Mining Classification Rules for Liver Disorders

Humar Kahramanli, Novruz Allahverdi

***Abstract*—** Nowadays data mining is a very popular technique and has been successfully applied in medical area. Classification is a essential approach in data mining. One of the classification methods is a Artificial Neural Networks. Artificial Neural Network (ANN) generally achieve high accuracy of classification. However, knowledge acquired by ANN is incomprehensible for humans. This fact is causing a serious problem in data mining applications. The rules that are derived from ANN are needed to be formed to solve this problem and various methods have been improved to extract these rules. Selection of the activation function is important in the performance of ANN. Networks with adaptive activation function seem to provide better fitting properties than classical architectures with fixed activation function neurons [1]. In this study, first neural network has been trained with adaptive activation function. Then for the purpose of extracting rules from adaptive ANN which has been trained for classification, OptaiNET that is an Artificial Immune Algorithm (AIS) has been used and a set of rules has been formed for liver disorder.

Keywords—Adaptive Neural Networks, Artificial Immune Systems, Rule Extraction, Liver Disorders.

I. Introduction

There is a lot of various data generated in everyday medical practice. Large databases are often collected containing data of various kind, type, and importance. Such one database can grow very large in number of collected data [2]. The growing amounts of data has made manual analysis by medical experts a tedious task and sometimes impossible. Many hidden and potentially useful relationships may not be recognized by the analyst. The explosive growth of data requires an automated way to extract useful knowledge [3].

Data mining technology has become increasingly important in the field of large databases and data warehouses. Through data mining, interesting knowledge and regularities can be extracted and the discovered knowledge can be applied in the corresponding field to increase the working efficiency and to improve the quality of decision-making [3].

Data mining typically results in a set of rules that can be applied to future events or that can provide knowledge about interrelation-ships among data. This set of rules is most useful when it can be dependably applied to new data [4]. Data Mining is the process of extracting useful and often previously unknown information, patterns, and trends from larges quantities of data, generally stored in databases [5].

Humar Kahramanli is with the Department of Electronic and Computer Education, Selcuk University, Konya, TURKEY. (Corresponding author :+90-332-2233331; fax: 90-332-2422179; e-mail: hkahramanli@ selcuk.edu.tr).

Novruz Allahverdi is with the Department of Electronic and Computer Education, Selcuk University, Konya, TURKEY. (e-mail: noval@ selcuk.edu.tr).

Classification is an important topic in data mining research. Given a set of data records, each of which belongs to one of a number of predefined classes, the classification problem is concerned with the discovery of classification rules that can allow records with unknown class membership to be correctly classified [6].

Medical data often seem to contain a great number and uncertain or irrelevant features. How to extract enough necessary and useful diagnostic rules used to be highly depended on the clinical experience [7]. Medical diagnosis can be considered a classification problem: a record is a given patient's case, predictor attributes are all patient's data (including symptoms, signals, clinical history and results of laboratory tests), and the class is the diagnosis (disease or clinical condition that the physician has to discover, based on the patient's data) [8].

As classification has been a major approach in data mining, neural networks play even more significant role in classification. In data mining, the large number of dimensionality and the huge volume of data make neural networks competitive in classification due to their imperviousness of "the curse of dimensionality" and low computational cost [9].

Neural Networks are synchronous systems what gives possibility to speed up the calculations. Another advantage of neural networks is programming by learning. It cause that there is no need to create complicated mathematical equations, but only general definition of parameters, what make possible to realize the task independently of problem's kind [10]. Neural Networks adjust their internal parameters by performing vector mappings from the input to the output space. The basic target of neural networks is the estimation of the requested function [11]. Artificial Neural Networks are one of the most commonly used classifier technique. The reason for being commonly used is to present some properties such as learning from examples and exhibiting some capability for generalization beyond the training data [12]. Although they may achieve high accuracy of classification, the knowledge acquired by such systems is represented in a large number of numerical parameters and network architectures, in a way that is incomprehensible for humans [13]. This may cause problems in some cases. To solve this problem, researchers are interested in developing a humanly understandable representation for neural networks. This can be achieved by extracting production rules from trained neural networks [14].

