Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 8, August - 2016

Rule Extraction From Training Artificial Neural Network Using Variable Neighbourhood Search For Wisconsin Breast Cancer

Yusuf UZUN
Seydişehir Vocational School,
Necmettin Erbakan University,
Konya, Turkey
yuzun@konya.edu.tr

Hüseyin ARIKAN
Seydişehir A.C.M.F,
Necmettin Erbakan University,
Konya, Turkey
harikan@konya.edu.tr

Gülay TEZEL
Computer Engineering Faculty,
Selcuk University,
Konya, Turkey
gtezel@selcuk.edu.tr

***Abstract*— Artificial neural network (ANN) may achieve successful classification value, but obtained results sometimes can be unintelligible and may not be interpreted. In literature, different rule extraction methods from trained ANN are applied to overcome this problem. This study presented a new method of extraction correct and intelligible rules from Trained ANN using variable neighbourhood search (VNS) metaheuristic method. Since the VNS method is independent from the ANN, it does not modify the results of ANN. Real time Wisconsin Breast Cancer (WBC) data was examined for determining the feasibility of the suggested method. The VNS method was used to obtain the best fitness values belong to input attributes, Id, which was maximized the fitness function S_r of output node *r*. The suggested method was the computational evaluated on investigated data sets and obtained results from the suggested approach indicated that had a potential to produce accurately and cored rules. The proposed method obtained accuracy value 98.97% for WBC dataset.**

Keywords—variable neighbourhood search; artificial neural network; optimization

I. INTRODUCTION

Breast cancer is a malignant tumour that develops from breast tissue. Some clinical risk factors for improving breast cancer include females, obesity, inactivity, alcohol, radiation, menstruation at early age, old age, and genetic, etc. [1].

Any study with regard extraction rules from TANN using VNS has not been observed in literature searches. This paper was focused on the rule extraction from previously trained ANN using VNS. ANN used in the study was trained using activation function.

There are different studies improved for the rules extraction from ANN. Zhang et al. proposed a method to extract rules extraction from a GA-pruned ANN, which is called RulExt algorithm [2]. Fukumi and Akamatsu developed a method to extract rules from TANN using the evolutionary algorithms (EA) that had a deterministic mutation [3]. Zhenya et al. suggested rules extraction from fuzzy neural network using particle swarm optimization [4]. Dorado et al. proposed a rule extraction from ANN using genetic algorithm programming [5]. Elalfi et al. presented an approach for extraction correct and intelligible rules from trained ANN using a genetic algorithm [6]. Tokinaga et al. presented extraction methods using the genetic programming (GP) technique to generate smart and descriptive assessment systems [7]. Hruschka et al. proposed a clustering method to extract rules from ANN for the problem of classification [8]. Kahramanlı and Allahverdi developed a technique for rules extraction from ANN using artificial immune systems (AIS) [9]. Özbakır et al. presented TACO algorithm to extract rules from TANN [10]. Özbakır and Delice developed binary particle swarm optimization (BPSO) method for extracting rule from TANN [11]. Kasiri et al. suggested a fuzzy extracting rule method from TANN using genetic algorithm for check and parameter forecast [12]. Marghny developed extracting rules for TANN using genetic algorithms [13]. ElAlami proposed destructive technique for extracting rule from a trained ANN [14]. Kamruzzaman et al. presented a method to extract rules from TANN [15].

Rule extraction from TANN is an important inference technique to interpret results. A rule is generally represented by a structure as "if … then …". An important disadvantage of many ANN is their lacking explanation ability [16]. Since ANN has been successfully used in many studies, their outcomes incomprehensible, the knowledge dispersed state over the activation functions and the connections between neurons [17]. Keedwell et al., suggested a method to extract rules from TANN using genetic algorithm [22]. As a result, researchers are interested in improving accuracy and comprehensible representation by rules extraction from TANN to solve this problem.

The work was arranged as follows: In Section 2, materials and methods method was explained. In Section 3, experimental study was presented. In Section 4, evaluations of the study were described. In Section 5, conclusions were presented.

