Medical Diagnosis using Incremental Evolution of Neural Network

Rahul Kala[1], Harsh Vazirani[2], Anupam Shukla[3] and Ritu Tiwari[4]

[1, 2, 3, 4]Soft Computing and Expert System Laboratory,
Indian Institute of Information Technology and Management Gwalior
Morena Link Road, Gwalior, Madhya Pradesh-474010, INDIA
[1]rahulkalaiiitm@yahoo.co.in, [2]harshiiitmg@gmail.com, [3]dranupamshukla@gmail.com, [4]rt_twr@yahoo.co.in

Citation: R. Kala, H. Vazirani, A. Shukla, R. Tiwari (2010) Medical Diagnosis using Incremental Evolution of Neural Network, *Journal of Hybrid Computing Research* 3(1), 9-17.

***ABSTRACT*:** *There is an increasing concern for the rapid growth of the diseases, which necessitates the need of automated systems for disease diagnosis. The automatic diagnosis of disease not only helps in early detection of the disease and taking of necessary preventive measures, it also helps in precise diagnosis that may be of a lot of help to the medical practitioner. The use of neural networks in disease detection is common. The neural networks try to learn from the historical data in the training phase. This learning is generalized in the testing phase to new data. The major problem in neural networks is their fixed architecture which poses a problem for the network designer to find out the optimal architecture. This leads to the use of automatic generation of neural networks. In this paper we propose the use of an incremental evolution of multi-layer perceptron for the diagnosis of PIMA Indian Diabetes. Here the network starts with a small number of hidden layer neurons, which increase with time. Evolutionary algorithm is used for the optimization. The evolution uses Simulated Annealing as a local search strategy. The resulting diagnosis using this strategy showed a better performance as compared to a number of classical neural approaches. This hence makes an excellent diagnostic system.*

***KEYWORDS*:** Medical Expert System, PIMA Indian Diabetes, Bio-medical engineering, Evolutionary Neural Networks, Simulated Annealing.

I. INTRODUCTION

Automatic medical diagnosis is an important problem that deals with the use of technology for the diagnosis of various diseases. The automated diagnostic systems find a lot of application in assisting the doctors for carrying out quick and effective diagnosis [1]. The diagnostic systems make use of the past data of the patients to carry out the diagnosis. These systems analyze and monitor the trends in the historical occurrences or non-occurrences of these diseases. These trends are later generalized to the new data as well [2]. This result in the engineering of intelligent expert systems that carry out the task of diagnosis as domain experts based on the inputs provided [3]. The medical expert systems make use of a historical database that is a collection of past trends, knowledge base based on which the decision of the presence of disease is made, inference engine etc.

Diabetes is an emerging problem in the medical world. The disease may be classified into three types, namely type 1, type 2, and gestational diabetes that occur during pregnancy. Each of these types has its own symptoms and is found is different population distributions [4]. Type 2 diabetes is increasing throughout border area, along with risk factors for the disease. Some 1.1 million border residents 18 and older suffer from type 2 diabetes, and 836,000 are pre-diabetic as per 2007 PAHO survey [5]. It was estimated that more than 2.5 million people worldwide experience vision loss due to diabetic retinopathy [6]. According to the survey by American Diabetes Association (ADA), 49 percent of the U.S. adults polled said they most feared cancer as a potential health problem while just 3 percent said they worried about diabetes. The survey results suggest that people need to assess their diabetes risk and take it more seriously [7].

The field of biomedical-engineering relies upon the use of machine learning for the decision making. Machine learning deals with the analysis of the historical database having instances of occurrences and non-occurrences of the disease. Based on this database we try to analyze the trends in the historical database. These trends constitute the knowledge base of the system. A good learning technique results in the extraction of good rules that may easily be generalized to the testing data as well. The system hence gives the correct output to any applied input, provided it obeys the trends of the historical database.

The problem of medical diagnosis is essentially classificatory in nature. The two classes of the system denote the presence and absence of the disease [8, 9]. Any system is supposed to classify the inputs into either of the two classes. The classificatory problems behave best when the inter-class separation is high and the intra-class separation is low. In such a case it is reasonably easy for the system to construct decision boundaries that separate the classes. However for many problems the attributes are not ideal enough to showcase an easy separation of the classes by decision boundaries. This emphasizes on the need of good machine learning technique to construct flexible decision boundaries. The effect of noise especially adds a problem. The noisy data may many times result in befooling the system and trying to drive it to the wrong direction. It further may result in poor generalization of the resulting system.

