Comparative Analysis of Intelligent Hybrid Systems for detection of PIMA Indian Diabetes

Rahul Kala
Department of Information Technology
Indian Institute of Information Technology and Management Gwalior
Gwalior, Madhya Pradesh, India
rahulkalaiiitm@yahoo.co.in

Anupam Shukla
Department of Information Technology
Indian Institute of Information Technology and Management Gwalior
Gwalior, Madhya Pradesh, India
dranupamshukla@gmail.com

Ritu Tiwari
Department of Information Technology
Indian Institute of Information Technology and Management Gwalior
Gwalior, Madhya Pradesh, India
rt_twr@yahoo.co.in

Citation: R. Kala, A. Shukla, R. Tiwari (2009) Comparative analysis of intelligent hybrid systems for detection of PIMA Indian diabetes, *Proceedings of the 2009 IEEE World Congress on Nature & Biologically Inspired Computing*, Coimbatote, India, pp 947 – 952.

Final Version Available At: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5393877

***Abstract*—The past few years have seen a lot of applications of Hybrid Soft Computing approaches that seem to have completely replaced the traditional uni-system approaches. The added abilities that come from the hybrid approaches motivate their use in every system. Bio-Medical Engineering is yet another field which has seen a major change in he past few years. We find various new approaches being applied to this field as well as many new models being proposed. At this juncture, we study the effectiveness of various new hybrid approaches in the field of Bio-medicals in this paper. PIMA Indian database has been used for this purpose from the UCI Machine Learning Repository. The basic aim is to compare the various hybrid approaches from the recent literature and compare their performances. We have chosen 3 major Hybrid Systems and standard Back Propagation Algorithm for this purpose. These are Adaptive Neuro Fuzzy Inference Systems, Ensembles and Evolutionary Artificial Neural Networks. We also try to explain the results from our theoretical understanding of the individual Hybrid Systems.**

Keywords-PIMA Indian Diabetes, Ensemble, ANFIS, Modular Neural Network, Evolutionary ANN, Classification, Bio-Medicals

I. INTRODUCTION

Bio-Medical Engineering is a rapidly growing field as a result of the need and rise of automation. This field calls for the collaboration between the people from the medical background and the engineers to develop intelligent systems for the various tasks in bio-medicals. These systems are used for the detection of the various diseases. These act as Clinical Decision Support Systems (CDSS) in order to assist the doctors in their task of identification of the presence of the diseases. They hence act as valuable tools for the doctors in the analysis of the diseases. This is especially important considering the work load over the doctors and the vast presence of the diseases. The increasing health consciousness among the people has further resulted in a lot of emphasis in the development and use of such systems.

Here we make the use of Soft Computing techniques for the detection of diseases. This is essentially a classification problem where the task is to classify the given parameters into either of the two classes that stand for the presence or absence of diseases. The classification problems have always deserved a special mention from the scientific communities as they have their own issues and complexities. One of the most interesting facts with these problems is that despite the revolution in the methods and means of classification, the artificial systems are still far behind the classification powers of humans when compared with accuracies. A related terminology is pattern matching that deals with the ability to recognize any given pattern or set of attributes.

The Hybrid Approaches form a very exciting field of work and research. A good collection of methods and applications can be found in the books [1, 2]. The increase in efficiency that they have brought to numerous problems in the various domains is a key reason behind their use and popularity. These systems make use of a collection of Artificial Intelligence and Soft Computing Systems for the purpose of problem solving. The benefits of one system make over for the limitations of the other systems. As a

result the whole system has an added performance. The base systems used for this purpose are Fuzzy Inference Systems (FIS), Artificial Neural Networks (ANN) and Evolutionary Algorithms (EA).

The ANNs [3] form good means of learning from the past data (or machine learning) and generalizing the learnt trends into the unknown inputs for the PIMA Indian Diabetes. These networks hence undergo two separate stages of training and testing. One of the chief ways of training of the ANNs is Back Propagation Algorithm (BPA). The algorithm however many a times gets trapped in local minima.