II. MODEL

A. *Artificial Neural Network (ANN)*

ANN is an experimental method inspired by neural networks contained within the brain and a very popular method used in classification problems. In this study, the backpropagation algorithm was preferred to TANN. The architecture of ANN with one hidden layers indicated in Figure 1. ANN consists of an input, an output and at least one hidden layers that were weighted interconnected with each other. The number of nodes in the input and output layers are determined according to the problems. In this study, the ANN was trained using the binary coded vectors belonging to input features and output classes, as input and output desired dataset respectively. The training process of ANN was continued until the desired output value was reached.

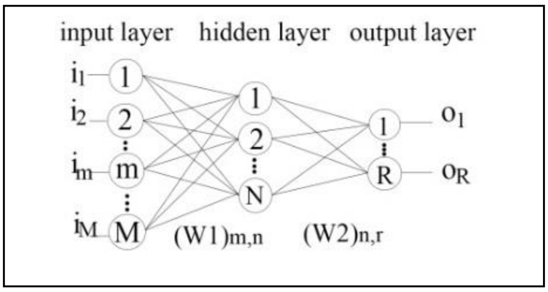

Fig. 1. *The ANN structure.*

Once training process was completed, two weight vectors were obtained. $W1_{m,n}$ is the weight vector between input and hidden layers' nodes, and $W2_{n,r}$ is the weight vector between hidden and output layers. Also sigmoid function was preferred as activation function in hidden and output layers' nodes. Using $W1_{m,n}$ and $W2_{n,r}$ weight vectors, the value of the *r*th output node S_r was provided by;

$$S_{\mathrm{r}} = \left\{ \frac{1}{1+e^{-\left[\sum_{n=1}^{N}(W2)_{\mathrm{n,r}}\frac{1}{1+e^{-\left[\sum_{m=1}^{M} i_m (W1)_{\mathrm{m,n}}-\theta n\right]}}\right]}} \right\} \quad (1)$$

Where *M, N* and *R* are the number of input, hidden and output layers' nodes. θ_n is threshold for *n*th neuron of hidden layer. The nonlinear exponential function, S_r is the fitness function of VNS based on the sigmoid function.

The maximum achievable value of this function is 1. Finally, it was required to achieve the input vector that maximizes the function S_r for rules extraction from TANN [6]. The encoded best values were used to obtain a rule of class$_r$;

$$Max\ S_{\mathrm{r}}(i_{\mathrm{m}}) = \left\{ \frac{1}{1+e^{-\left[\sum_{n=1}^{N}(W2)_{\mathrm{n,r}}\frac{1}{1+e^{-\left[\sum_{m=1}^{M} i_m (W1)_{\mathrm{m,n}}-\theta n\right]}}\right]}} \right\} \quad (2)$$

i_m= 0 or 1

B. *Variable Neighborhood Search (VNS)*

VNS is a method that was developed to solve optimization problems which were based on the exchange of neighborhood structures in a local search process [19].

The VNS starts operating on a starting values *x* and a series of neighborhoods N_k, ($k=1,2,\ldots,k_{max}$). During each iteration, a random value *x´* is calculated according to the *k*th neighborhood, $N_k(x)$. Later, a local search method is implemented to achieve a second value *x´´* from the value *x´*. If the value *x´´* is better than the starting value *x*, the *x´´* value is updated and the transaction continues with the first neighborhood $N_1(x)$, otherwise the operations are repeated taking into account the next neighborhood, N_{k+1}. The obtained final value to the process found would be a local best value according all neighborhood values of the VNS [20].

Pseudo code of the basic VNS algorithm is given bellow;

1. Initialization. Determine the series of neighborhoods structures N_k, for ($k=1,\ldots,k_{max}$), that will be used in the VNS; build an starting values *x*; select a stopping criteria;

2. Repeat the below process steps until the stopping criteria is provided:

   a. Assign the value, k =1;

   b. Repeat the below process steps until $k = k_{max}$;

      i. Generate a random solution x´ from the kth neighborhood of *x* ($x´ \epsilon N_k(x)$);

      ii. If this solution is better than the current starting value, *x=x´*, and continue the search with the first neighborhood N_1 (k= 1), otherwise, assign the value $k=k$ +1.