Neural networks find ample of applications in medical diagnosis. They are easily able to learn from the historical data and generalize their learning to the testing data. A number of neural network models have been applied to the diagnosis of numerous kinds of diseases. This includes the curve fitting neural network models like Multi-Layer Percepton, Radial basis Function Networks, etc.; the classificatory models of neural networks like Self-Organizing Maps, Learning Vector Quantization, etc.; or the recurrent neural network models [10-13]. The common diseases for which expert systems are engineered include skin diseases, heart diseases, thyroid disorders, etc. All the neural models are able to solve the various diseases by varying magnitudes.

One of the biggest disadvantages of the neural approaches is that the network architecture needs to be fixed by a human expert. The various training parameters also need to be specified. The performance of the network largely depends upon these parameters. The human fixing of the parameters may many times result in sub-optimal parameter fixing. Further the training algorithm may result in the network converging to local minima. These problems are solved by the evolutionary neural networks that use the evolutionary algorithms to evolve the most optimal neural network. The evolution of neural network is a very time consuming activity. This may hence require the assistance of a heuristic technique to act as a local search strategy in the evolution process. This results in every individual readily getting to the most optimal point in the vicinity.

In this paper we propose an incremental evolution of the neural network. The algorithm uses Multi-Layer Perceptron as the neural network model. An evolutionary algorithm facilitates the neural network training. The maximum number of permissible neurons in the neural network increase along with evolutionary algorithm generations. The algorithm uses simulated annealing as the local search strategy.

This paper is organized as follows. Section 2 presents some of the related works. The algorithm design and working is presented in section 3. Section 4 presents the simulation results and the comparison of the algorithm to the other conventionally used algorithms. The conclusion remarks are presented in section 5.

II. RELATED WORK

In this section we briefly overview the advances in the domain of biomedical expert systems as well as classification. Numerous models on medical diagnosis have been developed and tested by Shukla et al. [10-13]. These models use a variety of methods namely Multi-Layer Perceptron with Back Propagation Algorithm, Radial Basis Function Networks, Self Organizing Maps, Learning Vector Quantization, Adaptive Neuro Fuzzy Inference Systems, etc. for the diagnosis of diseases. The major diseases include diabetes, heart diseases, epilepsy, breast cancer, thyroid, etc. In all combinations of model and disease, an effective diagnosis could be made. This emphasizes on a high degree of accuracies of the individual systems. There is however always a scope to remove the individual limitations of the models and further enhance the recognition score.

A number of models from the hybrid soft computing have been applied on the problem of PIMA Indian Diabetes by Kala et al. [14]. This includes the ensemble approach, neuro-fuzzy system, and evolutionary neural networks. All the hybrid methods gave a good accuracy for diagnosis. Based on the comparisons in the same work it was clear that the evolutionary neural networks and ensemble techniques remove the limitations existing in the individual neural network models. An extended version of these works may be found in [15, 16].

The problem of classification especially requires a good system modeling in order to enable the system separate the various classes in the system. The major problem is especially the classification of the inputs that lie close to the decision boundaries. Kala et al made an implementation of a Modular Neural Network for machine learning [17]. This model clustered the entire input space into clusters. Each cluster was solved using its own neural network. The approach was applied for learning of a self made database of face recognition. Results proved that the approach could better identify the faces. Further the system was scalable to handle much more data. In another approach [18] an ensemble approach is used for problem solving. Here a variety of models were used for solving the same set of inputs and outputs. The integrator used a voting mechanism for deciding the final output. This approach was on a speech database. The combination of face and speech was applied along with a better integration technique in [19]. Here each module returned the probabilities of the occurrence of the various classes. These were summed up for all the modules to get the final probability vector. The integrator declared the class corresponding to the maximum sum as the final output class.