FIS [4] make use of Fuzzy Set theory for the modeling of the problem of PIMA Indian Diabetes. These systems are effectively able to map the inputs to the outputs by the applications of Fuzzy Rules. These systems however need to be manually trained and tuned which is not always feasible for all the problems.

The Evolutionary Algorithms form excellent algorithms for the purpose of search and optimization problems. These algorithms are very effective in searching for Global Minima in a finite and limited amount of time [5].

This paper is organized as follows. Section 2 deals with the various approaches used in this paper. In section 2(a) we deal with the first method of application i.e. ANFIS. We discuss about the Modular Neural Networks (MNNs) and Ensemble Techniques in Section 2(b). Section 2(c) is dedicated towards the use of a connectionist approach in Evolutionary Neural Networks. In section 3 we describe the database used along with the problem solving methodology. Section 4 gives the results. The conclusion remarks are given in section 5.

II. Approaches Used

In this section we describe the various approaches used in this paper for solving the problem of identification of the PIMA Indian Diabetes. The various approaches make extensive use of various Soft Computing techniques in a hybrid manner for the purpose of identification. Here we make use of 3 major techniques along with the standard Back Propagation Algorithm. These are ANFIS that is a combination of ANN with FIS, Modular Neural Network that make use of modularity in the problem and apply ANN for problem solving and Evolutionary ANN where the complete ANN is evolved with the use of GA.

A. ANFIS

Adaptive Neuro Fuzzy Inference Systems (ANFIS) [6, 7] is the first hybrid method employed in solving the problem. This method is a fusion of the ANN with FIS. Essentially ANFIS is a Fuzzy Inference System (FIS) that is made over the Neural Network Architecture. This enables us to use Neural Network training algorithms in the training of FIS over a historical database. This tunes FIS and makes it capable to act according to the demands of the problem and give expected results to problems whenever it happens to be exposed towards them. This significantly contributes towards the minimization of error and achieving great performance in solving the problem.

Much like the ANN, the ANFIS has a layered architecture where the various layers do some task of the FIS. The various layers are connected by connections that connect the various modules of the ANFIS. The various layers used in the ANFIS are input, fuzzification, AND, normalization, rule output and defuzzification. The general layered structure is given in Figure 1.

Problem solving in ANFIS consists of a series of steps. The first step is to make an initial Fuzzy model as per the requirements of the problem. After the model has been formed, we need to optimize it by a training algorithm. For this we make use of a historical database of known inputs and outputs. This is a supervised learning technique used by the ANFIS. Two commonly used learning algorithms are Back Propagation Algorithm and Hybrid Learning Algorithm. The testing involves giving the unknown inputs to the algorithm and then letting it classify the data as per the training performed. This step decides the efficiency of the algorithm.

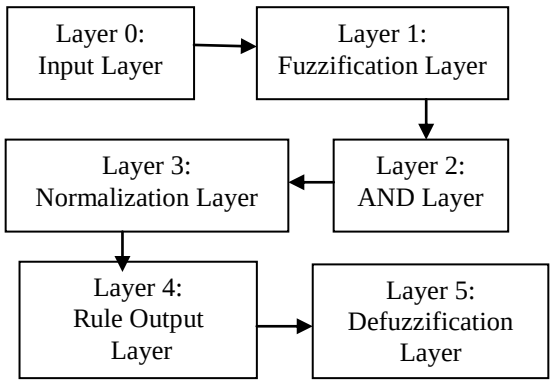

Figure 1: The ANFIS layered architecture

B. Ensemble

The ensemble is the second Hybrid method used for the identification of PIMA Indian database. The ensemble is a type of Modular Neural Network (MNN) [8, 9]. The MNN believes in exploiting the modularity in the problem by dividing the problem into a set of modules. The modules work independent of each other and contribute towards the problem solving.

The conventional ANN has problems that result in a loss of performance when the training database is too large in size or the problem being concerned is very complex. This inhibits their performance and even leads to very large training times. This problem in the MNNs is solved by using modularity by MNNs.

