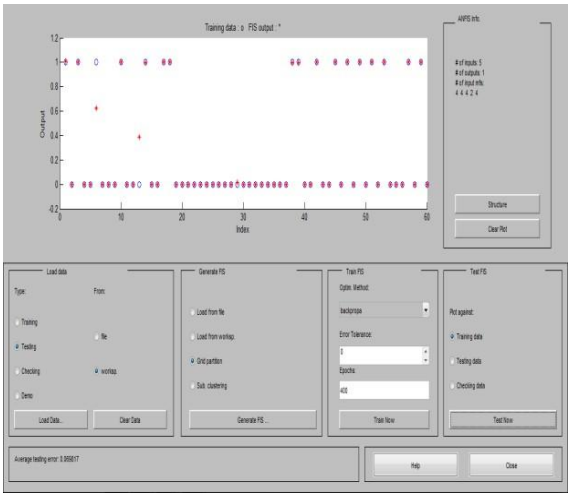

Fig. 5.8. Testing the FIS with training data set

The training error is the difference between the training data output value, and the output of the FIS corresponding to the same data input value [76]. The training error records the RMSE of the training data set at each epoch. Checking data is used for testing the generalization capability of the FIS. Checking error is difference between the checking data output value, and the output of FIS corresponding to the same checking data. Fig.6.9 shows checking data tested against the trained FIS.

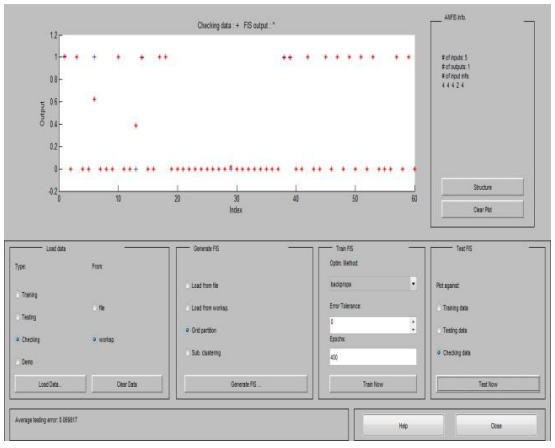

Fig. 5.9. Testing the checking data set

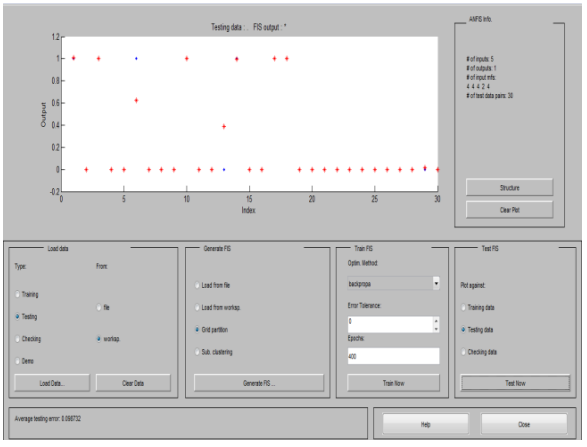

Fig.5.10. Testing of test data and training data

When checking data is tested against the trained FIS it showed an error of0.069817.Fig. 5.10 show the testing of the data set. The average testing error for testing data set obtained was 0.096732 which gave an accuracy of 90.32 %.Testing the dataset against trained FIS is

shown in Fig. 5.11.The overall performance of ANFIS is shown in Table 6.2 The training set gave an accuracy of 93.1 % while the test set gave an accuracy of 90.4 %.The checking set gave an accuracy of 93.1 %. Surface views of all possible values of the inputs to their corresponding output value are shown in Fig.5.12.

