# INTERPRETATION TRAINED NEURAL NETWORKS BASED ON GENETIC ALGORITHMS

Safa S. Ibrahim[1] and Mohamed A.Bamatraf [1]

[1]Department of Computer Science, Assiut University, FCI, Assiut, Egypt

***ABSTRACT***

*In this paper, constructive learning is used to train the neural networks. The results of neural networks are obtained but its result is not in comprehensible form or in a black box form. Our goal is to use an important and desirable model to identify sets of input variable which results in a desired output value. The nature of this model can help to find an optimal set of difficult input variables. Accuracy. Genetic algorithms are used as an interpretation of achieving neural network inversion. On the other hand the inversion of neural network enables to find one or more input patterns which satisfy a specific output. The input patterns obtained from the genetic algorithm can be used for building neural network system explanation facilities.*

***KEYWORDS***

*Neural Networks, Genetic Algorithms , Constructive Learning, Accuracy.*

## 1. INTRODUCTION

Extracting classification rules is a procedure of extraction of information and knowledge that are hid in data in the form of classification rules, unknown by people and potentially useful from a large quantity of data with multiple characteristics that is uncompleted, containing noise, fuzzy and random. As a form of cross-discipline direction that syncretises numerous disciplines as well as database technology, statistics, artificial neural networks, knowledge acquirement and information extraction, nowadays data mining has becomes one of the most front research direction in the international realms of information-based decision making.

Many methods have been proposed to construct the networks; the most important methods are destructive, constructive, and genetic algorithms [2, 33].

Constructive learning of neural networks adds nodes or links to the structure of the network during training. Generally, it starts with a network with no hidden units, which is trained for a period. Then with no changing in the presenting weights, more new hidden nodes are added to the network, the training starts again, and so on. Many variations are feasible, including different patterns of links and schemes for freezing and melt weights. The most familiar constructive learning algorithm is cascade correlation [1, 28, 31] of which many variations are possible [3, 4]. Various other constructive algorithms are summarized in [5]. An approach for adaptively and unconventionally constructing a multilayer feed forward neural network (FNN) is introduced in [29]. Regression problems has been demonstrated as a review for the constructive learning algorithms in feedforward neural networks in [27]. Constructive learning algorithms were used to handle multi-category classification with convergence to zero classification errors [2].

Extracting comprehensible rules from neural networks is still needing efforts because that multi-layered artificial neural network are often regarded as ”black boxes”, which map input data into a

DOI : 10.5121/ijaia.2013.4102 

class through a number of mathematically weighted connections between layers of neurons. In order to bear this limitation, the hypothesis generated by artificial neural networks could be transferred into a more comprehensible representation; these con- version methods are known as rule extraction algorithms. Many researchers dealt with this problem such as in [8, 9, 13, 14, 15, 21, 34]. They introduce methods for discovering M-of-N rules from trained artificial neural network for the standard three layered feed forward networks.

In [22] an approach named REFNE is proposed to improve the comprehensibility of trained artificial neural network ensembles that performs classification tasks. REFNE utilizes the trained ensembles to generate instances and then extract symbolic rules from those instances. A novel strategy using genetic algorithms to search for symbolic rules in a trained artificial neural network is introduced in [10]. Many approaches have been introduced via trained artificial neural network to extract accurate and comprehensible rules from databases based on genetic algorithm [45, 24, 25]. These methods do not modify the training results.

Genetic Algorithm is a search tool which is usually employed when the search-space in question is large and rather unknown. Genetic Algorithms are stimulated by the means of natural choice where stronger individuals are possible the winners in a competing environment. In our work, we refine GA to deal with the complex topology, independently from how complex the topology is, to extract a set of highly accurate, comprehensible and simple rules [6, 7].

Earlier approaches based on an extensive analysis of network links and output values have already been confirmed to be obdurate in that the scale-up issue increases exponentially with the number of nodes and links in the network. A novel approach using genetic algorithms to search for comprehensible rules in a trained neural network is demonstrated in this paper. Preliminary experiments concerning classification are reported here, with the results representing that our proposed approach is winning in extracting rules. While it is accepted that additional work is required to persuasively demonstrate the dominance of our approach over others, there is nonetheless enough novelty in these results to justify early distribution.