One good modular neural network model is presented for the biometric recognition in [20, 21]. Here the authors make three different modules for a multi-modal biometric recognition system. One module is dedicated to each biometric modality i.e. face, speech and fingerprint. Fuzzy integration is the integration technique of use. Each module in turn uses a hierarchical Modular Neural network with an evolutionary base and a fuzzy integration technique. In another work [22] co-evolution is used as a mechanism of evolution of a modular neural network. In this model the various modules of the modular neural network evolve in a co-evolutionary approach. The various modules help each other to evolve with good recognition rate and develop distinct characteristics for an optimal overall system performance.

III. ALGORITHM FRAMEWORK

The general algorithm is an evolutionary neural network. The neural network model used is Multi-Layer Perceptron. This model consists of a number of neurons arranged in a layered architecture. The first layer is a passive layer called as the input layer. The last layer is the output layer, where the outputs are collected. There may be a number of hidden layers in between. The use of multiple hidden layers complicates the problem with the construction of rapidly changing decision boundaries. The resulting network gives a reasonable performance in the training database, but the performance is very poor in the testing database due to low generalization [2]. Most of the bio-medical diagnosis problems can be effectively solved by a single hidden layer. We therefore assume that the neural network consists of a single hidden layer. Further the activation functions of the hidden layer and output layer are assumed to be constant that do not change in the evolutionary process. We assume the hidden layer to have an activation function of *tansig* and the output layer to have an activation function of *purelin*. This is shown in figure 1.

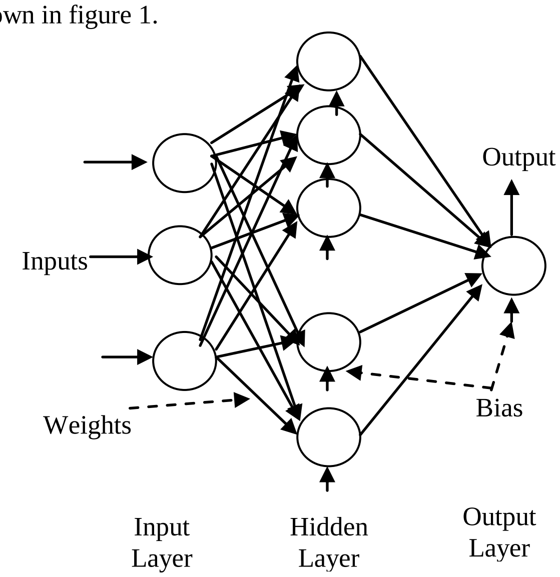

Figure 1 General Architecture of Multi-Layer Perceptron Neural Network

Each neuron here does the weighted sum of the inputs with addition of bias as given in equation (1) and then passes the resultant sum for application of the activation function as given in equation (2). This becomes the output of the neuron that may be given to the other neurons as per the network architecture.

$$y = \sum_i w_i x_i + b \tag{1}$$

$$o = f(y) = f(\sum_i w_i x_i + b) \tag{2}$$

The evolutionary approach here does the task of fixing of the number of neurons in the hidden layer as well as the various weights and biases.

The evolutionary algorithm used here follows an incremental evolution approach. An evolutionary approach always creates neural networks of varying sizes. This depends upon the evolutionary operators as well as the system specifications. In this algorithm we restrict the maximum number of neurons in the hidden layer to $n_{max}(t)$. Any network cannot have more than this number of neurons. This criterion is explicitly checked into all the evolutionary operators. This number is kept variable and changes as the algorithm proceeds. Initially the maximum permissible neurons $n_{max}(t)$ is kept to a very low value. As the generations increase, this number gradually increases in a linear manner. Hence the initial few generations witness very small networks. As the generations increase, the networks start getting complex in nature. The increase in $n_{max}(t)$ is given by equation (3).

$$n_{\max}(t) = n_{\max}(0) + \frac{n_{\max}(G) - n_{\max}(0)}{G} t \tag{3}$$

Here G is the maximum possible generations.