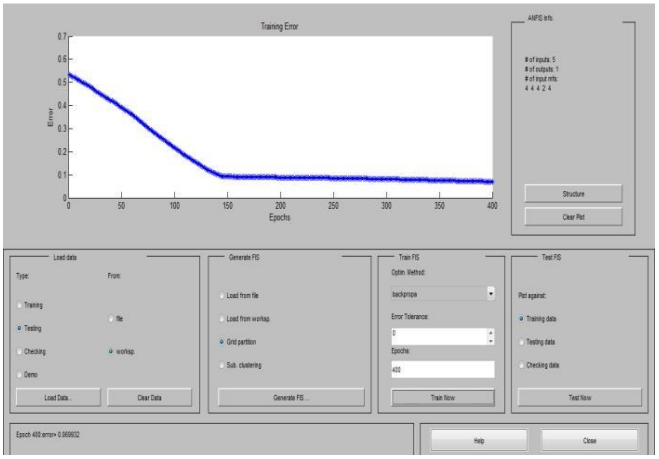

Fig. 5.11. Testing of training dataset

Table 5.2 Overall performance of ANFIS

Architecture	***Process***	***Sample***	***RMSE***	***MSE***
ANFIS	*Training set*	*60*	*0.069932*	*0.00480*
	Testing set	*30*	*0.096732*	*0.00935*
	Checking set	*10*	*0.069817*	*0.00487*