For our experimentation, we will use a wide set of data sets which are commonly used in classification tasks. To check the results obtained, we compare them with some well-known methods of classification such as Bagging with a tree based approach, c4.5 with reduced error pruning (REP), Nave Bayesian, and RBF Networks.

A wide variety of methods are now available, recently reviewed in [11, 12, 23, 15]. Tickle et al [38] revisits the Andrews [11] classification of rule extraction methods and emphasise distinction between decompositional and pedagogical approaches. Rule extraction methods typically start by finding a smallest network, in terms of number of hidden units and overall connectivity. Setiono [39] for example adds penalty terms to the error function to bias back propagation like training towards such sparse networks. The next simplification, the key feature of the method, is to quantize or cluster the hidden unit activations. It is then promising, link by link, to extract combinations of inputs which will activate each hidden node, singly or together and thus output generates rules. This unit by unit analysis characterizes the decompositional approach. Although it can yield exact representations, the computational time may grow exponential with number of inputs (attributes) as noted by Tickle et al. [15] for decompositional algorithms such as Subset and KT. Taha and Ghosh [40] suggest for binary inputs such as our data generating a truth table from the inputs and simplying the resultant Boolean function. But this simplification is itself combinatorially nasty and thus the method works only for small networks. They also refine the Liu and Setiono [41] methods using linear programming.

The rest of the paper is organized as follows. The structure of the networks is presented in Section 3. In Section 4, rules from trained artificial neural network based on genetic algorithm

will be discussed. The dataset used its representation of the neural networks, illustrative example for rules extraction algorithm to some public databases and comparison to other methods presented in Section 5. Description of the results and discussion will be presented in Section 6. Finally, in Section 7 we will outline the conclusions of this paper.

## 2. RULE EXTRACTION PHASE

In this section we will introduce an approach which extracts accurate and comprehensible rules from databases via trained artificial neural network using genetic algorithm. This method does not modify the training results. After using constructive learning we have the following ANN topology which is shown in Figure 1. It consists of *L* nodes in the input layer, *h* nodes in the hidden layer and *n+ 1* node in the output layer. Also, two groups of weights can be obtained. The first set, $W_{ij}$ includes the links between the input node *i* and the hidden node *j*. The second set, $V_{jk}$ includes the links between the hidden node *j* and the output node *k*. A sigmoid function as an activation function has been used in the hidden and output nodes of the ANN.

The total input to the *j*th hidden node, $IHN_j$, is given by

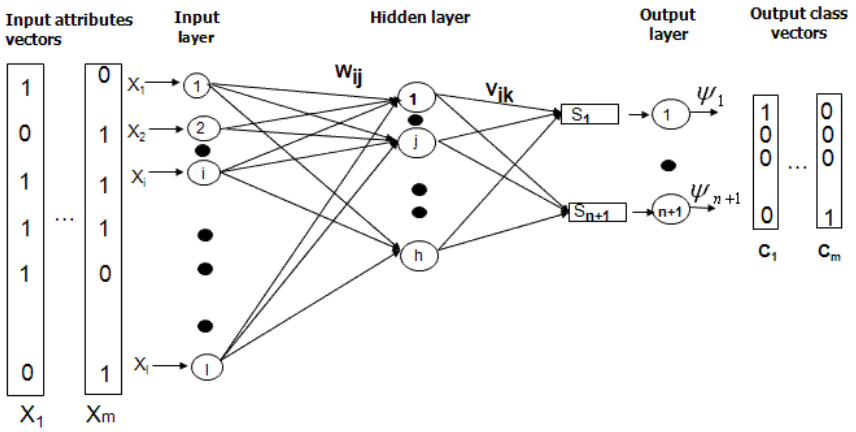


Figure 1: The structure of the network.

$$\mathrm{IHN_j} = \sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{x_i}\, \mathrm{W_{ij}} \tag{1}$$

The output of the *j*th hidden node,$OHN_j$, is given by

$$OHN_j = \frac{1}{1 + e^{-IHN_j}} \tag{2}$$

The total input to the *k*th output node, $ION_k$, is given by

$$ION_k = \sum_{j=1}^{J} V_{jk} \frac{1}{1 + e^{-IHN_j}} . \quad (3)$$

So, the final value of the *k*th output node, $\psi_k$, is given by

$$\psi_k = \frac{1}{1 + e^{-ION_k}} \quad (4)$$

The function, $\psi_k = f(x_i, W_{ij}, v_{jk})$ is an exponential function in $x_i$ since $W_{ij}$, $V_{jk}$ are constants. Its maximum output value equal one.