The complete evolutionary algorithm is shown in figure 2. The algorithm first generates random neural networks. It is ensured that each of the initial neural networks have neurons less than $n_{max}(0)$. This becomes the initial population for the evolution. The evolution takes place until the maximum generations are not reached. At each generation the first task is to generate the population of the next generation. This is done by the application of the evolutionary operators on the previous generation population. The various evolutionary operators are discussed in the next sub-sections. The factor $n_{max}(t)$ increases at every generation and needs to be calculated. Simulated Annealing serves as a local search strategy for the individual to hunt for the best place in the vicinity. This greatly relives the complexity of the evolutionary process. Each individual of the evolutionary process is a multi-layer perceptron. The conventional simulation is done for all the items of the training data. This determines the performance of the neural network or the fitness of the individual. The larger networks are given a penalty. The various steps are discussed one by one.

A. Evolutionary Operators

The task of generation of next generation population from the previous generation is done by the application of the evolutionary operators. The various operators used in this algorithm include (i) selection, (ii) mutation, (iii) crossover, (iv) elite, (v) jump and (vi) new.

Selection operator does the task of selection of the individual for the evolutionary process. The selection is more likely to select the fitter individuals than the weaker individuals. The algorithm uses stochastic uniform selection mechanism. The selection uses a rank based scaling mechanism. This scaling saves the algorithm from being over dominated by fitter individuals. Mutation operator tries to add new characteristics into the individual. This operator adds new characteristics to the system by slight modification of the existing individuals. The algorithm uses a Gaussian mutation technique. All the weights and biases are modified randomly by an amount that depends upon Gaussian random number.

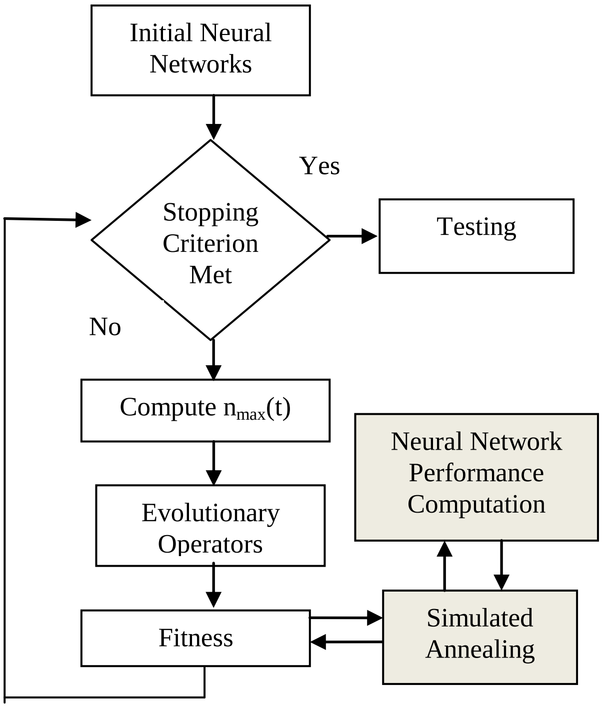

Figure 2 General Structure of the Algorithm

The crossover operation mixes two individuals or neural networks to produce the children neural networks. The problem in this problem with the crossover is of variable sizes of the two networks. Let the two parents have sizes n_1 and n_2. The crossover operator creates children that have sizes *ceil*$((n_1+n_2)/2)$ and *floor*$((n_1+n_2)/2)$. The distribution of the weights and biases from the parents to the children takes place in a random manner where any of the weights or bias to any child may randomly come from either of the parents. This is the scattered crossover technique. The other evolutionary operator is elite. This operator simply passes the best individual of one generation directly into the higher generation.

The next evolutionary operator used is jump. This operator modifies the structure of the neural network individual. It adds a new hidden layer neuron to the individual. The corresponding weights and bias are given random values. Similar operator is the new evolutionary operator. This operator adds completely random individuals with the maximum allowable size $n_{max}(t)$ to the population pool. The philosophy of incremental evolution is used here where the individuals are expected to grow into complex shapes with time. These operators hence try to advance neural networks to attain larger shapes along with time. It may be noted that the smaller neural networks are preferable as they lead to better generalization. Smaller networks hence have greater chance of survival in the evolutionary process by being selected in the elite, selection, and crossover operations. This makes the evolutionary process bias towards smaller networks and the smaller networks start dominating the population pool. It is equally important to same larger networks from extinction as well as to add an exploratory nature to the algorithm. The addition of neurons may have a large impact on the performance. Addition of a few neurons may result in better learning and hence better recognition. This exploration for higher performance by larger networks is caused by these two operators.

