Universal Journal of Computer Science and Engineering Technology
1 (2), 73-78, Nov. 2010.

Cancer Diagnosis Using Modified Fuzzy Network

Essam Al-Daoud
Faculty of Science and Information Technology
Computer Science Department, Zarka University
13110 Zarka, Jordan
essamdz@zpu.edu.jo

Abstract— in this study, a modified fuzzy c-means radial basis functions network is proposed. The main purposes of the suggested model are to diagnose the cancer diseases by using fuzzy rules with relatively small number of linguistic labels, reduce the similarity of the membership functions and preserve the meaning of the linguistic labels. The modified model is implemented and compared with adaptive neuro-fuzzy inference system (ANFIS). The both models are applied on "Wisconsin Breast Cancer" data set. Three rules are needed to obtain the classification rate 97% by using the modified model (3 out of 114 is classified wrongly). On the contrary, more rules are needed to get the same accuracy by using ANFIS. Moreover, the results indicate that the new model is more accurate than the state-of-art prediction methods. The suggested neuro-fuzzy inference system can be re-applied to many applications such as data approximation, human behavior representation, forecasting urban water demand and identifying DNA splice sites.

***Keywords-* fuzzy c-means, radial basis functions, fuzzy-neuro, rules, cancer diagnosis**

I. INTRODUCTION

The subjectivity of the specialist is an important problem of diagnosing a new patient. It can be noted that the decision of the professionals is related to the last diagnostic. Therefore, to enhance the diagnostic and to interpret the patients signal accurately, the huge volume of the empirical input- output data must be automated and used effectively. Cancer diagnosis can be seen as a matching procedure whose objective is to match each set of the symptoms (feature space) to a specific case. Many studies have been introduced to develop cancer diagnosis systems by using intelligent computation see for example [1-2]. Kiyan and Yildirim applied general regression neural network, multilayer perceptrons (MLP), and probabilistic neural network on Wisconsin breast cancer dataset. They show that the general regression neural network is the most accurate model for breast cancer classification [3]. Zhou et al. introduced a new system based on neural network ensemble [4]. They named it Neural Ensemble Based Detection (NED) and used it to identify the images of the cancer cells. Radial Basis Functions (RBF) represents alternative approach to MLP's in universal function approximation [5]. It outperforms MLP due to the convergence speed and the capability in handling the non-stationary datasets.

Fuzzy-Neuro system uses a learning procedure to find a set of fuzzy membership functions which can be expressed in form if-then rules. Fuzzy-Neuro has many advantages: Firstly it allows incorporating our experience and the previous knowledge into the classifier. Secondly it provides an understanding about the characteristic of the dataset. Thirdly it helps to find the dependencies in the datasets. Fourthly it gives an explanation which allows us to test the internal logic [6-8]. In this paper, a new intelligent decision support system for cancer diagnosis is constructed and tested. The suggested system is based on a modified version of fuzzy c-means method and radial basis functions neural network. It can be trained to establish a quality prediction system for a cancer disease with different parameters. Moreover the suggested neuro-fuzzy inference system can be applied to many applications such as data approximation, dynamic system processing, urban water demand forecasting, identifying DNA splice sites and image compression. In general the suggested model can be applied to any data needs classification, interpretation, adaptation or rules' extraction. For example the human behavioral representation in synthetic forces consists from several fuzzy parameters; e.g., interactions, responses, biomechanical, physical, psychophysical and psychological parameters. Such this data are very suitable to be modeled by using the suggested neuro-fuzzy inference system due to the fact that human behavior represents highly complex nonlinear and adaptable systems.

II. FUZZY-NEURO SYSTEMS

Fuzzy-Neuro system can be designed by using various architectures. To improve the performance of the system, three matters must be handled: finding the optimal number of the rules, discovering the appropriate membership functions, and tuning of both. The following is a short overview of the major works in this area [9-12]:

- **Fuzzy Adaptive Learning Control Network (FALCON):** FALCON consists from five layers. Tow nodes for input data, one for the desired output and the rest is for the actual output. The supervised learning is implemented by using backpropagation algorithm .
- **Generalized Approximate Reasoning Based Intelligent Control (GARIC):** Several specialized feedforward network are used to implement GARIC. The

Corresponding Author: Essam Al-Daoud, Computer Science Department, Zarka University, Jordan.

main disadvantage of GARIC is the complexity of the learning algorithm.