Rule extraction techniques are grouped into three approaches named as decompositional, pedagogical, and eclectic. In contrast with the decompositional approach, which analyzes the activation and weights of the hidden layers of the neural network, the pedagogical approach

treats the ANN as a black box and extract rules by only looking at the input and output activations [15, 16]. Pedagogical approach aims at extracting symbolic rules, which map the input-output relationship as closely as possible to the way the ANN understands the relationship. The number of these rules and their form do not directly correspond to the number of weights or the architecture of ANN [17]. Finally, the eclectic approach is characterized by any use of knowledge concerning the internal architecture and/or weight vectors in a trained ANN to complement a symbolic learning algorithm [18].

The aim of this study is to develop a classification rules for diagnosing Liver Disorder. The work is organized as follows: In the second chapter, previous studies related with this study are introduced. In the third chapter framework of this study and related background theory are presented. In the fourth chapter, performance metrics are explained. In the fifth chapter, evaluation methods are described. The results of the experiments and evaluation of these results are presented in sixth chapter. In the final, this paper is concluded.

II. LITERATURE REWIEW

To reveal the information concealed in an ANN, researchers have proposed a number of rule extraction techniques. One of the first rule extraction techniques from neural networks was proposed by Gallant [19]. He was working on connectionist expert systems. In this work, each ANN node represents a conceptual entity. Towell and Shavlik showed how to use ANNs for rule refinement [20]. The algorithm was called SUBSET, which is based on the analysis of the weights that make a specific neuron active. Alexander and Mozer developed a rule extraction method, based on connection weights, that supposes activation functions showing approximately Boolean behavior [21]. Sethi and Yoo developed a rule extraction method based on the connection weights [22]. Lu et al. proposed an approach for rule extraction from ANNs based on the clustering of hidden unit activation values [23]. Setiono and Leow presented the fast method is based on the relevance of hidden units, considering their information gains [24]. Palade et al. presented a method of rule extraction from ANNs that are based on interval propagation across the network, using a procedure of inverting an ANN [25]. Elalfi et al. presented an algorithm for extracting rules from databases via trained ANN using genetic algorithm [26].

Most of the approaches described in the literature have basically two motivations. On the one hand, some authors noticed the need for simplification of neural networks to facilitate the rule extraction process, and are in favor of using specialized training schemes and architectures to perform such task. The assumption underlying these approaches is that neural networks can help the extraction of interesting rules. On the other hand, some papers have proposed algorithms mainly intended to clarify the knowledge encoded in previously trained ANNs [27].

When the related literature is reviewed, it can be seen that a lot of research has been done on the Bupa Liver Disorders database. Gonçalves et al. introduced and evaluated the Inverted Hierarchical Neuro-Fuzzy BSP System (HNFB), a neuro-fuzzy model that was specially created for the task of pattern classification and fuzzy rule extraction [28]. The presented model was tested on several benchmark applications. On BUPA Liver Disorders dataset, HNFB achieved %79.69 classification accuracy. Raicharoen and Lursinsap propose the learning algorithms that do not need any kernel functions [29]. The separability is based on the critical Support vectors (CSV) that are essential to determine the locations of all separating hyperplanes. CSV achieved 97.68% classification accuracy in train and 81.39% classification accuracy in test data. Bagirov and Ugon developed the concept of the max-min separability [30]. They have proposed an algorithm to find this piecewise linear function which separates disjoint point sets by minimizing an error function. On BUPA Liver Disorders dataset, they achieved 89.86% classification accuracy. Cordella et al. proposed a new genetic programming based approach to classification problems [31]. The prototypes of the classes consist of logical expressions establishing conditions on feature values and thus describing clusters of data samples. The proposed method achieved 73.8% classification accuracy.

