A Resilient Back Propagation Algorithm for Breast Biopsy Classification Based on Artificial Neural Networks

Prof Fawzi M. Al Naima
Computer Engineering Department, College of Engineering, Nahrain University, Iraq
Email: fawzi_cmc@yahoo.com

Ali Hussian Ali Al Timemy
Biomedical Engineering Department, Al-Khawarzmi College of Engineering, Baghdad University, Iraq
Email: alialtimemy2006@yahoo.com

ABSTRACT

This paper presents the classification of benign and malignant breast tumor based on Fine Needle Aspiration Cytology (FNAC) and Feed forward Neural Network (FFNN) trained with Resilient Back Propagation algorithm (RBP). Five hundred and sixty nine sets of cell nuclei characteristics obtained by applying image analysis techniques to microscopic slides of FNAC samples of breast biopsy have been used in this study. These data were obtained from the University of Wisconsin Hospitals, Madison. The dataset consist of thirty features which represent the input layer to the FFNN. The FFNN will classify the input features into benign and malignant. The sensitivity, specificity and accuracy were found to be equal 97.5%, 100% and 98.73% respectively. It can be concluded that RPB network gives fast and accurate classification and it works as promising tool for classification of breast cell nuclei.

Key Words: Breast cancer, Tumor classification, FFNN, RBP, FNAC

1. Introduction

Neural Networks (NN) derive their power due to their massively parallel structure, and an ability to learn from experience. They can be used for fairly accurate classification of input data into categories, provided they are previously trained to do so. The accuracy of the classification depends on the efficiency of training. The knowledge gained by the learning experience is stored in the form of connection weights, which are used to make decisions on fresh input [1].

One computer technique under investigation is the Artificial Neural Network (ANN) [2]. Neural networks are tools for multivariate analysis that can be used to estimate disease risk. They are able to model complex nonlinear systems with significant variable interactions.

With 1 million new cases in the world each year, breast cancer is the commonest malignancy in women and comprises 18% of all female cancers. In the United Kingdom, where the age standardized incidence and mortality is the highest in the world, the incidence among women aged 50 approaches two per 1000 women per year, and the disease is the single commonest cause of death among women aged 40-50, accounting for about a fifth of all deaths in this age group. There are more than 14 000 deaths each year, and the incidence is increasing particularly among women aged 50-64, probably because of breast screening in this age group.

Of every 1000 women aged 50, two will recently have had breast cancer diagnosed and about 15 will have had a diagnosis made before the age of 50, giving a prevalence of breast cancer of nearly 2% [3].

Conventional methods of monitoring and diagnosing the diseases rely on detecting the presence of particular features by a human observer. Due to large number of patients in intensive care units and the need for continuous observation of such conditions, several techniques for automated diagnostic systems have been developed in recent years to attempt to solve this problem. Such techniques work by transforming the mostly qualitative diagnostic criteria into a more objective quantitative feature classification problem [4,5,6,7].

Breast cancer may be detected via a cautious study of clinical history, physical examination, and imaging with either mammography or ultrasound. However, definitive diagnosis of a breast mass can only be established through fine-needle aspiration (FNA) biopsy, core needle biopsy, or excisional biopsy [2]. Among these methods, FNA is the easiest and fastest method of obtaining a breast biopsy, and is effective for women who have fluid-filled cysts. FNA uses a needle smaller than those used for blood tests to remove fluid, cells, and small fragments of tissue for examination under a microscope [8].

Research works on the Wisconsin diagnosis breast cancer (WDBC) data grew out of the desire of Dr. Wolberg to diagnose breast masses accurately based solely on FNA [9,10,11]. They applied image processing techniques to derive the WDBC dataset directly from digital scans of FNA slides [12]. Then they employed machine learning techniques to differentiate benign from malignant samples [13], which could be the earliest study of machine learning application to breast cancer detection. Several attempts have been proposed for diagnosis of the breast cancer that includes FFNN [14], Radial Basis Function (RBF) network [15], Self organization maps [8], fuzzy classifiers [16,17,18,19,20], Linear Vector Quantization (LVQ) [21] and Support Vector Mechanics (SVM) [15,22,23].

The purpose of this study is to develop a RBP network which wills classify the breast biopsy samples into malignant and benign cysts based on the input features of cell nuclei characteristics. This network will act to help in the classification purposes of breast cancer.