- **Neuro-Fuzzy Controller (NEFCON):** NEFCON Consists from two phases. The first is used to embed the rules and the second modifies and shifts the fuzzy sets. The main disadvantage of NEFCON is that it needs a previously defined rule base.
- **Adaptive Network Based Fuzzy Inference System (ANFIS)**: ANFIS works with different activation functions and uses un-weighted connections in each layer. ANFIS consists from five layers and can be adapted by a supervised learning algorithm.
- **Neuro-Fuzzy Classification (NEFCLASS)** NEFCLASS can be created from scratch by learning or it can be refined by using partial knowledge about patterns.
- **Fuzzy Learning Vector Quantization (FLVQ): FLVQ** is based on the fuzzification of LVQ and it is similar to Adaptive Resonance Theory (ART). The main disadvantage of FLVQ is not tested widely [13].
- **Evolutionary Fuzzy Neural Network (EFNN):** EFNN uses evolutionary algorithms to train the fuzzy neural network, **Aliev** et. At. Train the recurrent fuzzy neural networks by using an effective differential evolution optimization (DEO) [14].

The proposed method will be compared with ANFIS for two reasons: firstly ANFIS has been written in many programming languages including Matlab fuzzy logic toolbox. Secondly ANFIS is widely tested in various applications such as noise cancellation, system identification, time series prediction, medical diagnosis systems, and control [15]. Fig. 1 illustrates the architecture of ANFIS. For simplicity, we assume that ANFIS has two inputs x and y and one output z, suppose that the rule base contains two fuzzy if-then rules of Takagi and Sugeno's type [16]:

Rule 1: *If x is A_1 and y is B_1, then* $f_1 = p_1x + q_1y + r_1$
Rule 2: *If x is A_2 and y is B_2, then* $f_2 = p_2x + q_2y + r_2$

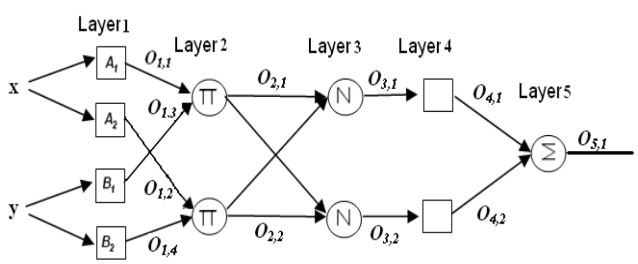

Figure 1. ANFIS architecture

Let $O_{j,i}$ represents the output of the i^{th} node in the layer j, the ANFIS output is calculated by using the following steps [16]:

1- $O_{1,i} = \mu_{A_i}(x)$ *i =1 ,2*

2- $O_{1,i} = \mu_{B_{i-2}}(x)$ *i=3,4*

3- $O_{2,i} = O_{1,i} \times O_{1,i+2}$ *i=1,2*

4- $O_{3,i} = \frac{O_{2,i}}{O_{2,1} + O_{2,2}}$ *i=1,2*

5- $O_{4,i} = O_{3,i} f_i = O_{3,i}(p_i x + q_i x + r_i)$ *i=1,2*

6- $O_{5,1} = \sum O_{4,i}$

The membership function for *A* (or *B*) can be any parameterized membership function such as:

$$\mu_A = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^2} \quad \text{or} \quad (1)$$

$$\mu_A = \exp\left(-\left(\frac{x - c_i}{a_i}\right)^2\right) \quad (2)$$

Network can be trained by finding suitable parameters for layer 1 and 4. Gradient decent are typically used for non linear parameters of layer 1 while batch or recursive least squares are used for linear parameters of layer 4 or even combination of both.

III. THE PROPOSED MODEL

The main purposes of the suggested model are to diagnose the cancer diseases by using fuzzy rules with relatively small number of linguistic labels, reduce the similarity of the membership functions and preserve the meaning of the linguistic labels. The learning algorithm of the proposed model consists of three phases:

Phase 1: Modified fuzzy c-means algorithm (MFCM). The standard fuzzy c-means has various well-known problems, namely the number of the clusters must be specified in advanced, the output membership functions have high similarity, and FCM is unsupervised method and cannot preserve the meaning of the linguistic labels. On the contrary, the grid partitions method solves some of the previous matters, but it has very high number of the output clusters. The basic idea of the suggested MFCM algorithm is to combine the advantages of the two methods, such that, if more than one cluster's center exist in one partition then merge them and calculate the membership values again, but if there is no cluster's center in a partition then delete it and redefined the other clusters. Algorithm 1 illustrates the modified fuzzy c-means algorithm

Algorithm 1. Modified fuzzy c-means algorithm
Input: Pattern vector, target vector, K the number of the patterns and the partitions intervals of each attribute P_{k,i_k}
Output: Centers, membership values and the new projected partitions.