In our previous work, a new adaptive activation function and a method for extracting rules from trained neural networks have been presented [32]. This study is focused on the problem of extracting classification rules for liver disorders and to form the rule set for diagnosis. Networks with adaptive activation function seem to provide better fitting properties than classical architectures with fixed activation function neurons [33]. Moreover, in the previous work, it has been seen that the network trained with adaptive activation function achieved high classification accuracy. Therefore, the ANN was trained with both adaptive activation function and fixed sigmoid activation function. However, both neural networks with fixed sigmoid and adaptive activation function can't achieve enough classification accuracy. Therefore, in this study we used the neural network model presented by Tezel and Özbay [34], which uses adaptive activation function in hidden layer and fixed sigmoid activation function in output layer. The used adaptive activation function presented by Kahramanli and Allahverdi [32]. The study on rule extraction from trained ANN is based on the work of Elalfi et al. [26] and presents algorithm for extracting rules from neural network using artificial immune systems.

III. STUDY ENVIRONMENT AND BACKGROUND THEORIES

A. Basic Structure of a Neuron

The human nervous system is a very complex neural network. The brain is the central element of the human nervous system, consisting of near 10^{10} biological neurons that are connected to each other through sub-networks. Each neuron in the brain is composed of a body, one axon and multitude of dendrites. The dendrites receive signals from other neurons. The axon can be considered as a long tube, which divides into branches terminating in little endbulbs. The small gap between an end bulb and a dendrite is called a synapse. The axon of a single neuron forms synaptic connections with many other neurons. Depending upon the type of neuron, the number of synapses connections from other neurons may range from a few hundreds to 10^4 [35].

The cell body of a neuron sums the incoming signals from dendrites as well as the signals from numerous synapses on its surface. A particular neuron will send an impulse to its axon if sufficient input signals are received to stimulate the neuron to its threshold level. However, if the inputs do not reach the required threshold, the input will quickly decay and will not generate any action. The biological neuron model is the foundation of an artificial neuron [35].

Artificial Neural Networks (ANNs) denote a set of connectionist models inspired in the behavior of the human brain [36]. An artificial neuron, which the basic element of an ANN consists of three basic components: weights, thresholds, and activation function. An activation function performs a mathematical operation on the signal output. More sophisticated activation functions can also be utilized depending upon the type of problem to be solved by the network [35]. Networks with adaptive activation function seem to provide better fitting properties than classical architectures with fixed activation function neurons [1].

B. A Neuron-Adaptive Activation Function

The used Neuron-adaptive Activation Function is as follows:

$$\phi(x) = A1e^{-x^2} + \frac{A2}{1+e^{-B*x}} \tag{1}$$

where *A1, A2, B* are real variables which will be adjusted during training.

In this study was used a training algortihm of Xu and Zhang [37].

The input-output relation of the *i*th neuron in the *k*th layer can be described by:

$$I_{i,k} = \sum_j w_{i,j,k} O_{j,k-1(u)} - \theta_{i,k} \quad , \quad j = 1,2....k \tag{2}$$

where $w_{i,j,k}$ the weight that connects the *j*th neuron in layer *k*-1 nad the *i*th neuron in layer *k*, $\theta_{i,k}$ is the threshold of the unit, *j* is the number of neurons in layer *k-1,* and

$$O_{i,k}(u) = \varphi(I_{i,k}(u)) = A1_{i,k} e^{-I_{i,k}^2(u)} + \frac{A2_{i,k}}{1+e^{-B_{i,k}*I_{i,k}(u)}} \tag{3}$$

To train this neural network an energy function

$$E = \frac{1}{2}\sum_{j=1}^{m} \left(d_j(u) - O_{j,l}(u)\right)^2 \tag{4}$$

is adopted. In (4) $d_j(u)$ is the *j*th desired output value and *m* is total number of output layer neurons, *l* is the total number of constructed network layers. The aim of learning is minimize the energy function. This can be obtained by using a variation of the steepest descent gradient rule [38] expressed as follows:

$$\omega_{i,j,k}^{(r)} = \eta\omega_{i,j,k}^{(r-1)} + \beta\frac{\partial E}{\partial w_{i,j,k}} \tag{5}$$

$$\theta_{i,k}^{(r)} = \eta\theta_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial \theta_{i,k}} \tag{6}$$

$$A1_{i,k}^{(r)} = \eta A1_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial A1_{i,k}} \tag{7}$$

$$A2_{i,k}^{(r)} = \eta A2_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial A2_{i,k}} \tag{8}$$

$$B_{i,k}^{(r)} = \eta B_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial B_{i,k}} \tag{9}$$

where, η is the momentum and β is the learning rate.