Table. I. WBC dataset

ID	Attribute	Data	Number of binary variables	Sub-ranges
A_1	Clump Thickness	Continuous	3	(-; 4], (4; 7], (7,-)
A_2	Cell size uniformity	Categorical	3	(-; 4], (4; 7], (7,-)
A_3	Cell shape uniformity	Continuous	3	(-; 4], (4; 7], (7,-)
A_4	Marginal adhesion	Continuous	3	(-; 4], (4; 7], (7,-)
A_5	Single epi cell size	Continuous	3	(-; 4], (4; 7], (7,-)
A_6	Bare Nuclei	Continuous	3	(-; 4], (4; 7], (7,-)
A_7	Bland Chromatin	Continuous	3	(-; 4], (4; 7], (7,-)
A_8	Normal Nucleoli	Continuous	3	(-; 4], (4; 7], (7,-)
A_9	Mitoses	Continuous	3	(-; 4], (4; 7], (7,-)
O	Class	Categorical	2	{Benign}, {Malignant}

C. *The Dataset*

The WBC dataset obtained from UCI Machine Learning Repository was used to extract rules from trained ANNs [21]. This dataset with two classes consisted of 699 instances and contained 9 attributes. There were 16 missing values in attributes, which were discarded from dataset and remaining 683 cases were used. The ID number (A_1, A_2..., A_9), name, data type (continuous/categorical) of attributes in WBC dataset as well as the number of binary variables and subintervals of attributes are shown in Table 1. For the sub-ranges, were performed Simple Partition and Simple Binning methods using Weka 3.7 software.

The values of categorical attributes were converted into binary-coded values that are shown in Tab. 2 to present ANN's input layer. Following methods were used for coding in binary format of the dataset [6].

It was assumed that the data set contained the Z attributes. Each attribute of the dataset, A_z (z = 1, 2,...,Z), was encoded into d_z length binary sub-string as $\{i_1, i_2, \ldots, i_{d_z}\}$. If attribute A_z belongs to sub-string $\{i_1, i_2, \ldots, i_{d_z}\}$, the element i_m = 1 and values belong to all the other elements are zeros. So that, the total number of available nodes in the input layer of ANN, M, could be obtained [6].

Vectors of the input attributes, I_d, to the input layer belonging to the ANN can be formatted as $I_d = \{i_1, i_2, \ldots, i_m \ldots i_M\}_d$, where d = (1, 2, . . . , D), where D is the number of training data given to the input layer [6].

Vector of the output class, O_r (r = 1, 2, . . . , R), was encoded into R length binary sub-string as $\{o_1, o_2, \ldots, o_r\}$, where R is the total number of the different probable classes. If the output vector belongs to O_r then the value of the o_r is equal to 1 and values of the remaining elements in the O_r vector are zeros [6]. In this study, the number of output layer's nodes (R) was determined as two because of the dataset with two classes.

The class attribute *Class* of WBC dataset has Boolean value. A value of 0 indicates that patient is benign. If the value 1 indicate that patient is malignant.

Table. II. An example of the converted binary-coded values

A_1			A_2			A_3			A_4			A_5			A_6			A_7			A_8			A_9			O	
i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}	i_{13}	i_{14}	i_{15}	i_{16}	i_{17}	i_{18}	i_{19}	i_{20}	i_{21}	i_{22}	i_{23}	i_{24}	i_{25}	i_{26}	i_{27}	o_1	o_2
0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1
0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0

D. *Performance Evaluation and Experimental Study*

Accuracy is used to measure classification ability of output class. In this study, equation 3 was used accuracy value for performance evaluation of proposed algorithm. Accuracy for each value of output class is calculated as follow.

$$Acc. = \left\{\frac{Total\ number\ of\ correctly\ classification\ outputs}{Total\ number\ of\ outputs}\right\} \quad (3)$$