2. The Classification Problem

The classification problem addressed in this study is the detection of malignant breast tumors from a set of benign and malignant samples, called the WDBC dataset, which was obtained from the University of Wisconsin Hospitals, Madison, available at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. Features in this dataset were computed from digitized FNA samples [10, 11, 13], as follows:

After the FNA sample was taken from a breast mass, the material was mounted on a microscope slide and stained to highlight the cellular nuclei. A portion of well differentiated cells was scanned using a digital camera. The image analysis software system Xcyt was used to isolate individual nuclei. An approximate boundary of each nucleus was provided as input and taken to convergence to the exact nuclear boundary, using a semi-automatic segmentation procedure called ‘‘snakes’’. Beginning with a user defined approximate boundary as an initialization, the snake locates the actual boundary of the cell nucleus. In order to evaluate the size, shape and texture of each cell nuclei, the following ten characteristics were derived:

(1) *Radius* is computed by averaging the length of radial line segments, which are lines

from the center of mass of the boundary to each of the boundary points.

(2) *Perimeter* is measured as the sum of the distances between consecutive boundary points.

(3) *Area* is measured by counting the number of pixels on the interior of the boundary and adding one-half of the pixels on the perimeter, to correct for the error caused by digitization.

(4) *Compactness (COM)* combines the *perimeter* and *area* to give a measure of the compactness of the cell, calculated as

$$COM = \frac{PER^2}{area} \qquad (1)$$

This dimensionless number is minimized

for a circle and increases with the irregularity of the boundary.

(5) Smoothness (*SM*) is quantified by measuring the difference between the length of each radial line and the mean length of the two radial lines surrounding it. If this number is small relative to the distance between consecutive boundary points, then the contour is smooth in that region. To avoid the numerical instability associated with small divisors, the following equation is used to calculate the smoothness

$$SM = \frac{\sum_{po\,\mathrm{int}\,s} \left| r_i - (r_i + r_{i+1})/2 \right|}{PER} \qquad (2)$$

where r_i is the length of the line from the center of mass of the boundary to each boundary point.

(6) *Concavity* is captured by measuring the size of any indentations in the boundary of the cell nucleus.

(7) *Concave* points is similar to concavity, but counts only the number of boundary points lying on the concave regions of the boundary, rather than the magnitude of such concavities.

(8) *Symmetry* is measured by finding the relative difference in length between pairs of line segments perpendicular to the major axis of the contour of the cell nucleus. The major axis is determined by finding the longest chord, which passes from a boundary point through the center of the nucleus. The segment pairs are then drawn at regular intervals. To avoid numerically unstable results due to extremely small segments, the sums are again divided, rather than summing the quotients,

$$symmetry = \frac{\sum_i \left| left_i - right_i \right|}{\sum_i \left| left_i + right_i \right|} \qquad (3)$$

where $left_i$ and $right_i$ denote the lengths of perpendicular segments on the left and right of the major axis, respectively.

(9) *Fractal dimension* is approximated using the ‘‘coastline approximation’’ described by Mandelbrot [24]. The perimeter of the nucleus is measured using increasingly larger ‘‘rulers’’. As the ruler size increases, the precision of the measurement decreases, and the observed perimeter decreases. Plotting these values on a log–log scale and measuring the downward slope gives the negative of an approximation to the fractal dimension.

(10) *Texture* is measured by finding the variance of the gray-scale intensities in the component pixels.

The mean value, standard error, and the extreme (largest or ‘‘worst’’) value of each characteristic were computed for each image, which resulted in 30 features of 569 images, yielding a database of 569 X 30 samples representing 357 benign and 212 malignant cases.

3. Feed-Forward neural Networks

A neural network is a parallel distributed information processing structure consisting of processing elements (neurons) interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches into as many collateral connections as desired [25].

Neural networks develop information processing capabilities by learning for examples. Learning techniques can be roughly divided into two categories [26]: supervised learning and unsupervised learning.

In unsupervised learning the neural network is autonomous; it processes the data it is presented with, finds out about some of the properties of the data set and learns to reflect these properties in its output. What exactly these properties are, that network can learn to recognize, depends on the particular network model and learning method.

One of the most popular neural net paradigms is the feed-forward neural network. In a feed-forward neural network, the neurons are usually arranged in layers [27]. A feed-forward neural net is denoted as,

where: $N_I \times N_1 \times \ldots \times N_i \times \ldots N_L \times N_O$

- N_I represents the number of input units;
- L represents the number of hidden layers;

- N_i represents the number of units from the hidden layer *I*, *i*=1,2,…*L*.
- N_O represents the number of output units.