B. Simulated Annealing

The neural networks may have a variety of architecture. Each of these architectures is a collection of a large number of weights and biases. This makes the search space for the evolutionary algorithm very large in size. Besides the search space has a very complex architecture. Evolutionary algorithms face problems in such complex fitness landscapes. The evolutionary process hence needs to be assisted by a local search strategy. The local search strategy helps every individual to reach the most optimal point in its vicinity. This aids the evolutionary algorithm whose job now is primarily to place individuals near the minima. In this algorithm simulated annealing is used as a local search strategy. We run a few iterations of this evolutionary process. The simulated annealing modifies all the parameters by some small amount. If the resultant individual has a better fitness, it is accepted. If it has a lower fitness it is accepted with a probability given in equation (4).

$$p = e^{\frac{-\delta f}{T}} \qquad (4)$$

Here T is the temperature constant. δf is the change in fitness.

C. Neural Network

Any individual of the evolutionary approach represents a neural network. The fitness of the individual is the performance of the neural network in solving the diagnosis problem. The training database is used for the measurement of the performance. The training targets consist of an entry of 1 denoting the presence of the disease and 0 denoting its absence. The total error between the outputs and the targets is measured and summed for all the items in the training database. This is given in equation (5).

$$E = \sum_i \frac{|o_i - t_i|}{N} \qquad (5)$$

Here N is the total number of elements in the training database.

The algorithm puts a penalty on the networks with more number of neurons. This saves the algorithm from producing very large networks that may have a poor generalization. The penalty is proportional to the total number of neurons. The net fitness of the individual that needs to be minimized by the evolutionary algorithm may hence be given by equation (6)

$$Fit = E + \alpha H \tag{6}$$

Here α is the penalty constant and H is the total number of neurons in the neural network.

IV. RESULTS

The algorithm was implemented in JAVA. The data module of the entire program was used to feed data into the system. Similarly there were modules for the neural network as well as the evolutionary algorithm. The simulated annealing module was kept as a part of the neural network module. The fitness function of the evolutionary algorithm executed the simulated annealing module which in turn used neural network module for the evaluation.

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 [23]. 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 entire data was divided into training and testing data sets. The training dataset consisted of about 70% of the data randomly chosen from the dataset. The remaining 30% data was used as the testing dataset. The training data set was used for the evolution of the neural network using the discussed approach. The best individual after the entire evolution was the most optimal neural network. This network was then executed against the training and the testing data sets and the corresponding performance was noted. The methodology is shown in figure 3.

The simulation of the algorithm was done for 50 generations with 50 individuals. The number of neurons could vary from 1 to 25. At ant generation 40% of the individuals were contributed by the application of crossover operator, 20% by the application of mutation operator, 25% by the operation of new operator and 15% by the application of the jump operator. The elite count was kept to 1. The penalty constant was kept as 0.1 and the simulated annealing temperature constant had a value of 2. The simulation under these parameter values took approximately 2 minutes. At the end of the evolutionary process, the neural network so obtained had an accuracy of diagnosis of 82.96 on the training data and 82.38% on the testing data. The plot of the best individual in various generations is shown in figure 4.