In this study, a novel approach to extract rules from TANN using WBC dataset was proposed. VNS method was preferred for optimization of fitness function so that the most effective and accurate rules could be extracted. The aim is to obtain the optimum string values by maximizing the fitness function, S_r. The algorithm of proposed model was shown in detail in bellow. In addition, the flowchart of the approach used to obtain the optimum solution values by maximizing the S_r is shown in Fig. 2. The process steps of the proposed approach are given bellow;

1. Separate input and output vectors from the dataset. Data converts binary coded values
2. Train the ANN using input and output vectors with binary-coded
3. Generate weight vectors ($W1_{m,n}$ and $W2_{n,r}$) from the trained ANN.
4. Initialize set parameters. Set as fitness function S_r shown in equation 2.
5. Create an initial population i. Determine the series of neighborhood structures N_k, for (k=1,…,k_{max}); select the stopping criteria;

6. Repeat the below process until the stopping criteria is provided.

   6.1 Set k=1; Maximize the fitness function $S_r(i_d)$; Repeat the below process steps until $k = k_{max}$;

   a. Generate a random solution $x´$ from the kth neighborhood of x ($x´ \epsilon N_k(x)$);

   b. If this solution is better than the current initial solution, $x=x´$, and continue the search with the first neighbourhood N_1 (k= 1), otherwise, assign the value k=k +1;

   6.2 Evaluate the fitness function according to initial population.

7. Maintain all the solutions of bigger S_r values than an indicated threshold value.

8. Rule extraction: Find fitness solutions which have the largest value of the solutions produced. Add these deleted solutions to the rule list.