The first task of problem solving by MNNs is to break the problem into a set of modules. This breaking is done at the time of design of the system itself. Whenever the problem is given to the system, it is first divided among the various modules. Then all the modules independently solve the problem or their part of the problem. All the solutions are calculated independently by the various modules and the results are supplied to a central integrator. The integrator does the task of combining all the individual results of the

various modules and giving the final answer of the system to the supplied input.

One of the major tasks in the use of ensembles is the breaking up of the problem into a set of modules. The modules need to be cautiously divided keeping the performance maximum without compromising much with the generality of the problem. It may be easily observed that as we increase the number of modules, the problem starts becoming more and more localized in nature. We know that localization is not a desirable thing in any system as it hinders the algorithm giving correct results when exposed to unknown inputs. But this is desirable till the added benefits of training time and performance give a boost to the system.

Ensembles are a popularly used type of MNNs especially for such classificatory problems [10]. We would only be discussing the particular model being used for problem solving in this paper. Other approaches and techniques are not discussed.

The first step is training of the system. Each of the modules or the ANN is trained independently by all the training data present in the system. All the ANNs may be trained in parallel. Here the output of each ANN is the probability vector that denotes the presence or absence of each of the class. Say the system has n classes. Each ANN gives as output the probability vector $<p_1, p_2, p_3, \dots, p_n>$ where each p_i denotes the probability of occurrence of the class i. Here we measure the probabilities in the scale of -1 to 1 with 1 denoting the maximum probability and -1 denoting the least.

In the implemented approach whenever the input is given to the system, it distributes it to all the various modules present in the system. Each module is a separate independent ANN. The modules analyze the problem and work over a solution. They then return a probability vector containing the probability occurrences of each of the classes. Say if the classificatory problem had n classes to which the input could belong to , each ensemble returns the probability vector $<p_1, p_2, p_3, \dots, p_n>$ where each p_i denotes the probability of occurrence of the class i. Ideally only one out of all the numbers in this vector must be 1 and all others must have -1, but the same may not be the case due to the imperfection in the system.

All the various probability vectors are given to the central integrator as per the principles of the MNNs. The integrator receives all the probability vectors and does the task of deciding the final output of the system. For this the first task is to average the probabilities of the various probably vectors. This makes a final probability vector that combines the results of all the modules. This is a mechanism where the integrator collects the individual responses of the modules to the problem to get the average response. Then the integrator selects the class that gets the maximum response or has the maximum probability of occurrence. This class is returned as the final class. This concept is given in Figure 2.

Another classical method to solve the problem is by making use of polling or voting mechanism. Here each module returns the class which it believes is the class to which the input belongs. The integrator gets all the various classes by different modules and then carries a voting in between the various modules. The class that gets the highest votes is decaled as the final winner. This concept is given in Figure 3.

C. Evolutionary Artificial Neural Networks

The last hybrid approach that we use to solve the problem is Evolutionary Neural Networks [11, 12]. In this approach we try to train the ANN using GA. The BPA used for the training of ANN has many limitations that are solved by the use of GA. The ANN may get trapped in local minima. Also we have to specify the architecture at the start of the algorithm which may not be optimal in accordance with the problem.