We have to note that an input vector, $X_m$, belongs to a $class_k$ **iff** $\psi_{k \epsilon C_m} = 1$ and all other elements in $C_m = 0$.

Consequently, for extracting relation (rule) between the input attributes, $X_m$ relating to a specific $class_k$ one must find the input vector, which *maximizes* $\psi_k$. This is an optimization problem and can be stated as:

$$Maximizes[\psi_k(x_i)] \quad (5)$$

subjected to: $x_i \epsilon$ (0 or 1).

Since the objective function $\psi_k(x_i)$ is nonlinear, it is a nonlinear integer optimization problem, and the constrains are binary so. Such problems with very large search spaces and strong non-linearity can be solved using GA. To use the general schema of GA first of all it is necessary to determine a representation of individual, and the way of coding it in the chromosome. Genetic operators (crossover and mutation) act on selected individuals. Crossover acts with assumed relatively high probability. Next, every attribute is mutated with assumed, small probability. The final effect of population proceeding creates the next generation.

## 2.1. Constructive Learning Algorithm

Typically, constructive learning begins with a network with one hidden unit; one or more new hidden units are added to the network, training resumes, and so on. The major steps of our algorithm can be shortened as follows.

1. Input the training set of patterns.
2. Initial set of weights and thresholds are initialized randomly from the interval [-1, 1].
3. Start learning with one unit in the hidden layer.
4. Update the weights and thresholds to minimize the objective function by any optimization algorithm.

5. Terminate the network training with this number of hidden units when a local minimum of the objective function has been reached.

6. If the desired accuracy is not reached, increase the number of hidden units with random weights and thresholds and go to step 4, otherwise, go to step 7.

7. Stop.

Since the objective function $\psi_k$(xi) is nonlinear and the constrains are binary so, it is a nonlinear integer optimization problem. The genetic algorithm can be used to solve it since genetic algorithm is a robust search method and is suitable for difficult problems - e.g. problems with very large search spaces and strong nonlinearity. To use the general schema of GA first of all it is necessary to determine a representation of individual, and the way of coding it in the chromosome. Genetic operators (crossover and mutation) act on selected individuals. Crossover acts with assumed relatively high probability. Next, every attribute is mutated with assumed, small probability. The final effect of population proceeding creates the next generation.

## 3. RESULTS & DISCUSSION

The performance of the learning algorithm which is evaluated by the accuracy (%) on the monk's, weather's and breast cancer's problems. Where accuracy is how close a measured value is to the actual (true) value and Precision is how close the measured values are to each other. In other hand, the precision of an object value is a measure of the reliability of the experiment, or how reproducible the experiment is.

The accuracy of an object value is a measure of how closely the experimental results agree with a true or accepted value. Both accuracy and precision are conditions used in the fields of science, engineering and statistics. The accuracy of the classifier will be evaluated using the following parameters:

*Precision* is the percentage of true positives (TP) compared to the total number of cases classified as positive events.

$$Precision = \frac{TP}{TP + FP} \times 100\%, \quad (6)$$

where FP represents false positives. According to Cios and Moore [46],"This measurement is very popular in machine learning and pattern recognition communities". To better understand the performance of the learning algorithm, we will define a number of other measurements. Let us begin by examining a contingency Table 1. Contingency tables contain four variables:

A *true* – positive (TP) occurs when a classifier correctly classified class1.

A *true* – negative (TN) occurs when a classifier correctly classified class2.

A *false* – positive (FP) occurs when a classifier incorrectly classified class1.

A false – negative (FN) occurs when a classifier incorrectly classified class2.

Another measurement of performance, frequently used in conjunction with precision, is accuracy.

Table 1: Contingency table

| Test results | Disorder Present | Disorder Absent | Total |
|---|---|---|---|
| Positive | TP FN | FP TN | TP+FP<br>FN+TN |
| Total | TP+FN | FP+FN | |

Accuracy is the number of correctly classified cases compared to the total number of cases presented to a system. It is defined by the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \quad (7)$$

perhaps the most common measurements in classification problems are sensitivity and specificity.

They are statistical measures of the performance of a binary classification test.

The *sensitivity* measures the amount of real positives which are rightly identified as such (e.g. the percentage of sick people who are identified as having the condition)

The specificity measures the proportion of negatives which are correctly identified (e.g. the percentage of well people who are identified as not having the condition).