By convention, the input layer does not count, since the input units are not processing units, they simply pass on the input vector x. Units from the hidden layers and output layer are processing units. Figure 1 gives a typical fully connected 2-layer feed-forward network with a 3×4×3 structure.

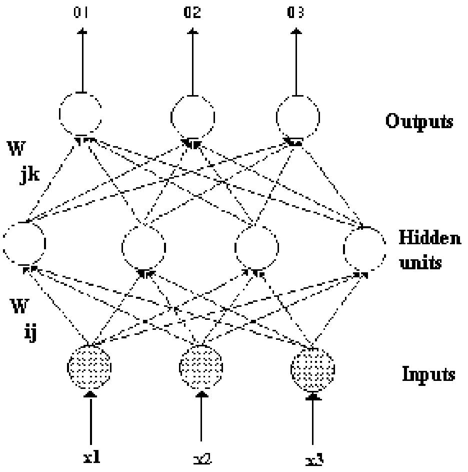

Figure 1. A 3x4x3 feed-forward neural network.

Each processing unit has an activation function that is commonly chosen to be the sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (4)$$

The net input to a processing unit j is given by:

$$net_j = \sum_i w_{ij} x_i + \theta_j \qquad (5)$$

Where x_i's are the outputs from the previous layer, w_{ij} is the weight (connection strength) of the link connecting unit i to unit *j*, and θ_j is the bias of unit *j*, which determines the location of the sigmoid function on the x-axis.

The activation value (output) of unit *j* is given by:

$$a_j = f(net_j) = \frac{1}{1+e^{-net_j}} \qquad (6)$$

The objective of different supervised learning algorithms is the iterative optimization of a so called error function representing a measure of the performance of the network. This error function is defined as the mean square sum of differences between the values of the output units of the network and the desired target values, calculated for the whole pattern set. The error for a pattern p is given by

,

$$E_p = \sum_{j=1}^{N_O} (d_{pj} - a_{pj})^2 \qquad (7)$$

where d_{pj} and a_{pj} are the target and the actual response value of output neuron *j* corresponding to the pattern p.

The total error is,

$$E = \sum_{p=1}^{P} \frac{1}{2} E_p = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{N_O} (d_{pj} - a_{pj})^2 \qquad (8)$$

where *P* is the number of the training patterns.

During the training process a set of pattern examples is used, each example consisting of a pair with the input and corresponding target output. The patterns are presented to the network sequentially, in an iterative manner, the appropriate weight corrections being performed during the process to adapt the network to the desired behavior. This iterating continues until the connection weight values allow the network to perform the required mapping. Each presentation of the whole pattern set is named an epoch.

One of the most popular supervised learning algorithms for feed-forward neural networks is Backpropagation [27]. In this algorithm the minimization of the error function is carried out using a gradient-descent technique. The necessary corrections to the weights of the network for each moment t are obtained by calculating the partial derivative of the network error function in relation to each weight w_{ij}. A gradient vector representing the steepest increasing direction in the weight space is thus obtained. The next step is to compute the resulting weight update. In it simplest form, the weight update is a scaled step in the opposite direction of the gradient. Hence, the weight update rule is

$$\Delta_p w_{ij}(t) = -\varepsilon \cdot \frac{\partial E_p}{\partial w_{ij}}(t), \qquad (9)$$

where $\varepsilon \in (0,1)$ is a parameter determining the step size and is called the learning rate.

A momentum may be used with the idea of incorporating in the present weight update some influence of the past iteration. The weight update rule becomes

$$\Delta_p w_{ij}(t) = -\varepsilon \cdot \frac{\partial E_p}{\partial w_{ij}}(t) + \alpha \cdot \Delta_p w_{ij}(t-1) \qquad (10)$$

where α is the momentum term and determines the amount of influence from the previous iteration to the present one.

4. The RBP Algorithm

The algorithm RBP introduced by M. Riedmiller in 1993 is a local adaptive learning scheme, performing supervised batch learning in feed-forward neural networks. The basic principle of RBP is to eliminate the harmful influence of the size of the partial derivative on the weight step. As a consequence, only the sign of the derivative is considered to indicate the direction of the weight update. To achieve this, we introduce for each weight w_{ij} its individual update-value Δ_{ij}(t), which solely determines the size of the weight-update [28,29].