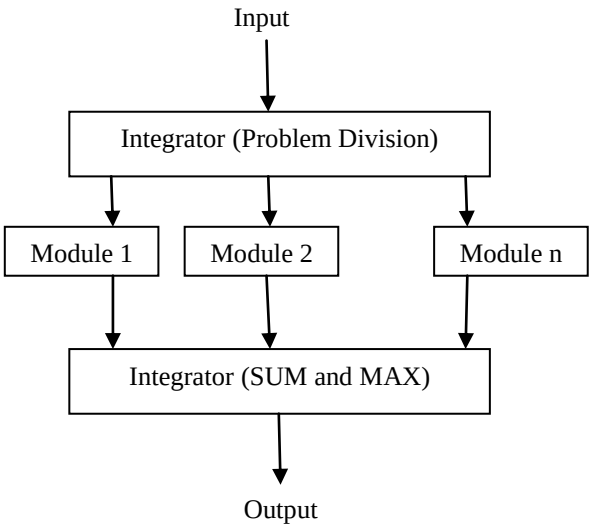

Figure 2: The ensemble using probability based approach

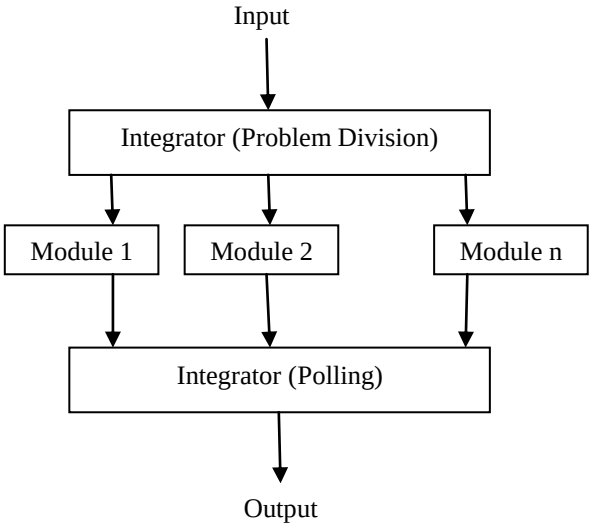

Figure 3: The ensemble using polling approach

The Evolutionary ANN technique tries to evolve and train the ANN by using the optimization problem of GA. It

may easily be seen that ANN training is essentially an optimization problem where the performance or the error function acts plays the role of objective function. These systems may be used only for the ANN training with fixed network architecture or for evolving the architecture as well. The former is a much simpler problem with a limited size of the search space as compared to the later which is more complex a problem with the search space spanning to a lot of dimensions.

Here we discuss the model of Evolutionary ANN that we follow for solving the problem of PIMA Indian diabetes. This is a connectionist approach. Here we tried to evolve the ANN weights as well as the correct connectionist architecture. The ANN that we generally take for problem solving in the problems is a fully connected model where every neuron of every layer is connected to all the neuron of the next layer. This architecture leads to a great demand of computation that increases the overall training time. Subtracting neuron may not be that vital as the network may not train for smaller number of neurons. If the network fails to train, the general approach is to add a neuron. But imagine the immense increase in dimensionality as a result of this addition of a single neuron. This expands the dimensionality by a big amount. This may make it very difficult for the training algorithm to train the network as a result of the same.

Hence we make use of the concept of a connectionist approach which tries to inhibit or stop certain connections to limit the computation time as well as overall complexity of the problem. This makes the calculations very fast and even the further training can take place easily as the connections are limited. In this problem we use GA to evolve connectionist architecture as well as to fix the ANN weights. The maximum number of neurons however needs to be defined at start.

The first section of the chromosome consists of a 0 or 1 depending upon the existence of connection between the input layer and the hidden layer neurons. A 1 depicts the existence of connection and a 0 depicts an absence of connection. Similarly the next section depicts the existence if weight between a hidden layer neuron and an output layer neuron. The bits have their same meaning. The other half of the chromosome contains the actual values of weights between the various connections. Here the first section of the second half contains weights between input and hidden layer neurons and the second section between hidden and output layer neurons. These are the actual values of connections that hold if the connection exists or not. If the physical connection is not there, the corresponding weight value may be ignored. The last part contains the values of the biases. The general structure of the chromosome is shown in Figure 4.

The crossover and mutation were modified to match the requirement of the problem. The crossover used was a 2-point crossover which was applied in such a way that the 1st point always lay in the sections that constitute the architecture and consisted of the bits. The second part always lay on the section containing the values of weights and bias. These were the real values. Similarly the mutation was modified to ensure that the section reserved for bits always contains only one of the 2 valid inputs i.e. 0 and 1.