1- Delete all the attributes that have low correlation with the target
2- For each class in the target vector apply the following steps on the corresponding patterns.
3- Choose $c=K/2$ seeds (first c patterns are selected as seeds).
4- Compute the membership values M using

$$m_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\| u_k - c_i \|}{\| u_k - c_j \|}\right)^{2/(q-1)}}, \quad k=1,2,\ldots K \text{ and } q>1. \tag{3}$$

5- Calculate c cluster centers using:

$$c_i = \frac{\sum_{k=1}^{K} m_{ik}^{q} u_k}{\sum_{k=1}^{K} m_{ik}^{q}} \tag{4}$$

6- Compute the objective function.

$$J(M, c_1, c_2, \ldots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{k=1}^{K} m_{ik}^{q} \| u_k - c_i \|^2 \tag{5}$$

7- If either J is less than a certain threshold level or the improvement in the previous iteration is less than a certain tolerance then go to step 8, else go to step 4.
8- If there are centers that exist in one partition= $\prod_{k=1}^{K} P_{k,i_k}$ then merge it

$$c_{new} = \frac{\sum_{v=1}^{n} c_v}{n}, \text{ c=c-v+1} \tag{6}$$

9- If all partitions that are related to a projected partition $= P_{k,i_h} \times \prod_{k=1,k\neq h}^{K} P_{k,i_k}$ do not contain a center then delete the projected partition P_{k,i_h} and redefined the attribute h partitions.

10- If step 8 or 9 is true then go to step 4.

Phase 2: Sort the initial fuzzy rules (centers) for each target class, the weight of rule x with regard to class y is calculated as following:

$$W(R_x^y) = NP_y - \sum_{i=1, i\neq y}^{z} NP_i, \tag{7}$$

Where NP is the number of patterns that have high participate in the antecedents and the consequences of the rule x (the high participate means that the membership is not less than T for each attribute. In this paper $T=0.5$).

Phase 3: Modified RBF learning algorithm (MRBF). Fig. 2 shows the architecture of the MRBF, the hidden layers consist from n layer, where n is the number of the target classes. Each hidden layer growths iteratively, one node (Rule) per iteration until accurate solution is found. The output layer consists from n nodes, one node for each class. The MRBF is trained by solving the system of equations using pseudo-inverse.

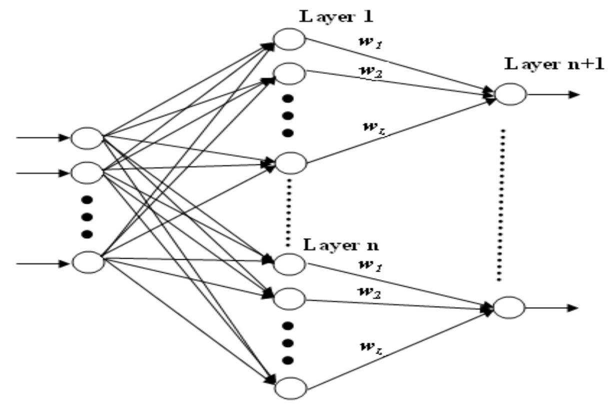

Figure 2. MRBF architecture

Algorithm 2. Modified RBF learning algorithm.

Input: Pattern vector, target vector, K the number of the patterns

Output: The weight of the hidden-output layer, the representative rules

1- Pick up the next highest weight of the rules (centers) for class i and represent it as new node in the hidden layer i.
2- Calculate the new outputs of all the hidden layers and all the patterns. Where the output of the node j and the pattern k is

$$\lambda_{kj} = \varphi\left(\| x_k - t_j \| / \sigma_j\right) = e^{-\|x_k - t_j\|^2 / \sigma_j^{\,2}} \tag{8}$$

t_j is the current center(rule) and σ is the width.