9. Convert the solutions to the linguistic rules (if-then).

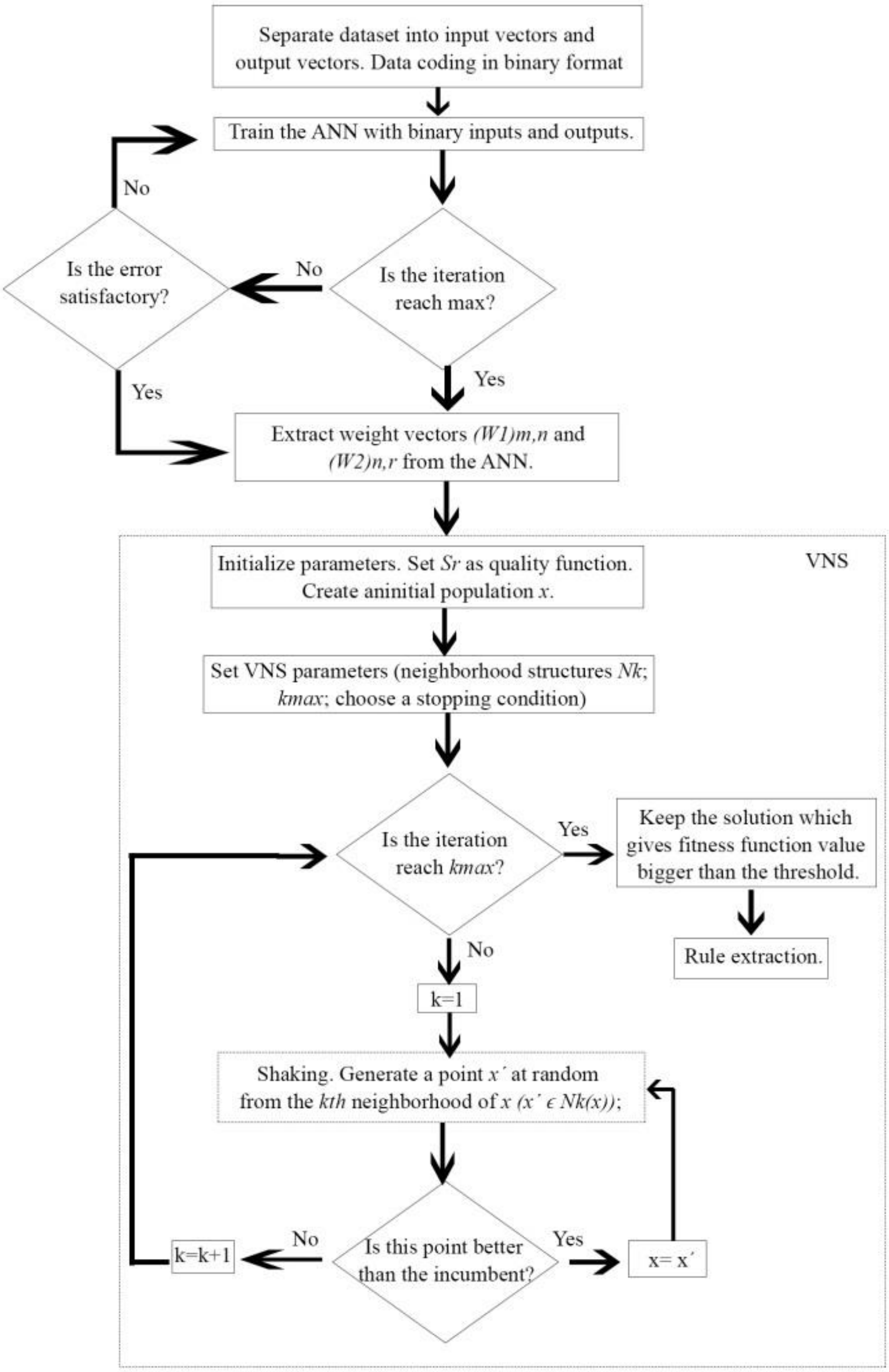

Fig. 2.*Flowchart for proposed approach*

III. Evaluation of the Study

In this study, WBC dataset with 683 input vectors (I_d) was used to extract rules with VNS from TANN. The optimum ANN structure with minimum fitness value was obtained with 7 hidden nodes, learning coefficient as 0.001, momentum coefficient as 1 after the experiments. The momentum parameter is used to prevent the system from converging to a local minimum. Maximum series of neighborhood structures N_k (k_{max}) as 2. The first neighborhood structure is used in the study is random two interchange. The second neighborhood structure is randomly crossing to exchange. The stopping criteria of ANN are the number of iteration and minimum error rate. These values were selected as 20000 and 0.01 respectively.

The number of nodes in the input and output layers are 27 and 2 respectively. When VNS algorithm was applied to TANN, it was seen that number of rules were changing depending on the threshold value.

The extracted rules and the refinement rules belonging to output class taking into account the fitness values obtained from WBC dataset are presented in Tables 3 and 4.

The binary codes of related attributes were used to extract rules. If more than one attribute are available in the same time. The attributes must combine with “OR” operator. In the rule extraction process. the presence of “OR” operator isn’t desirable situation because it is called as uncertainty conjunction. So, the small number of “OR” operator shows the consistency and efficiency of rule extraction method.

When are examined extracting example rules in Tables 3 and 4, it isn't appeared any uncertainty conjunction in the rules. Therefore, same values of the extracted rules and refinement rules columns. When a rule which is not included in the rules Extracted is input, will be achieved "It does not belong to any class" output. These results obtained by this algorithm show that the method is consistently and efficiently.

Table. III. Examle rules are extracted for the class 0

Rule no.	***Fitness***	***Extracted rules***
1	0.99968	If Clump Thickness = (-; 4] and Cell size uniformity= (-; 4] and Cell shape uniformity = (-; 4] and Marginal adhesion = (-; 4] and Single epi cell size = (-; 4] and Bare Nuclei = (-; 4] and Bland Chromatin = (-; 4] and Normal Nucleoli = (-; 4] and Mitoses= (-; 4] is Class= “Benign”
2	0.99961	If Clump Thickness = (-; 4] and Cell size uniformity= (-; 4] and Cell shape uniformity = (-; 4] and Marginal adhesion = (-; 4] and Single epi cell size = (-; 4] and Bare Nuclei = (-; 4] and Bland Chromatin = (-; 4] and Normal Nucleoli = (-; 4] and Mitoses= (-; 4] is Class= “Benign”
3	0.99926	If Clump Thickness = (-; 4] and Cell size uniformity= (-; 4] and Cell shape uniformity = (-; 4] and Marginal adhesion = (-; 4] and Single epi cell size = (-; 4] and Bare Nuclei = (-; 4] and Bland Chromatin = (-; 4] and Normal Nucleoli = (-; 4] and Mitoses= (-; 4] ise Class= “Benign”