They are defined in the following equations:

$$Sensitivity = \frac{TP}{TP + FN}, \quad (8)$$

where a sensitivity of 100% means that the test recognizes all sick people as such.

$$Specifity = \frac{TN}{TN + FP}, \quad (9)$$

where a specificity of 100% means that the test recognizes all healthy people as healthy.

The results which are given in Fig. 2 show the accuracy of our method compared to other classifiers like Bayesian, RBF Network, etc. Results for those classifiers were constructed using Weka experimenter, Weka is a collection of machine learning algorithms for data mining tasks, the algorithms can either be applied directly to a data set or called from your own Java code. Weka contains tools for data pre-processing, association rules, classification, regression, clustering, and visualization. It is also well-suited for developing new machine learning schemes http://www.cs.waikato.ac.nz/ ml/weka/. It is clear in most of the data sets the high performance of our model when compared to other classifiers, except for RBF Network that gave equal accuracy for the weather data set and for monk1 data set and Bagging algorithm gave equal accuracy. In the Monk3 data set, however, there was a significant decrease in accuracy rate compared to Naive Bayesian and RBF Network algorithms. However, in this data set our rule extraction method extracted more simple and comprehensible rules in comparison with other algorithms.

## 4. CONCLUSION

It is well known that the knowledge acquired by ANNs is generally incomprehensible for humans. This reality was a major drawback in this paper, in which ultimately understandable patterns (like classification rules) are very important.

We apply genetic algorithms to extract approximate rules from neural net- works. The genetic algorithm approach described here determines rules better than those found from various decision tree based methods.

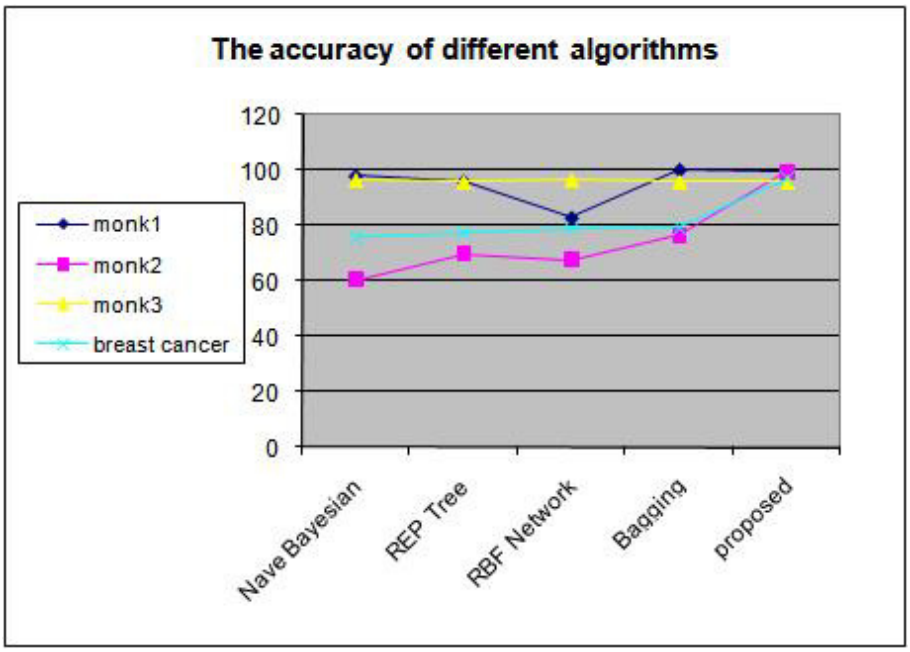


Figure 2: The accuracy of different algorithms.

It is simple to implement, but requires considerable computing resources for the large neural networks of the present paper. As such its greatest value is for determining heuristic rules for medium term use by practitioners who welcome the intuitive description that rules provide.

## REFERENCES

[1] C.J. Giles, D. Chen, G. Sun, H. Chen, Y. Lee, M.W. Goudreau, Constructive learning of recurrent neural networks: Limitations of recurrent cascade correlation and a simple solution, IEEE Trans. Neural Networks 6(1)(1995)829-836.