It is introduced a second learning rule, which determines the evolution of the update-value Δ_{ij}(t),. This estimation is based on the observed behavior of the partial derivative during two successive weight-steps:

$$\Delta_{ij}(t) = \begin{cases} \eta^+ \cdot \Delta_{ij}(t-1), & if \ \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) > 0 \\ \eta^- \cdot \Delta_{ij}(t-1), & if \ \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \\ \Delta_{ij}(t-1), & else \end{cases} \qquad (11)$$

where $0 < \eta^- < 1 < \eta^+$

In words, the adaptation rule works as follows. Every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_{ij}(t) is decreased by the factor η^-. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions.

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, if the derivative is negative, the update-value is added:

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t), & if \ \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \Delta_{ij}(t), & if \ \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0, & else \end{cases} \qquad (12)$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \qquad (13)$$

However, there is one exception. If the partial derivative changes sign that is the previous step was too large and the minimum was missed, the previous weight-update is reverted:

$$\Delta w_{ij}(t) = -\Delta w_{ij}(t-1), \quad if \ \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \qquad (14)$$

Due to that 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting . $\frac{\partial E}{\partial w_{ij}}(t-1)=0$ in the Δ_{ij}, update-rule above.

The partial derivative of the total error is given by

$$\frac{\partial E}{\partial w_{ij}}(t)=\frac{1}{2}\sum_{p=1}^{P}\frac{\partial E_p}{\partial w_{ij}}(t) \qquad (15)$$

Hence, the partial derivatives of the errors must be accumulated for all P training patterns. This means that the weights are updated only after the presentation of all training patterns.

5. The Proposed FFNN Design

In the present work, the neural networks are used for the classification purposes. Three issues need to be settled in designing an ANN for a specific application:

- Topology of the network;
- Training algorithm;
- Neuron activation functions.

In our topology, the number of neurons in the input layer is 48 neurons for the ANN classifier. The output layer was determined by the number of classes desired. In our study, the output is either benign or malignant therefore; the output layer consists of one neuron. The hidden layer consists of twenty eight neurons. The general architecture of the proposed network is shown in Fig. 2.

Before the training process is started, all the weights are initialized to small random numbers. This ensured that the classifier network is not saturated by large values of the weights. In this experiment, the training set was formed by choosing 79 data sets for the testing process.

The tangent sigmoid (tansig) function was used as the neural activation function. The most important reason for choosing the sigmoid as an activation function for our networks is that the sigmoid function f(x) is differentiable for all values of x, which allows the use of the powerful BP learning algorithm.

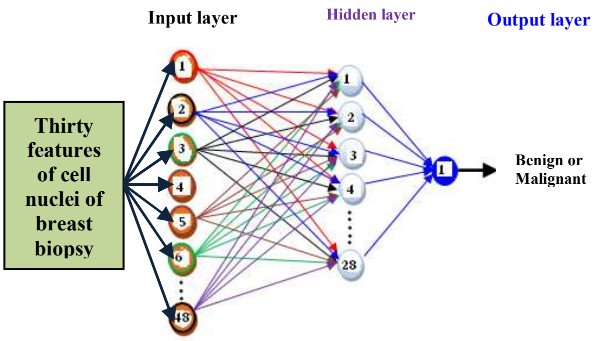

Figure 2. General architecture of the proposed FFNN

The proposed network was trained with all 490 cases (317 benign and 173 malignant cases). These 490 cases are fed to the FFNN with 48 input neurons, one hidden layer of 28 neurons and one output neuron. MATLAB software package version 7 is used to implement the software in the current work. When the training process is completed for the training data (490 cases), the last weights of the network were saved to be ready for the testing procedure. Learning rate is set to 0.5, the output of the network was -1 for the class benign normal and 1 for the class malignant. The training algorithm used for this network is BPA. The performance goal was met at 2600 epochs after a training time of 67 sec.

The testing process is done for 79 cases (40 benign and 39 malignant). These 79 cases are fed to the proposed network and its their output is recorded for calculation of the sensitivity, specificity and accuracy of prediction.

The accuracy of the classification depends on the efficiency of training. The knowledge gained by the learning experience is stored in the form of connection weights, which are used to make decisions on fresh input.

6. Results and Discussion

The performance of the algorithm was evaluated by computing the percentages of Sensitivity (SE), Specificity (SP) and Accuracy (AC), the respective definitions are as follows [30]:
Sensitivity: is the fraction of real events that are correctly detected among all real events.

$$SE = \frac{TP}{(TP + FN)} \times 100 \qquad (16)$$

Specificity: is the fraction of nonevents that has been correctly rejected.

$$SP = \frac{TN}{(TN + FP)} \times 100 \qquad (17)$$

$$AC = \frac{(TP + TN)}{(TN + TP + FN + FP)} \times 100 \qquad (18)$$

where TP is the number of true positives, TN is the number of true negatives, FN is the number of false negatives, and FP is the number of false positives. Since it is interesting to estimate the performance of classifier based on the classification of benign and malignant breast cell nuclei, the true positives (*TP*), false positives (*FP*), true negatives (*TN*), and false negatives (*FN*) are defined appropriately as shown below:

FP: Predicts benign as malignant.
TP: Predicts malignant as malignant.
FN: Predicts malignant as benign.