Table. IV. Examle rules are extracted for the class 1

Rule no.	***Fitness***	***Extracted rules***
1	0.99999	If Clump Thickness = (7; -) and Cell size uniformity= (-; 4] and Cell shape uniformity = (7; -) and Marginal adhesion = (-; 4] and Single epi cell size = (-; 4] and Bare Nuclei = (7; -) and Bland Chromatin = (7; -) and Normal Nucleoli = (7; -) and Mitoses= (7; -) is Class= “Malignant”
2	0.99999	If Clump Thickness = (7; -) and Cell size uniformity= (7; -) and Cell shape uniformity = (7; -) and Marginal adhesion = (7; -) and Single epi cell size = (7; -) and Bare Nuclei = (-; 4] and Bland Chromatin = (7; -) and Normal Nucleoli = (7; -) and Mitoses= (7; -) is Class= “Malignant”
3	0.99999	If Clump Thickness = (4; 7] and Cell size uniformity= (-; 4] and Cell shape uniformity = (-; 4] and Marginal adhesion = (-; 4] and Single epi cell size = (-; 4] and Bare Nuclei = (-; 4] and Bland Chromatin = (-; 4] and Normal Nucleoli = (-; 4] and Mitoses= (4; 7] is Class= “Malignant”

Table. V. Performance evaluation between suggested and other methods

Methods	***Accuracy Value. %***	***Reference***
VNS	**98.97**	Proposed approach
MTACO-Miner	97.98	Tripathy et. al.
TACO-Miner	97.71	Özbakır et. al.
SVM	96.3	Martens et.al.
RMR	95.92	Thabtah et. al.
Ant-Miner	95.61	Smaldon et. al.

The accuracy value that is achieved by proposed method is shown in Table 5. Also. It is shown accuracy value with regard other method for WBC dataset. Average number of rules for approximately 0.99925 threshold values is shown in Table 5. The classification accuracy values of extracted rules for proposed and other methods are shown in Table 5. The suggested method obtained accuracy value 98.97 % using equation 3 in extracting rules from TANN.

IV. Conclusions

In this study, it was used a new method to extract accurate classification rules from trained ANN using VNS metaheuristic method on the WBC datasets. The proposed method did not perform any of approach, review and information extraction to the fitness function. The computational results showed success of the proposed VNS method to extract understandable rules having high accuracy rates for experimental studies.

It appears that used at least 7 of the 8 input attributes in the rules that specified in Tabs. 3 and 4. This shows that extracted rules are the stronger the stability and interpretability. In addition, the concept of uncertainty that specifying the operator "OR" in the extracted rules was not observed. Thus, using this algorithm can be achieved decision support information which has high accuracy on large data sets.

As shown in Table 5, experiment studies have shown that the proposed algorithm is able to produce accurate and efficient classification rules. This is a quick decision making on the operations performed and accurate results are assure obtaining.

Acknowledgment

This work was supported by Necmettin Erbakan University and Selcuk University Scientific Research Projects (BAP) Coordinators hips.

References

[1] J. M. Jerez-Aragones, J. A. Gomez-Ruiz, G. Ramos-Jimenez, J. MunozPerez and E. Alba-Conejo, “A combined neural network and decision trees model for prognosis of breast cancer relapse,” Artificial Intelligence in Medicine, 27(1), pp. 45–63, 2003.

[2] Z. Zhang, Y. Zhou, Y. Lu and B. Zhang, "Extracting Rules from A GA-pruned Neural Network, Systems, Man, and Cybernetics,” IEEE International Conference, 3, 1996.

[3] M. Fukumi, and N. Akamatsu, “Rule Extraction from Neural Networks Trained Using Evolutionary Algorithms with Deterministic Mutation,” Proceedings IEEE World Congress on Computational Intelligence, IEEE International Joint Conference on Neural Networks, 1998.