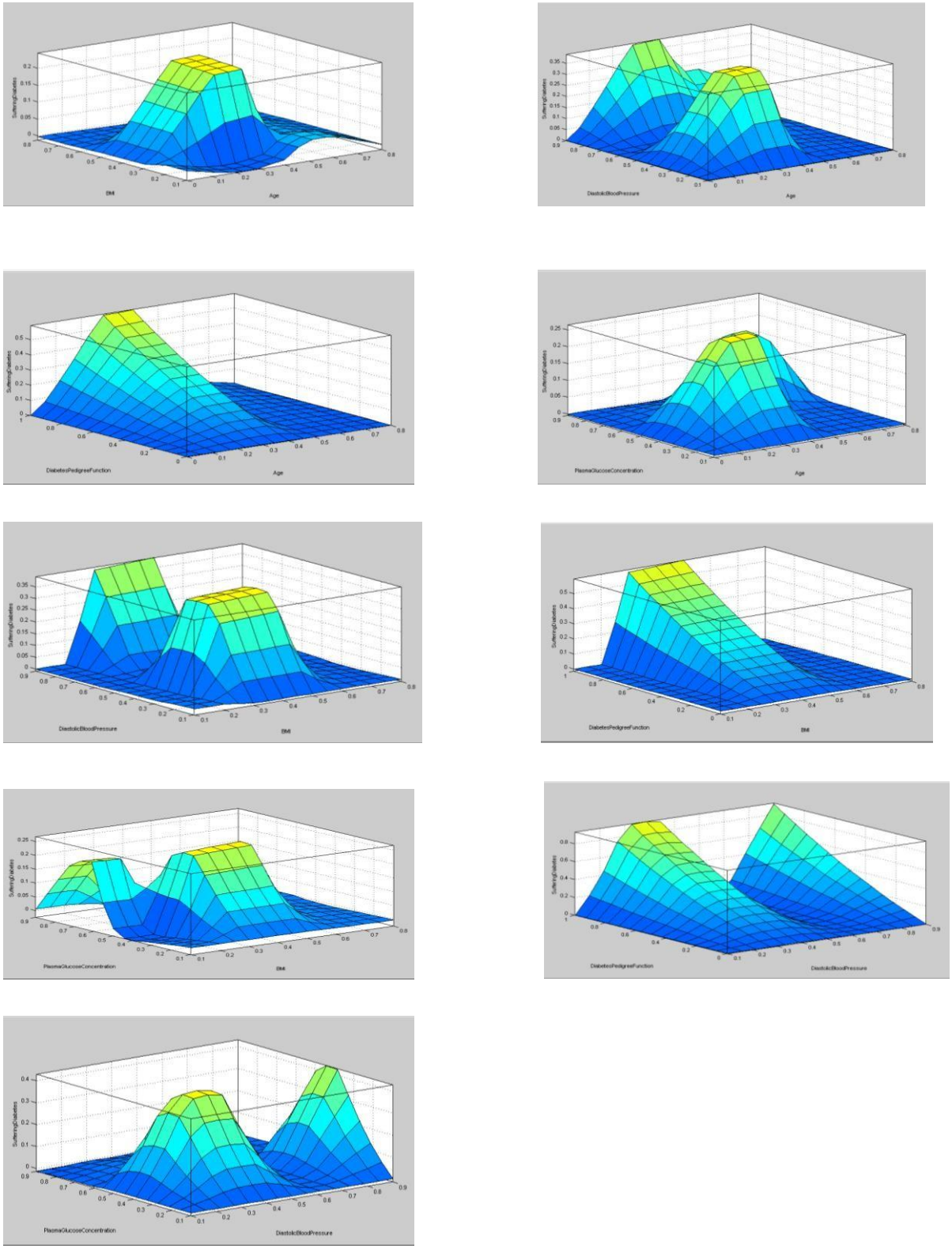

Fig. 5.12. Surface view of all possible values of the inputs to their corresponding output value

5.4.2 Results of Artificial Neural Network using Levenberg - Marquardt Back propagation Algorithm

Neural Network toolbox from MATLAB 13 was used to evaluate the performance of the proposed networks. Levenberg-Marquardt back propagation algorithm was used where the training automatically stopped when generalization stopped improving, as indicated by an increase in the Mean Square Error (MSE). MSE is the average squared difference between outputs and targets. Lower values are better while zero means no error. The results of applying the ANN methodology to distinguish between diabetic and non-diabetic people based upon the selected symptoms showed very good abilities of the network to learn the patterns corresponding to symptoms of the person. Fig.6.13 shows the best validation performance was 0.13902 at epoch 6. The MSE was equal to 0.0421 as shown in Fig 5.14.

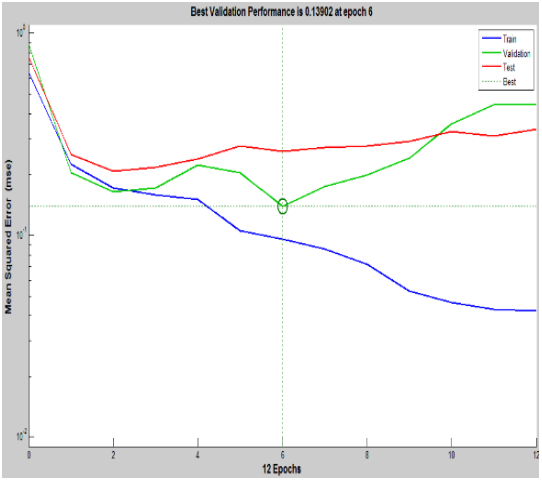

Fig. 5.13.Neural Network performance analysis

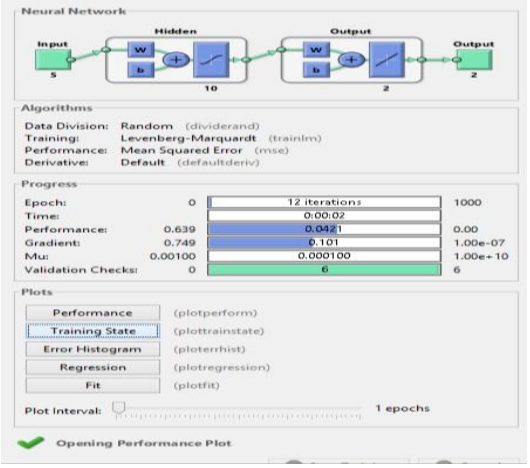

Fig. 5.14. Neural Network Training

Regression analysis was performed to investigate correlation between the desired and predicted results based on value of correlation coefficient, R. The perfect fit between the training data and the produced results was indicated by the value of R which is equal to1.Fig6.18 shows the regression plot where the overall value of R was 0.71083. This means that the network has an accuracy of 71.08 %. The training set gave an accuracy of 79.7 %. 28.2% accuracy was observed in testing phase. The validation set gave an accuracy of 66.6%. Overall performance of neural network is shown in table5.3.