3- Find the new weights of the hidden-output layer for each class by solving the following system:

$$\begin{bmatrix} w_1 & w_2 & \ldots & w_z \end{bmatrix}^T = \psi^{+} \begin{bmatrix} t_1 & t_2 & \ldots & t_z \end{bmatrix}^T \tag{9}$$

Where z is the number of the processed centers (rules) and ψ^{+} is the pseudo-inverse of the matrix ψ

$$\psi = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \ldots & \lambda_{1z} \\ \lambda_{21} & \lambda_{22} & \ldots & \lambda_{2z} \\ \ldots & \ldots & \ldots & \ldots \\ \lambda_{k1} & \lambda_{k2} & \ldots & \lambda_{kz} \end{bmatrix} \tag{10}$$

4- If the error is less than a threshold then stop, else go to step 1

IV. Experimental Results

In this section we will apply ANFIS and the modified Fuzzy RBF (MFRBF) on "Wisconsin Breast Cancer" data set. This data set contains 569 instances (patterns) distributed into two classes (357 benign and 212 malignant). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass [17]. The number of the attributes that are used in this paper is 11 (10 real-valued input features and diagnosis). The features are summarized in Table 1.

TABLE I. Diagnostic Breast Cancer Features

Feature	Max value	Min Value	Correlation
Radius	28.1	6.981	0.7300
Texture	39.3	9.710	0.4152
Perimeter	188.5	43.790	0.7426
Area	2501	143.50	0.7090
Smoothness	0.2	0.0526	0.3586
Compactness	0.3	0.0194	0.5965
Concavity	0.4	0	0.6964
Concave points	0.2	0	0.7766
Symmetry	0.3	0.106	0.3305
Fractal dimension	0.1	0.05	-0.0128

Matlab 7.0 is used to implement the both algorithms, the data is normalized by using the Matlab function premnmx() and then the correlation between each feature and the target is calculated and listed in Table 1. It can be observed that the symmetry feature and fractal dimension feature have the lowest correlation, thus they are deleted and the other 8 features are used. Fig. 3 shows the first feature (Radius) distribution. A k-folding scheme with k=5 is applied. The training procedure is repeated 5 times, each time with 80% (455 patterns) of the patterns as training and 20% (114) for testing. All the reported results are obtained by averaging the outcomes of the five separate tests.

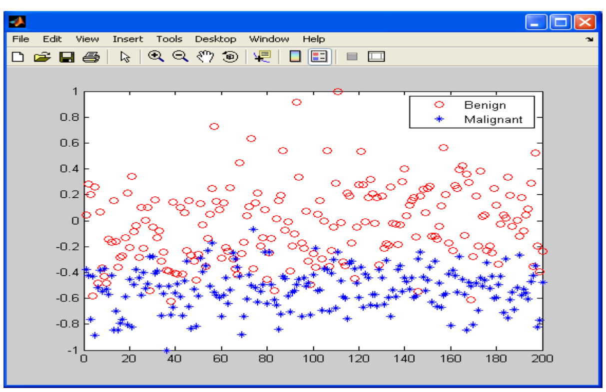

Figure 3. The first feature (Radius) distribution

The initial shadow partitions for each feature in Algorithm 1 is chosen to be (small, Medium, Large) corresponding to ([-1, -3.3), [-3.3,3.3), [3.3,1]). The number of the initial centers (rules) is K/2=227. After running Algorithm 1 for 7 epochs many centers are merged and the final number of the centers is 23. On the other hand, the projected partitions are redefined as shown in Table 2. The deleted partitions can be substituted by its neighborhood partition; for example, if the large partition is deleted then the medium partition means (medium or large). The projected partitions in Table 2 indicate that the fifth feature (smoothness) can be ignored.

TABLE II. The Output Projected Partitions

Feature	Max value	Min Value	Correlation
Radius	28.1	6.981	0.7300
Texture	39.3	9.710	0.4152
Perimeter	188.5	43.790	0.7426
Area	2501	143.50	0.7090
Smoothness	0.2	0.0526	0.3586
Compactness	0.3	0.0194	0.5965
Concavity	0.4	0	0.6964
Concave points	0.2	0	0.7766
Symmetry	0.3	0.106	0.3305
Fractal dimension	0.1	0.05	-0.0128

In phase 2, the rules are sorted according to its weights, the highest weight rule is:

If (radius is small and texture is small and perimeter is small and area is small and compactness is small and concavity is small and concave point is small)

Then Benign

For simplicity, the above rule will be written as following:

if (s, s, s, s, s, s, s) **then** Benign

The number of the layers are needed in phase 3 is two hidden layers, and one output layer, after two nodes (rules) are added to the hidden layers (one for each), the classification rate becomes 96% (4 out of 114 is classified wrongly). If another node is added to the first layer then the classification rate becomes 97% (3 out of 114 is classified wrongly). Table 3 compares the number of rules and the accuracy that are generated by ANFIS and MFRBF.