Connection between I/P and Hidden Layer	Connection between Hidden and O/P Layer	Weights between I/P and Hidden Layer	Weights between Hidden and O/P Layer	Biases
Connections (0 or 1)		Weights and Biases		

Figure 4: The chromosome representation

In order to determine the extreme values or ranges of weights and bias, we see some trained ANNs of fully connected types and then the extreme values of the weights and bias that they take. The selected range is one that comfortably covers the observed range at the same time not resulting in the search space or the configuration space from becoming very vast in nature.

III. METHODOLOGY

The aim of the system is to solve the problem of detection of PIMA Indian Diabetes. For this we make use of the database of UCI Machine Learning Repository [13]. The PIMA Indian Diabetes data set consists of a total of 8 attributes. These decide the presence of diabetes in a person. This database places several constraints on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The first attribute is the number of times the women was pregnant. The next attribute is Plasma glucose concentration a 2 hours in an oral glucose tolerance test. We further have the attributes Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/(height in m)^2), Diabetes pedigree function and Age (years).

The whole methodology may be divided into two parts. These are training and testing. The database contains a total of 768 instances of data. 535 (~70%) instances of data of the data were used for the training purposes and the rest 233 (~30%) of the data was used for the testing purposes.

In the training phase the various hybrid systems are given the historical database and trained according to the individual training method. The data used is the training database. The next stage involved is testing. Here the system is given unknown data from the testing data set. The system output is collected and compared with the standard output. This helps us in the determination of the performance of the system.

The general methodology [14, 15] may be summarized by Figure 5.

IV. RESULTS

Here we present and compare the results of the various algorithms and hybrid systems that we use. These are given in the following sub-sections.

A. ANN with BPA

The first method used for this was ANN with BPA. Here we used a single hidden layer which consisted of 12 neurons. The activation functions for the hidden layer was tansig and purelin. The training function used was traingd. The other parameters were a learning rate of 0.05 and a goal of 10-1. Training was done till 2000 epochs.

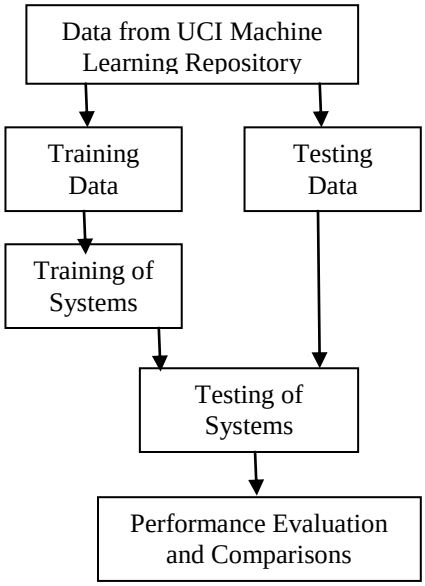

Figure 5: The general working methodology

After the network was trained and tested, the performance of the system was found out to be 77.336% for the training data set and 77.7358% for the testing data set. The training curve of the ANN is given in Figure 6.

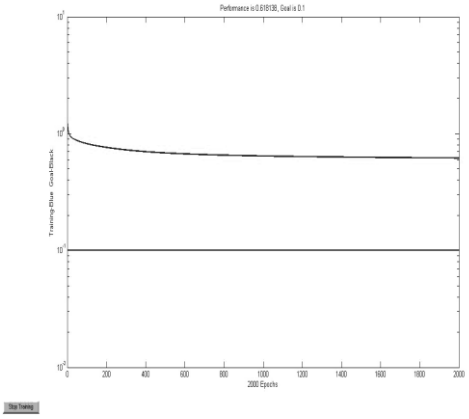

Figure 6: The training curve for ANN with BPA

B. Ensemble

The second experiment was done on ensembles. Here we had used 4 modules or ANNs. Each one of them was trained separately using the same training data set. The 4 ANNs were more or less similar with small changes to the ANN used in the previous section. The first was exactly the same as discussed above. The other ANNs only had changes made in the number of neurons in the hidden layer and the number of epochs. These had 14, 10 and 12 neurons respectively. The numbers of epochs were 2500, 200 and 4000. The four ANNs were trained separately.