[4] H. Zhenya, W. Chengjian, Y. Luxi, G. Xiqi and Y. Susu, “Extracting Rules From Fuzzy Neural Network By Particle Swarm Optimisation,” IEEE, 0-7803-4869-9/98, 1998.

[5] J. Dorado, J.R. Rabunal, D. Rivero, A. Santos and A. Pazos, “Automatic Recurrent ANN Rule Extraction with Genetic Programming,” In Proceedings of The 2002 International Joint Conference on Neural Networks IJCNN'02, pp. 1552-1557, 2002.

[6] E. Elalfi, R. Haque, and M.E. Elalami, “Extracting Rules from Trained Neural Network Using GA for Managing E-business,” Applied Soft Computing, 4, pp. 65-77, 2004.

[7] S. Tokinaga, J. Lu and Y. Ikeda, “Neural Network Rule Extraction by Using the Genetic Programming and Its Applications to Explanatory Classifications,” IECE Trans. Fundamentals, E88-A(10), pp. 2627-2635, 2005.

[8] E.R. Hruschka and N.F.F. Ebecken, “Extracting Rules from Multilayer Perceptrons in Classification Problems: A Clustering-based Approach,” Neuro computing, 70, pp. 384-397, 2006.

[9] H. Kahramanlı and N. Allahverdi, “Rule extraction from trained adaptive neural networks using artificial immune systems,” Expert Systems with Applications, 36, pp. 1513-1522, 2009.

[10] L. Özbakır, A. Baykasoğlu, S. Kulluk and H. Yapıcı, “TACO-miner: An ant colony based algorithm for rule extraction from trained neural networks,” Expert Systems with Applications, 36, pp. 12295-12305, 2009.

[11] L. Özbakır and Y. Delice, “Exploring comprehensible classification rules from trained neural networks integrated with a time - varying binary particle swarm optimizer,” Engineering Applications of Artificial Intelligence, 24, pp. 491-500, 2011.

[12] H. Kasiri, M.S. Abadeh, H. Momeni and A.R. Motavalian, “Fuzzy rule extraction from a trained artificial neural network using genetic algorithm for wecs control and parameter estimation,” Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 1, pp. 635-639, 2011.

[13] H. Marghny, “Rules extraction from constructively trained neural networks based on genetic algorithms,” Journal Neuro Computing, 74, 17, pp. 3180-3192, 2011.

[14] M.E. ElAlami, “Destructive algorithm for rule extraction based on a trained neural network,” International Journal of Computer Applications, 42-21, pp. 0975-8887, 2012.

[15] S.M. Kamruzzaman, Md. Abdul Hamid and A.M. Jehad Sarkar, “ERANN: An algorithm to extract symbolic rules from trained artificial neural networks,” IETE Journal Research, 58-2, 2013.

[16] R. Andrews, J. Diederich and A.B. Tickle, “A survey and critique of techniques for extracting rules from rained artificial neural networks,” Knowledge-Based Systems, 8, pp. 373-389, 1996.

[17] G. Bologna, “Is the worth generating rules from neural network ensembles?,” Journal of Applied Logic, 2, pp. 325-348, 2004.

[18] M.W. Craven and J.W. Shavlik, “Using sampling and queries to extract rules from trained neural networks,” In Machine learning: Proceedings of the 11th international conference, San Francisco, CA, 1994.

[19] N. Mladenovic´, and P. Hansen, Variable neighborhood search. Computers & Operations Research, 24, 1997, pp. 1097-1100.

[20] P. Hansen and N. Mladenovic´, A Tutorial on Variable Neighborhood Search. Report G-2003-46, Les Cahiers du GERAD, HEC Montreal, 2003.

[21] D.J. Newman, S.C. Hettich, L. Blake and C.J. Merz, UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[22] E. Keedwell, A. Narayanan and D. Savic, “Creating rules from trained neural networks using genetic algorithms,” International Journal of Computers Systeming Signals (IJCSS), 1(1), pp. 30–42, 2000b.