TN: Predicts benign as begin.

In our study, the output 1 indicates normal case. If the output is 2 this means that the patient may have abnormal kidney function. Sensitivity, specificity and accuracy of prediction have been calculated according to the above formals for all of the testing data (79 cases). Table 1 shows the resulted SE, SP and AC for testing data proposed networks.

Table 1. The results after training of the proposed network

	No. of cases	***SE***	***SP***	***CA***
RBP	79	97.5%	100%	98.73%

From the table, the obtained accuracy means that there was only one misidentification. This is regarded a very good and the system is reliable. The results showed that the algorithm can be reliable purposes in the classification purposes.
In practice, the number of neurons in the hidden layer varies according to the specific recognition task and is determined by the complexity and amount of training data available. If too many neurons are used in the hidden layer, the network will tend to memorize the data instead of discovering the features. This will result in failing to classify new input data.
The goal error was 0.01 and the network reached this error after 2600 epochs. Fig.3 displays the error of the training of the network versus epoch's number.

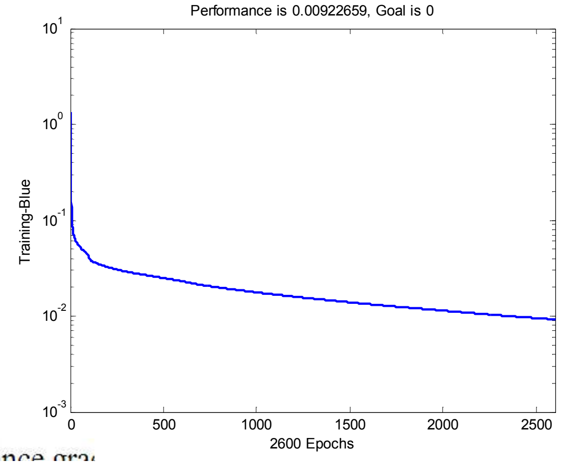

Figure 3. Error of the training versus epochs

7. Conclusion

In this paper, Resilient BPA has been implemented for classification of benign from malignant breast tumor. Five hundred and sixty nine sets of cell nuclei characteristics obtained by applying image analysis techniques to microscopic slides of FNAC samples of breast biopsy have been used in the current work. MATLAB software package version 7 is used to implement the software in the current work. These feature vectors which consist of thirty image analysis features each were carried out to generate training and testing of the proposed NN. The accuracy is calculated to evaluate its effectiveness of the proposed network. The obtained accuracy of the network was 98.73% whereas the sensitivity and specificity were found to be equal 100% and 98.73% respectively. We conclude that that the proposed system gives fast and accurate classification of breast tumors.

Acknowledgment

The authors would like to express their thanks to Dr. William H. Wolberg, W. Nick Street and Olvi L. Mangasarian for providing Breast Cancer Wisconsin (Diagnostic) Data Set used in this work. They are with the University of Wisconsin, 1210 West Dayton St., Madison, WI 53706 USA.

References:

[1] A. H. A. Al Timemy, F. M. Al-Naima and S. Mahdi, “Data acquisition system for myocardial infarction classification based on wavelets and neural networks,” in Proc. of the Fifth International Multi-Conference on Systems, Signals and Devices (IEEE SSD'08), Amman, Jordan, 2008.

[2] M. Chester, *Neural networks: a tutorial*, Englewood Cliffs, NJ: Prentice and Hall, 1993, ch.2.

[3] K. McPherson, C. M. Steel, J. M. Dixon, “ABC of Breast Diseases, Breast cancer—epidemiology, risk factors, and genetics”, BMJ, Vol. 321 9, pp. 624-628, September, 2000.

[4] E. D. Ubeyli, “Implementing automated diagnostic systems for breast cancer detection,” Expert Systems with Applications,Vol. 33, pp. 1054–1062, 2007.

[5] H. Kordylewski, D. Graupe and K. Liu, A novel large-memory neural network as an aid in medical diagnosis applications. IEEE Transactions on Information Technology in Biomedicine, 5(3), 202–209, 2001.