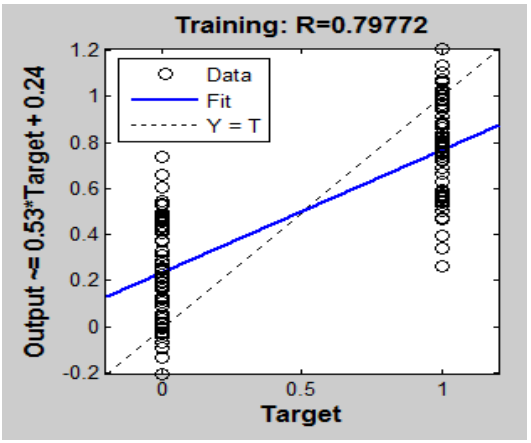

Fig.5.15.NeuralNetworkregressionfor Training data

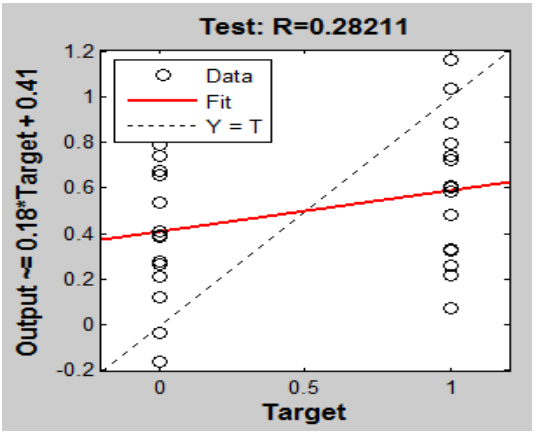

Fig.5.16.NeuralNetwork regression

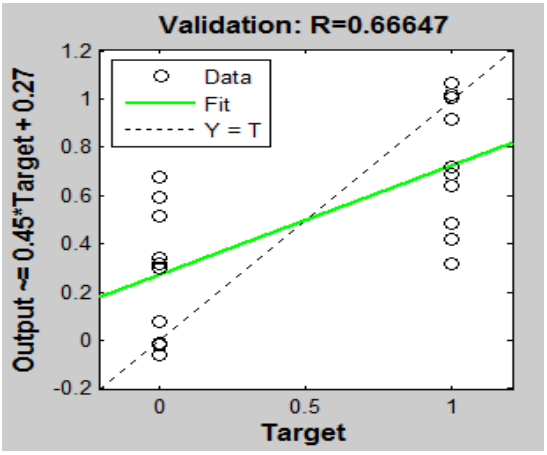

Fig. 5.17.Neural network regression for validation data

.

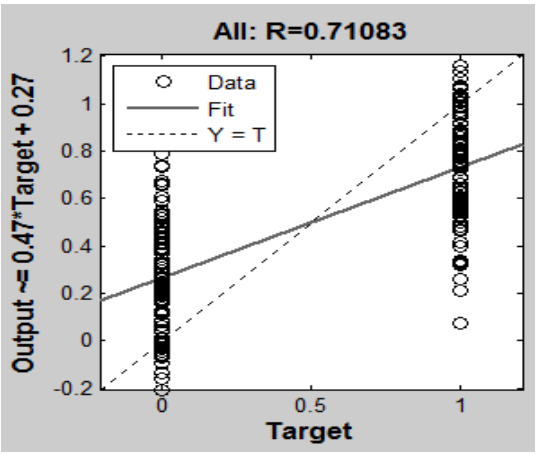

Fig.5.18. Neural Network regression for overall performance

Table 5.3 Overall performance of Neural Network

Architecture	***Process***	***Sample***	***RMSE***	***MSE***	***R***
Neural Network	*Training set*	*75*	*0.2051*	*0.0421*	*0.797*
	Testing set	*15*			*0.282*
	Validation set	*10*			*0.666*