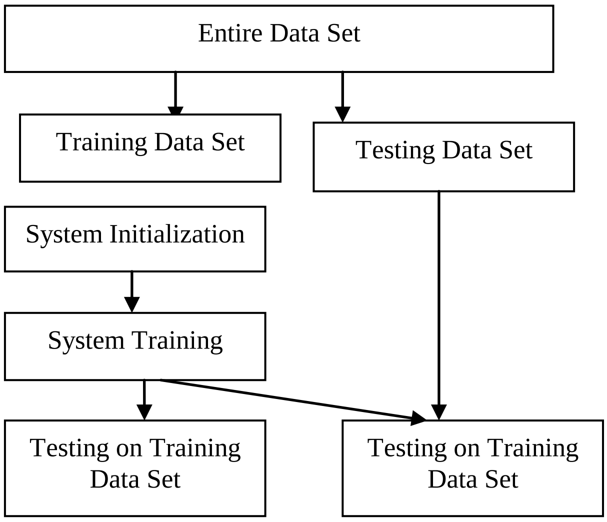

Figure 3 The working methodology

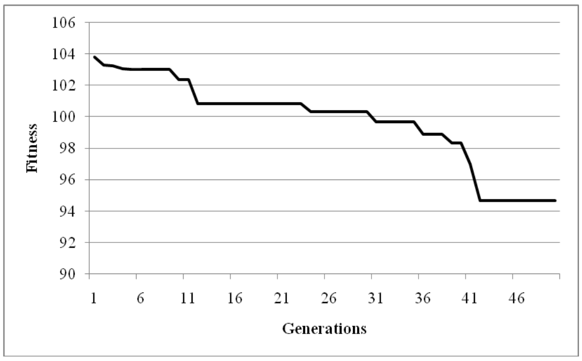

Figure 4 Plot of the fitness of best individual v/s generations

We further applied a few commonly used methods to the same data set to compare the efficiency of the proposed algorithm against these methods. The first method applied was of Artificial neural Network (ANN) trained with Back Propagation Algorithm. 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. 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 second method applied was 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 to each other with small changes. These had 12, 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.

The third method applied was of ANFIS. 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 final system so obtained had a performance of 88.9720% for the training data and 66.5236% for the testing data.

The fourth system was connectionist architecture of evolutionary ANN. Here the chromosome stored the presence or absence of connections in between neurons (along with weights) to eliminate a fully connected architecture. 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 final system had a performance of 77.38% for the training data set and 73.819% in the testing data set.

The last system applied was the conventional RBFN. Here the neurons had a spread of 55. The system so generated had a performance of 79.25% on the training data and 78.41% on the testing data.

The performance of the various models is analyzed in table 1. We can easily see that the proposed system gave the best performance than all the commonly known methods. This system hence presents an effective system for diagnosis of PIMA Indian diabetes.

Table 1. Comparison of results from various algorithms.

Method	**Training Accuracy**	**Testing Accuracy**
Proposed Algorithm	**82.96%**	**82.38%**
ANN with BPA	77.33%	77.73%
Ensemble (with BPA)	78.72%	76.98%
ANFIS	88.97%	66.52%
Evolutionary ANN	77.38%	73.81%
RBFN	79.25%	78.41%

V. CONCLUSIONS

In this paper we studied the problem of diagnosis of PIMA Indian diabetes. The problem was solved using an incremental evolution of neural network. Here the maximum permissible number of neurons in the hidden layer increased along with generations. As the algorithm proceeded, the neural network started attaining optimized architecture as well as weights and biases. The entire evolution was done using a variety of evolutionary operators. The evolutionary process was assisted by simulated annealing technique. This served as a local search strategy for the search of the global minima. The algorithm was able to achieve a high degree of accuracy for both training as well as testing data sets.

The performance of the algorithm was compared with a variety of techniques. This included the use of conventional Multi-Layer Perceptron with Back Propagation algorithm for training, modular neural networks, connectionist evolution of neural network, radial basis function network and adaptive neuro fuzzy inference system. The proposed algorithm performed better as compared to all these networks. This shows that the algorithm may be effectively used for the medical diagnosis.

The proposed algorithm was tested ageist a single data set of diabetes. The experiment may be repeated for more data sets to ensure the generalization of the observations. The algorithm may further be assisted by more intelligent heuristic or soft computing techniques to serve as the local search strategy. The complete search space of the neural network evolution is highly complex. Better techniques to adaptively explore this search space as well as to strike a good balance between exploration and exploitation may be devised. All this should result in escaping from the local minima and timely convergence to the global minima. All this may be carried in future.

REFERENCES

[1] J. D. Bronzino, *Biomedical Engineering Fundamentals*, CRC Press, 2006

[2] A. Shukla, R. Tiwari, R. Kala, *Real Life Applications of Soft Computing*, CRC Press, 2010

[3] N. K. Kasabov, *Foundation s of Neural Networks, Fuzzy Systems, and Knowledge Engineering,* MIT Press, 1998

[4] Western Sydney Endocrine Centre. Available via http://designchip.net/endocrine/index.php?option=com_content&task=view&id=6&Itemid=29, 2004