[2] R. Parekh, J. Yang, V. Honavar, Constructive neural networks learning algorithms for multi-category pattern classification, Technical Report TR95-15, AI Research Group, Dept. of Computer Science, Iowa State University, Tech. Rep. ISU-CS-TR97-06 (3)(1997)1924-1929.

[3] E. Littmann, H. Ritter, Learning and generalization in cascade network architectures, Neural Computation (8)7(1996)1521-1539.

[4] L. Prechelt, Investigation of the casCor family of learning algorithms, Neural Netwrks (10)5(1997)885-896.

[5] F. J. Smieja, Neural network constructive algorithms: Trading generalization for learning efficiency, Circuits, Systems and Signal Processing (12)2(1993)331-374.

[6] D. E. Goldberg, Genetic Algorithm in Search, Optimization, and Ma- chine Learning, Addison-Welsy, 2002.

[7] M. Mitchell, An Introduction to Genetic Algorithms, MIT press, 2005.

[8] R. Setiono, Extracting M-of-N rules from trained neural networks, IEEE Trans. on Neural Networks (11)2(2000)512-519.

[9] F. Wotawa, G. Wotawa, Deriving qualitative rules from neural networks-a case study for Ozone forecasting,AI Commun. (14)1(2001)23-33.

[10] A. Narayanan, E. Keedwell, D. Savic, Data Mining Neural Networks with Genetic Algorithms, 2008.

[11] M. Goebel, L. Gruenwald, A survey of data mining and knowledge discovery software tools, SIGKDD Explorations (1)1(1999)20-33.

[12] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann, 2006.

[13] W. Craven, Extracting Comprehensible Models from Trained Neural Networks, PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison, 1996.

[14] G. Towell and J. Shavlik, The extraction of refined rules from knowledge-based neural networks, Machine Learning (13)1(1993)71-101.

[15] R. Andrews, J. Diederich, A. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge Based System (8)6(1995)373-389.

[16] J. Neumann, Classification and Evaluation of Algorithms for Rule Ex- traction from Artificial Neural Networks, PhD. summer project, Uni- versity of Edingburgh, 1998.

[17] A. Darbari, Rule Extraction from Trained ANN: A Survey, Technical Report, Department of Computer Science, Dresden University of Tech- nology, Dresden, Germany, 2001.

[18] R. Krishnan, G. Sivakumar, P.Bhattacharya, A search technique for rule extraction from trained neural networks, Pattern Recognition Let- ters (20)3(1999)273-280.

[19] H. Tsukimoto, Extracting rules from trained neural networks, IEEE Trans. Neural Networks (11)2(2000)377-389.

[20] R. Setiono, Extracting rules from neural networks by pruning and hidden-unit splitting, Neural Computation (9)1(1997)205-225.

[21] W. Craven, Extracting Comprehensible Models from Trained Neural Networks, Ph.D. Dissertation, Univ. Wisconsin, 1996.

[22] Z. H. Zhou,Y. Jiang , S. F. Chen, Extracting symbolic rules from trained neural network ensembles, AI Commun. (16)1(2003)3-15.

[23] U. Markowska-Kaczmar and M. Chumieja, Discovering the mysteries of neural networks, International Journal of Hybrid Intelligent Systems (1)3,4(2004)153-163.

[24] U. Markowska-Kaczmar, The influence of prameters in evolutionary based rule extraction method from neural network, in: Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA 2005), 8-10 September, Wroclaw, Poland. IEEE Computer Society, 2005, pp.106-111.

[25] R. T. Santos, j. C. Nievola, A. A. Freitas, Extracting comprehensible rules from neural network via genetic algorithms, in: Proc. 2000 IEEE Symp. On Combinations of Evolutionary Computation and Neural Net- works, San Antonio, TX, USA. 1, 2000, pp. 130-139.

[26] A. Duygu Arbatli and H. Levent Aki, Rule extraction from trained neu- ral networks using genetic algorithms, Elsevier (30)3(1997)1639-1648.

[27] T. -Y. Kwok, D. -Y. Yeung, Constructive algorithms for structure learn- ing in feedforward neural networks for regression problems, IEEE Trans. on Neural Networks (8)3(1997)630-645.

[28] S. E. Fahlman, C. Lebiere, The Cascade-Correlation Learning Archi- tecture, Morgan Kaufmann (2)(1990)524-532.

[29] L. Ma, K. Khorasani, A new strategy for adaptively constructing multi- layer feedforward neural networks, Neurocomputing (51)(2003)361-385.