To derive the gradient information of *E* with respect to each adjustable parameter in equations (5) - (9), was defined

$$\frac{\partial E}{\partial I_{i,k}(u)} = \zeta_{i,k} \tag{10}$$

$$\frac{\partial E}{\partial O_{i,k}(u)} = \xi_{i,k} \tag{11}$$

From equations (3), (4), (10) and (11) partial derivatives of *E* can be founded with respect to adjustable parameters as follows:

$$\frac{\partial E}{\partial w_{i,j,k}} = \frac{\partial E}{\partial I_{i,k}(u)}\frac{\partial I_{i,k}(u)}{\partial w_{i,j,k}} = \zeta_{i,k} O_{j,k-1}(u) \tag{12}$$

$$\frac{\partial E}{\partial \theta_{i,k}} = \frac{\partial E}{\partial I_{i,k}(u)}\frac{\partial I_{i,k}(u)}{\partial \theta_{i,k}} = -\zeta_{i,k} \tag{13}$$

$$\frac{\partial E}{\partial A1_{i,k}} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial A1_{i,k}} = \xi_{i,k} e^{-I_{i,k}} \tag{14}$$

$$\frac{\partial E}{\partial A2_{i,k}} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial A2_{i,k}} = \xi_{i,k}\frac{1}{1+e^{-B_{i,k}*I_{i,k}}} \tag{15}$$

$$\frac{\partial E}{\partial B_{i,k}} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial B_{i,k}} = \xi_{i,k}\frac{A2_{i,k} I_{i,k}(u) e^{-B_{i,k} I_{i,k}(u)}}{\left(1+e^{-B_{i,k}*I_{i,k}}\right)^2} \tag{16}$$

And for (10) and (11) following equations can be founded:

$$\zeta_{i,k} = \frac{\partial E}{\partial I_{i,k}(u)} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial I_{i,k}} = \xi_{i,k}\frac{\partial O_{i,k}(u)}{\partial I_{i,k}(u)} \tag{17}$$

while

$$\frac{\partial O_{i,k}(u)}{\partial I_{i,k}(u)} = -2A1_{i,k} I_{i,k}(u) e^{-I_{i,k}^2(u)} + \frac{A2_{i,k} B_{i,k} e^{-B_{i,k} I_{i,k}(u)}}{\left(1+e^{-B_{i,k}*I_{i,k}}\right)^2} \tag{18}$$

$$\xi_{i,k} = \begin{cases} \sum_j \zeta_{j,k+1}\omega_{j,i,k+1} & if \ \ 1 \le k < l \\ O_{i,j}(u) - d_j(u) & if \quad k = l \end{cases} \tag{19}$$

Algorithm works as follows :

1. Apply the input vector, $x = (x_1, x_2, ..., x_N)$ to the input units.
2. Calculate the sum of weighted input signals to the hidden layer.
3. Calculate the outputs from the hidden layer.
4. Calculate the sum of weighted input signals to the outputs layer.
5. Calculate the outputs.
6. Calculate the error terms for the output units.
7. Calculate the error terms for the hidden units.
8. Update weights on the output and hidden layer.
9. Update real variables on the output and hidden layer.

This algorithm is not far from the backpropagation algorithm. The difference is being used an updated real variables in this algorithm. The training data are presented until the energy function is acceptably low and the network converges.

C. Artificial Immune System

Artificial Immune Systems are a relatively new biologically motivated paradigm, which has been explored for less than a decade [39].