Here we had used a probabilistic polling in place of the normal poling. The resulting system had a total performance of 78.7276% for the training data and 76.9811% for the testing data. It may be noted that the performance was 78.33% for the training data and 76.2264% for the testing data in the use of conventional ensemble where each module votes for some class.

C. ANFIS

The next experiment was done using ANFIS as a classifier. Here we used the same training as well as testing data sets. The FIS was generated using a grid partitioning method. Each of the attributes had 2 MFs with it. The system was allowed to be trained for a total of 100 epochs. The resulting training graph is shown in Figure 7.

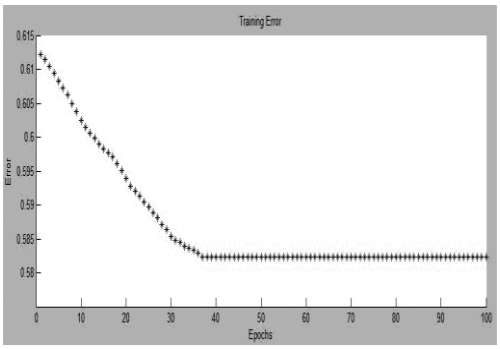

Figure 7: Training curve for ANFIS

The final system so obtained had a performance of 88.9720% for the training data and 66.5236% for the testing data.

D. Evolutionary ANN

The last hybrid system that was applied to the same problem was evolutionary ANN. Here we tried to evolve the ANN weights as well as the correct connectionist architecture. The connectionist approach is something that we had not discussed clearly earlier.

The parameters of the GA were a maximum number of 25 neurons, 25 as the population size with an elite count of 2. The creation function was uniform and double vector representation was chosen. Rank based fitness selection was used. Stochastic Uniform selection method was used. Crossover ratio was 0.8. The algorithm was run for 75 generations. The training curve is shown in Figure 8.

The final system had a performance of 77.38% for the training data set and 73.819% in the testing data set.

V. CONCLUSIONS

In this method we saw the working of four different Hybrid methods for the problem of detection of PIMA Indian diabetes. In all the cases we were able to solve the problem with fine accuracies using the hybrid approaches.

The first experiment done was of the use of ANN with BPA. These systems used the generalizing capabilities of the ANN for problem solving. We were able to get sufficient accuracies using these systems.

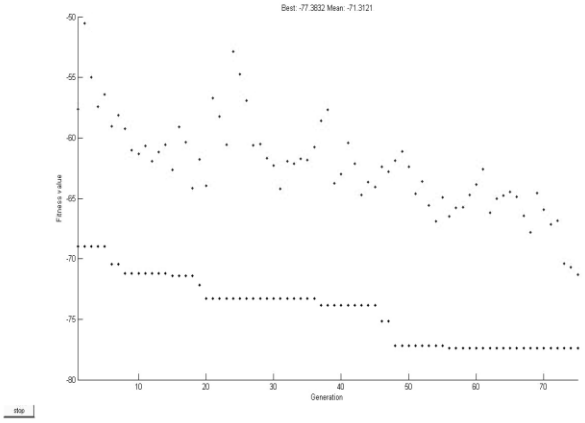

Figure 8: Evolutionary ANN training curve

The next implementation was using Ensembles. These systems removed the shortcomings of ANN with BPA. This resulted in a rise in accuracy of the systems. Next we applied ANFIS to the same problem. The results reveal that the system performed very well on the training data set but did not do that well on the testing data set. This may be attributed to the failure of the system to generate rules that could be generalized over the network. At the training phase, the system must have adjusted the MFs so as to meet the requirements, but the rules so generated failed to generalize. This may be even because of the selection of the wrong data that essentially belonged to some parts of the network.

The last approach was using the Evolutionary ANNs. These systems gave good performances where they could decide their own architecture besides the values for the weights and biases.