TABLE III. Comparison between ANFIS and MFRBF

Method	Rules Number	classification rate
ANFIS	2, σ =0.8	0.9474
MFRBF	2	0.9649
ANFIS	2, σ=0.5	0.9386
MFRBF	2	0.9649
ANFIS	3, σ=0.4	0.9474
MFRBF	3	0.9737
ANFIS	7, σ=0.3	0.9737
MFRBF	7	0.9737
ANFIS	19, σ=0.2	0.9649
MFRBF	19	0.9821

Table 3 indicates that by using MFRBF we can get high accuracy with fewer rules. On the contrary, by using ANFIS more rules are needed to get the same accuracy. Moreover the features projected partition in ANFIS is ambiguous and can not preserve the meaning of the linguistic labels, see Fig. 4.

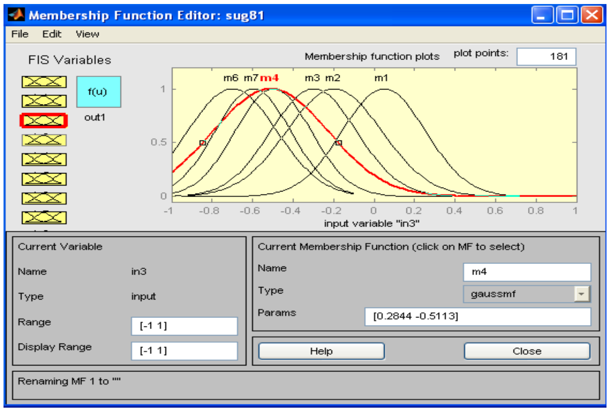

Figure 4. Ambiguous membership functions that are generated by ANFIS

The following is a sample rule produced by ANFIS:

If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf1) and (in5 is in5mf1) and (in6 is in6mf1) and (in7 is in7mf1) and (in8 is in8mf1)
Then (out1 is out1mf1)

On the other Hand, the output rules in MFRDF are unambiguous and do not need any farther processing. The best number of the rules is trade-off between the accuracy and the rules number, for example, the following three rules are recommend, these rules are produced by MFRBF with acceptable classification accuracy (97%):

If (s, s, s, s, s, s, s) **then** Benign
If (m or *l*, m or *l*, m or *l*, m or *l*, m or *l*, m or *l*, m or *l*)
Then Malignant
If (m or *l*, m or *l*, m or *l*, s, m or *l*, s , m or *l*) **then** Malignant

In Table 4, CLOP package are used to implement and to compare the suggested model with the state-of-art prediction methods (CLOP Package http://clopinet.com/CLOP/). Two measurements are used: Balance Error Rate (BER) and Area Under Carve (AUC). The results indicate that MFRBF is more accurate than the other methods, where the balance error rate is 2.2, while the balance error rate is 9.92 by using nonlinear support vector machine (NonLinearSVM).

TABLE IV. COMPARISON BETWEEN THE STATE-OF-ART PREDICTION METHODS

Method	Testing	
	BER	AUC
ANFIS	4.41	98.49
MFRBF	2.20	99.21
NeuralNet	6.15	97.81
LinearSVM	12.36	93.75
Kridge	8.53	96.22
NaiveBayes	10.4	95.21
NonLinearSVM	9.92	96.98

V. CONCLUSION

To produce unambiguous rules that are suitable for cancer diagnosis, a modified fuzzy c-means radial basis functions (MFRBF) is introduced. The experimental results show that: we can use MFRBF to get high accuracy with fewer and unambiguous rules. The classification rate is 97% (3 out of 114 is classified wrongly) by using only three rules. On the contrary, more rules are needed to get the same accuracy by using ANFIS. Moreover the features projected partition in ANFIS is ambiguous and can not preserve the meaning of the linguistic labels. The results indicate that MFRBF is superior to state-of-art prediction methods, where the balance error rate is 2.2 by using MFRBF, while the balance error rate is 9.92 by using nonlinear support vector machine.