Immune systems are naturally existing mechanisms, which are responsible for detecbinting and coping with intruders in living organisms [40]. The main purpose of the immune system is to recognize all cells (or molecules) within the body and categorize those cells as self or non self [41] and protect the organism against disease-causing cells called pathogens and to eliminate malfunctioning cells [42]. All elements recognizable by the immune system are called antigens [42]. It was discovered that people who had been inoculated against diseases contained certain agents that could in some way bind to other infectious agents. These agents were named antibodies [43]. A large number of antibodies may be produced by the immune system in response to the infection that will help to eliminate the antigen from the body [44]. Antibodies posses two paratopes, which are portions of the antibody that are used to match or identify other molecules. The regions on the molecules that the paratope can attach to are called the epitopes. Identification of the antigen is achieved by complementary matching between the paratope and the epitope, comparable to a lock and a key. The strength of the bind depends on how closely the two match. The closer the match between the antibody and the antigen then the stronger the molecular binding and the better the recognition [44].

AIS is a computational technique inspired by ideas coming from immunology and used to develop adaptive systems capable to solve different domain problems [40]. The AIS have become popular over the last year [43]. Applications of AIS include pattern recognition, fault and anomaly detection, data mining and classification, scheduling, machine learning, autonomous navigation, search and optimization areas [45].

The acronym opt-aiNET stands for "Optimization version of an Artificial Immune Network" [43]. It is a particular type of artificial immune system developed to solve optimization problems [46].

The opt-aiNET algorithm can be described as follows [43]:

1. *Initialization:* create an initial random population of network antibodies;
2. *Local search:* while stopping criterion is not met, do:
   - *Clonal expansion:* for each network antibody, determine its fitness (an objective function to be optimized) and normalize the vector of fitnesses. Generate a *clone* for each antibody, i.e., a set of antibodies which are the exact copies of their antibody;
   - *Affinity maturation:* mutate each clone inversely proportionally to the fitness of its parent antibody that is kept unmutated. For each mutated clone, select the antibody with highest fitness, and calculate the average fitness of the selected antibodies;
   - *Local convergence:* if the average fitness of the population does not vary significantly from one iteration to the other, go to the next step; else, return to Step 2;
3. *Network interactions:* determine the affinity (similarity) between each pair of network antibodies;
4. *Network suppression:* eliminate all network antibodies whose affinity is less than a pre-specified threshold, and determine the number of remaining antibodies in the network; these are named memory antibodies;
5. *Diversity:* introduce a number of new randomly generated antibodies into the network and return to Step 2.

IV. PERFORMANCE METRICS

Accuracy, sensitivity and specificity are the common performance metrics used in medical diagnosis tasks. The measure of the ability of the classifier to produce accurate diagnosis is determined by accuracy. The measure of the ability of the model to identify the occurrence of a target class accurately is determined by sensitivity. The measure of the ability of the model to separate the target class is determined by specificity. So that accuracy, sensitivity and specificity are calculated as follows [47]:

$$Accuracy = \frac{Total \ \ number \ \ of \ \ correctly \ \ diagnosed \ \ cases}{Total \ \ number \ \ of \ \ cases} \tag{20}$$

$$Sensitivity = \frac{Total \ \ number \ \ of \ \ positive \ \ cases \ \ correctly \ \ diagnosed}{Total \ \ number \ \ of \ \ positive \ \ cases} \tag{21}$$

$$Specificity = \frac{Total \ \ number \ \ of \ \ negative \ \ cases \ \ correctly \ \ diagnosed}{Total \ \ number \ \ of \ \ negative \ \ cases} \tag{22}$$

V. STUDY ENVIRONMENT

The idea behind this study is to use artificial immune systems for optimization of function, which produced from neural network. Proposed rule extraction algorithm is composed of three parts:

1-Data coding;
2-Classification of coding data;
3- Rule extraction.

A Data Coding

Every data in dataset are coded as binary string and presented as input to ANN. The following method has been used for coding [26]. Let the data have *N* attributes. Every attribute A_n $\{n=1,2,...,N\}$ has been divided in to m_N sub strings as $\{a_1, a_2, ..., a_{m_n}\}$ and coded as binary substring $\{b_{n1}, b_{n2}, ..., b_{nm_n}\}$. If attribute A_n belongs to substring a_i $(i=1,2,...,m_n)$, b_{nj}, it is givenby:

$$b_{nj} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad j = 1,2,...,m_n \tag{23}$$

Thus, the input vector of ANN can be given by:

$$X = \bigcup_{n=1}^{N} \bigcup_{i=1}^{m_n} b_{ni} \tag{24}$$

The length of input vector *X* is determined as follows:

$$m = \sum_{n=1}^{N} m_n \tag{25}$$

The dataset that will be applied in this study consists of two classes, so output layer of ANN can consist of one neuron. Output will be 1, when the presented vector belongs to class 1 and it will be 0, when the presented vector belongs to class 0. The ANN model that is seen in Figure 1has been used for classification.