[5] Pan American Health Organization/World Health Organization, Diabetes Increasing along U.S.-Mexico Border. Available via http://www.fep.paho.org/eng/TechnicalCooperation/Diabetes/SurveyResults/tabid/318/language/en-US/Default.aspx, 2006

[6] Brussels, International Diabetes Federation's World Diabetes Day 2002 Focuses On Eye Complications. WDD 2002. Available via http://www.idf.org/home/index.cfm?node=535, 2002

[7] Kathleen Doheny, Too Few Understand Diabetes' Dangers. HealthDay Reporter. . Available via http://health.usnews.com/articles/health/healthday/2008/10/28/too-few-understand-diabetes-dangers.html, 2008

[8] R. Kala, A. Shukla, R. Tiwari, A Novel Approach to Classificatory problem using Neuro-Fuzzy Architecture, *International Journal of Systems, Control and Communications (IJSCC)*, Inderscience Publishers, 2010

[9] R. Kala, A. Shukla, R. Tiwari, A Novel Approach to Classificatory Problem using Grammatical Evolution based Hybrid Algorithm, *International Journal on Futuristic Computer Applications*, 2010

[10] R. R. Janghel, A. Shukla, R. Tiwari, "Decision Support system for fetal delivery using Soft Computing Techniques", Proceedings of the *Fourth International Conference on Computer Sciences and Convergence Information Technology,* ieeexplore, pp 1514-1519, 2010, Seoul, Korea.

[11] R. R. Janghel, A. Shukla, R. Tiwari, P. Tiwari, "Clinical Decision support system for fetal Delivery using Artificial Neural Network", Proceedings of the *2009 International Conference on New Trends in Information and Service Science*, NISS 2009, pp 1070-1075, 2009, Gyeongju, Korea

[12] A. Shukla, R. Tiwari, P. Kaur, "Intelligent System for the Diagnosis of Epilepsy", Proceedings of the *IEEE World Congress on Computer Science and Information*

Engineering (CSIE), ieeexplore, pp 755-758, 2009, Los Angeles/Anaheim, USA

[13] A. Shukla, R. Tiwari, P. Kaur, R. R. Janghel, “Diagnosis of Thyroid Disorders using Artificial Neural Networks”, Proceedings of the *IEEE International Advanced Computing Conference*, ieeexplore, pp 1016-1020, Patiala, India

[14] R. Kala, A. Shukla, R. Tiwari, “Comparative analysis of intelligent hybrid systems for detection of PIMA indian diabetes” , Proceedings of the *IEEE 2009 World Congress on Nature & Biologically Inspired Computing*, NABIC '09, pp 947 - 952, 2009, Coimbatote, India

[15] A. Shukla, R. Tiwari, *Intelligent Medical technologies and Biomedical Engineering: Tools and Applications*, IGI Global Publishers, 2010

[16] A. Shukla, R. Tiwari, *Biomedical Engineering and Information Systems: Technologies, Tools and Applications*, IGI Global Publishers, 2010

[17] R. Kala, A. Shukla, R. Tiwari, “Fuzzy Neuro Systems for Machine Learning for Large Data Sets”, Proceedings of the *IEEE International Advance Computing Conference*, ieeexplore, pp 541-545, 2009, Patiala, India

[18] A. Shukla, R. Tiwari, H. K. Meena, R. Kala, “Speaker Identification using Wavelet Analysis and Modular Neural Networks”, *Journal of Acoustic Society of India (JASI)*, Vol 36, No. 1, pp 14-19, 2009

[19] R. Kala, H. Vazirani, A. Shukla, R. Tiwari, “Fusion of Speech and Face by Enhanced Modular Neural Network”, Proceedings of the *International Conference on Information Systems, Technology and Management*, ICISTM 2010, CCIS 54, pp. 363-372, 2010, Bankok, Thailand

[20] P. Melin, O. Castillo, *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*, Springer, 2005

[21] P. Melin *et al*, Modular Neural Networks and Fuzzy Sugeno Integral for Face and Fingerprint Recognition, In Applied Soft Computing Technologies: The Challenge of Complexity, Springer, 2006

[22] N. G. Pedrajas, C. H. Martinez, J. M. Perez, “Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks)”, *Neural Networks*, Vol. 15, 2002, pp 1259–1278

[23] 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