[30] L. Ma, K. Khorasani, New training strategies for constructive neu- ral networks with application to regression problems, Neural Networks (17)4(2004)589-609.

[31] M. A. Potter, A genetic cascade-correlation learning algorithm, in: In- ternational Workshop on Combinations of Genetic Algorithms and Neu- ral Networks,IEEE Computer Society Press, 1992, pp.123-133.

[32] S. Salcedo-sanz, C. Bousoo-calzn, A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications, Applied Intellegent (22)3(2005)207-217.

[33] J. Branke, Evolutionary algorithms for neural network design and train- ing in: Proceedings of the First Nordic Workshop on Genetic Algorithms and its Applications, 1995, pp.145-163.

[34] J.C. Hsieh, P.C. Chang, S.H. Chen, Integration of genetic algorithm and neural network for financial early warning system: An Example of Taiwanese Banking Industry, in: Proceedings of First International Conference on Innovative Computing Information and Control (I), 2006, pp.562565.

[35] X. Yao, A review of evolutionary artificial neural networks, Int. J. Intell. Syst. 8(4)(1996)539567.

[36] P.P. Palmes, T. Hayasaka, S. Usui, Mutation-based genetic neural net- work, IEEE Trans. on Neural Networks (16)3(2005)587-600.

[37] C. R. Reeves, J. E. Rowe, Genetic algorithms - principles and perspec- tives, A guide to GA theory, Kluwer Academic Publishers, 2003.

[38] A. Tickle, R. Andrews, M. Golea, J. Diederich, The truth is in there: di- rections and challenges in extracting rules from trained artificial neural networks, IEEE Trans. on Neural Networks (9)(1998)1057-1068.

[39] R. Setiono, A penalty-function approach for pruning feedforward neural networks, Neural Computation (9)1(1997)185-204.

[40] I. Taha, J. Ghosh, Three techniques for extracting rules from feed- forward networks, Intelligent Engineering Systems Through Artificial Neural Networks (6)(1996)5-10.

[41] H. Liu, R. Setiono, Incremental feature selection, Journal of Applied Intelligence (9)3(1998)217-230.

[42] R. Santos, J. C. Nievola, A. A. Freitas, Extracting comprehensible rules from neural networks via genetic algorithms, in: Proceedings of the 2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks (ECNN-2000), San Antonio, TX, USA, 2000.

[43] E. Keedwell, A. Narayanan, D. Savic, Creating rules from trained neural networks using genetic algorithms, International Journal of Computers, Systems and Signals (IJCSS)(1)1(2000)30-42.

[44] G. Bologna, A model for single and multiple knowledge based networks, ELSEVIER Artificial Intelligence in Medicine (28)2(2003)141-163.

[45] J. Huysmans, B. Baesens, J. Vanthienen, Using rule extraction to im- prove thed comprehensibility of predictive models, Technical Report KBI 0612, Katholieke Universiteit Leuven, Department of Decision Sci- ences and Information Management, Leuven, Belgium(43)(2006)1-55.

[46] K. J. Cios, G. William, Uniqueness of medical data mining, Artificial Intelligence in Medicine (26)1(2002)1-24.

[47] S. B. Thrun, et al., The Monk's problems - A performance compari- son of di▪erent learning algorithms, Department of Computer Science, Carnegie Mellon University, CMU-CS-91-197, 1991.

[48] H. W. William, O. L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc. of the National Academy of Sciences, U.S.A. (87)(1990)9193-9196.

## Authors

Safa S. Ibrahim[1] her Bachelor Science degree in computer science in 2004 at Assiut University, Faculty of Science, Dept. of Mathmatics, Egypt and received her Master in 2010 at Assiut University – Egypt in Data Mining.

Mohammed Bamatraf is currently an assistant professor at Hadhramout University of Science and technology, Yemen, Faculty of Engineering, Department of Computer Engineering, teaching several computer science subjects. He received his B.Sc (computer Science) from Poona University, India, his M.sc (computer Science) from Osmania University, India, and his PhD from Assiut University, Egypt. His Doctoral thesis was about modified data mining techniques and its application in medical diagnosis and intrusion detection. He published several papers in local as well as international conferences. His research areas of interest includes: data and network security, medical informatics, data mining and machine learning, and bioinformatics. His research activities are currently focused on the application of Bioinformatics and Machine Learning data and network security