[6] N. Kwak, andC. H. Choi, “Input feature selection for classification problems”. IEEE Transactions on Neural Networks, 13(1), 143–159, 2002.

[7] E. D. Ubeyli and I. Guler, “Feature extraction from Doppler ultrasound signals for automated diagnostic systems”. Computers in Biology and Medicine, 35(9), 735–764, 2005.

[8] Mu Tingting and A. K. Nandi, “Breast cancer detection from FNA using SVM with different parameter tuning systems and SOM–RBF classifier”, Journal of the Franklin Institute, vol 344,pp. 285–311, 2007.

[9] W.N. Street, W.H. Wolberg and O.L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, In proceedings of IST/SPIE Symposium on Electronic Imaging: Science and Technology, vol. 1905, San Jose, CA, 1993, pp. 861–870.

[10] W.H. Wolberg, W.N. Street, O.L. Mangasarian, Breast cytology diagnosis via digital image analysis, Anal.Quant. Cytol. Histol. 15 (6) (1993) 396–404.

[11] W.H. Wolberg, W.N. Street, O.L. Mangasarian, Machine learning techniques to diagnose breast cancer from fine-needle aspirates, Cancer Lett. 77 (1994) 163–171.

[12] W.H. Wolberg, W.N. Street, O.L. Mangasarian, Image analysis and machine learning applied to breast cancer diagnosis and prognosis, Anal. Quant. Cytol. Histol. 17 (2) (1995) 77–87.

[13] O.L. Mangasarian, Breast cancer diagnosis and prognosis via linear programming, Cancer Lett. 43 (4) (1995) 570–577.

[14] R. Setiono, “Generating concise and accurate classification rules for breast cancer diagnosis”. Artificial Intelligence in Medicine, 18(3), 205–217,2002.

[15] T.S. Subashini, V. Ramalingam and S. Palanivel, “Breast mass classification based on cytological patterns using RBFNN and SVM”, Expert Systems with Applications, doi:10.1016/j.eswa.2008.06.127 , 2008.

[16] G. Schaefera, M. Závišekb and T. Nakashima, “Thermography based breast cancer analysis using statistical features and fuzzy classification”, Pattern

Recognition,doi:10.1016/j.patcog.2008.08.007, 2008

[17] P. Reyes, C. Andre´s and M. Sipper, (1999). A fuzzy-genetic approach to breast cancer diagnosis. Artificial Intelligence in Medicine (17), 131–155.

[18] R. Mousa, Q. Munib and A. Moussa, "Breast cancer diagnosis system based on wavelet analysis and fuzzy-neural", Expert Systems with Applications 28 (2005) 713–723.

[19] D. Nauck and R. Kruse, "Obtaining interpretable fuzzy classification rules from medical data", Artificial Intelligence in Medicine(16), 149–169, 1999.

[20] J. Abonyi and F. Szeifert, "Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recognition Letters, 14(24), 2195–2207, 2003.

[21] D. E. Goodman, L. Boggess and A. Watkins, "Artificial immune system classification of multiple-class problems". In Proceedings of the artificial neural networks in engineering (pp. 179–183), 2002.

[22] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis", Expert Systems with Applications 36 (2009) 3240–3247.

[23] K. Polat and S. Gunes, "Breast cancer diagnosis using least square support vector machine. Digital Signal Processing, 17(4), 694–701, 2007.

[24] B. B. Mandelbrot, "The Fractal Geometry of Nature", W. H. Freeman and Company, New York, 1997.

[25] C. Barbălată and L. Leuştean, "Average monthly liquid flow forecasting using neural networks, draft paper, 2002.

[26] S. Haykin, "*Neural networks: a comprehensive foundation*". New York: Macmillan, 1994.

[27] E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing: Explorations in the microstructure of cognition; Vol. 1: Foundations. The MIT Press, Cambridge, Massachusetts, 1986.

[28] M. Riedmiller., Untersuchungen zu Konvergen und Generalisierungs-verhalen überwachter Lernverfahren mit dem SNNS, in A. Zell, editor, SNNS 1999 Workshop Proceedings, Stuttgart, September 1993.

[29] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in H. Ruspini, editor, Proceedings of the IEEE International Conference on Neural Networks (ICNN), San Francisco, USA, 1993, pp. 586-591.

[30] A. H. A. Al Timemy, "Self-Organization Maps for Prediction of Kidney Dysfunction", In *Proc. 16th Telecommunications Forum TELFOR*, Belgrade, Serbia, 2008.