ACKNOWLEDGMENT

This research is funded by the Deanship of Research and Graduate Studies in Zarka University /Jordan

REFERENCES

[1] L. Fengjun, “Function approximation by neural networks,” Proceedings of the 5th international symposium on Neural Networks: Advances in Neural Networks, Beijing, China, pp. 384-390, 2008.

[2] V. S. Bourdès, S. Bonnevay, P. Lisboa, M. S. H. Aung, S. Chabaud, T. Bachelot, D. Perol and S. Negrier, “Breast cancer predictions by neural networks analysis: a Comparison with Logistic Regression,” 29th Annual International Conference of the IEEE EMBS Cité Internationale, Lyon, France, pp. 5424-5427, 2007.

[3] T. Kiyani, and T. Yildirim, “Breast cancer diagnosis using statistical neural networks,” Journal of Electrical & Electronics Engineering, vol 4, no. 2, pp. 1149-1153, 2004.

[4] Z. Zhou, Y. Jiang, Y. Yang, and S. Chen, “Lung cancer cell identification based on artificial neural network ensembles,” Artificial Intelligence In Medicine, vol 24, no. 1, pp. 25-36, 2002.

[5] Y.J. Oyang, S.C. Hwang and Y.Y. Ou, “Data classification with radial basis function networks based on a novel kernel density estimation algorithm,” IEEE Transaction on Neural networks, vol. 16, no. 1, pp. 225-236, 2005.

[6] k. Rahul, S. Anupam and T. Ritu, “Fuzzy Neuro Systems for Machine Learning for Large Data Sets,” Proceedings of the IEEE International Advance Computing Conference 6-7, Patiala, India, pp.541-545, 2009.

[7] C. Juang, R. Huang and W. Cheng, “An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems,” IEEE Transactions on Fuzzy Systems, vol. 18, no. 4, pp. 686 – 699, 2010.

[8] C., Juang, Y. Lin and C. Tu, “Recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing,” Fuzzy Sets and Systems, vol. 161, no. 19, pp. 2552-2562, 2010.

[9] S. Alshaban and R., Ali, “Using neural and fuzzy software for the classification of ECG signals,” Research Journal of Applied Sciences, Engineering and Technology, vol. 2, no. 1, pp. 5-10, 2010.

[10] W. Li, and Z. Huicheng, “Urban water demand forecasting based on HP filter and fuzzy neural network,” Journal of Hydroinformatics, vol. 12, no. 2, pp. 172–184, 2010.

[11] K. Vijaya, K. Nehemiah, H. Kannan and N.G. Bhuvaneswari, “Fuzzy neuro genetic approach for predicting the risk of cardiovascular diseases,“ Int. J. Data Mining, Modelling and Management, vol. 2, pp. 388-402, 2010.

[12] A. Talei, L. Hock, C. Chua and C. Quek, “A novel application of a neuro-fuzzy computational technique in event-based rainfall-runoff modeling,” Expert Systems with Applications: An International Journal, vol. 37, no. 12, pp. 7456-7468, 2010.

[13] Y. S. Kim, “Fuzzy neural network with a fuzzy learning rule emphasizing data near decision boundary,” Advances in Neural Networks, vol. 5552, pp. 201-207, 2009.

[14] R. A. Aliev, B. G. Guirimov, B. Fazlollahi and R. R. Aliev, “Evolutionary algorithm-based learning of fuzzy neural networks,” Part 2: Recurrent fuzzy neural networks, Fuzzy Sets and Systems, vol. 160, no. 17, pp. 2553-2566, 2009.

[15] C. P. Kurian, S. Kuriachan, J. Bhat, and R. S. Aithal, “An adaptive neuro fuzzy model for the prediction and control of light in integrated lighting schemes,” Lighting Research & Technology, vol. 37, no. 4, pp. 343-352, 2005.

[16] E. Al-Daoud, “Identifying DNA splice sites using patterns statistical properties and fuzzy neural networks, EXCLI Journal, vol. 8, pp. 195-202, 2009.

[17] O. L. Mangasarian, W. N. Street and W. H. Wolberg, “Breast cancer diagnosis and prognosis via linear programming,” Operations Research, vol. 43, no. 4, pp. 570-577, 1995.