This paper reveals the functioning of the various hybrid approaches in problem solving and disease detection. We saw that all the methods were competitive and hence any generalization cannot be made regarding the effectiveness of a method over the other which is well supported by literature which does not lay any preference to any one of the hybrid approach for problem solving. The only exception was in the use of ANFIS which could not form generalized rules. It may be seen that the problem solution depends also upon the division of data between the training and testing data. Besides, there is a lot of dependence on the choice and measurement of attributes and the diseases at large. While some methods may look better for the problem of PIMA Indian diabetes, it cannot be guaranteed that the results would observe same behavior for other diseases. This again necessitates on the knowledge of both theoretical and practical aspects of the various hybrid methods and an iterative design approach to get a good soft computing system for problem solving.

REFERENCES

[1] Patricia Melin, Oscar Castillo, "Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing", Springer, 2005

[2] H Bunke, A Kandel (Eds), "Hybrid Methods in Pattern Recognition", World Scientific, May 2002

[3] Antonio Ciampi, Fulin Zhang, "A new approach to training back-propagation artificial neural networks: empirical evaluation on ten data sets from clinical studies", Statistics in Medicine, 21:1309-1330, 2002.

[4] Chang Xiaoguang and J.H Lilly, "Evolutionary design of a fuzzy classifier from data", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, pp 1894-1906, Aug 2004

[5] K.G. Srinivasaa, K.R. Venugopala, L.M. Patnaikb, "A self-adaptive migration model genetic algorithm for data mining applications", Information Sciences, Volume 177, Issue 20, pp 4295-4313, October 2007

[6] A. Vosoulipour, M. Teshnehlab, H. A. Moghadam, "Classification on Diabetes Mellitus Data-set Based-on Artificial Neural Networks and ANFIS", 4th Kuala Lumpur International Conference on Biomedical Engineering 2008, IFMBE Proceedings, pp 27-30, 2008

[7] Hasan Temurtas, Nejat Yumusak and Feyzullah Temurtas, "A comparative study on diabetes disease diagnosis using neural networks", Expert Systems with Applications, Volume 36, Issue 4, pp 8610-8615, May 2009,

[8] Fariba Shadabi Dharmendra Sharma, Robert Cox, "Learning from Ensembles: Using Artificial Neural Network Ensemble for Medical Outcomes Prediction", Innovations in Information Technology, 2006, pp 1-5, 2006

[9] Anupam Shukla, Ritu Tiwari, Hemant Kumar Meena, Rahul Kala, "Speaker Identification using Wavelet Analysis and Modular Neural Networks", Journal of Acoustic Society of India (JASI)

[10] Rahul Kala, Anupam Shukla, Ritu Tiwari, Fuzzy Neuro Systems for Machine Learning for Large Data Sets, Proceedings of the IEEE International Advance Computing Conference, ieeexplore, pp 541-545, DOI 10.1109/IADCC.2009.4809069, 6-7 March 2009, Patiala, India

[11] S. He, Q.H. Wu, J.R. Saunders, "A Group Search Optimizer for Neural Network Training", Computational Science and Its Applications - ICCSA 2006, pp 934-943, 2006

[12] Marijke F. Augusteijn, Thomas P. Harrington, "Evolving transfer functions for artificial neural networks", Neural Computing & Applications, Volume 13, Issue 1, pp 38-46, 2004

[13] Vincent Sigillito, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html], The Johns Hopkins University. 1990, Available At: http://archive.ics.uci.edu/datasets/Pima+Indians+Diabetes

[14] Anupam Shukla, Ritu Tiwari & Prabhdeep Kaur, "Diagnosis of Epilepsy disorders using Artificial Neural Networks", Proceedings of the Sixth International Symposium on Neural Networks, Book Chapter in Springer's Verlag book series on Advances in Intelligent and Soft Computing, Volume 56/2009, pp. 807-815. Springer Berlin / Heidelberg

[15] Anupam Shukla, Ritu Tiwari, Prabhdeep Kaur, "Knowledge Based Approach for Diagnosis of Breast Cancer", Proceedings IEEE

International Advance Computing Conference (IACC), March 6-7, 2009, Patiala, India. pp. 06-12, 2009.