The sum of weighted input signals for *j*th neuron of hidden layer is calculated as:

$$G_j = \sum_{i=1}^{m} x_i w_{ij} - \theta_j, \quad j = 1,2....k \tag{26}$$

where, *k* is number of neurons in hidden layer, w_{ij} is the weight between *i*th neuron of input layer and *j*th neuron of hidden layer and θ_j is threshold for *j*th neuron of hidden layer. Output of *j*th neuron of hidden layer is calculated as follows:

$$CG_j = A1_j e^{-G_j^2} + \frac{A2_j}{1 + e^{-B_j G_j}}, \quad j = 1,2,...,k \tag{27}$$

where, $A1_j$, $A2_j$, B_j are adjustable variables for activation functiom of *j*th neuron in hidden layer. The sum of weighted input signals for output neuron is calculated as:

$$O = \sum_{j=1}^{k} CG_j * v_j \tag{28}$$

where, v_j is the weight between *j*th neuron of hidden layer and output neuron. Output neuron is calculated as follows:

$$C = \frac{1}{1 + e^{-O+\xi}} \tag{29}$$

where ξ is threshold for output neuron.

Thus, we obtain a nonlinear function *C* that depends on *X*.

$$C(X) = \left(1 + e^{-\sum_{j=1}^{k} \left[A1_j e^{-\left(\sum_{i=1}^{m} x_i w_{ij} - \theta_j \right)^2} + A2_j \left(1 + e^{-B_j \left(\sum_{i=1}^{m} x_i w_{ij} - \theta_j \right)} \right)^{-1} \right] * v_j + \xi} \right)^{-1} \tag{30}$$

The vectors that make output value *C* "1" and "0", are needed to be found to extract rules from ANN. This is also an optimization problem. Opt-aiNET algorithm has been used for optimization.

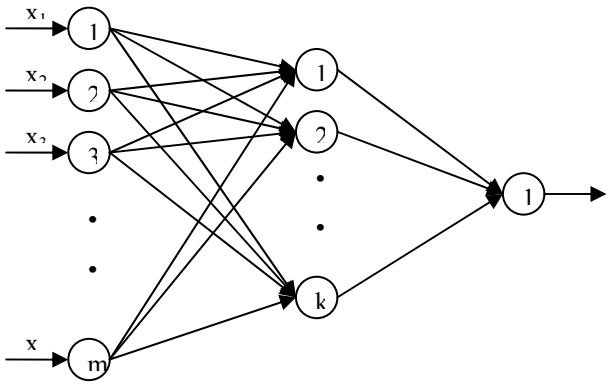

Fig. 1. The structure of used ANN

B Extracting of rules

As result of optimization, the binary vectors, which have *m* length, are produced. Each vector means a rule. To see the rules, these vectors must be decoded. For extracting a rule that belongs to class 0 or class 1 the best antibody must be decoded as follows [26]:

{

- The best antibody is divided into *N* segments.
- Each segment represents one attribute, A_n *(n=1,2,....,N)*, and has a corresponding bits length m_n which represents their values.
- The attribute values existed if the corresponding bits in the best antibody equal one and vice versa.

- The operators "OR" and "AND" are used to correlate the existing values of the same attribute and the different attributes, respectively

}.

VI. THE USED DATASET

Liver is an organ, which is effective in neutralizing toxics and throwing them from the body. If the amount of toxics reaches a level exceeding working capacity of the organ, the cells of related parts in the organ are destroyed. Appeared some substances and enzymes interfere in blood. During diagnosis of the disease, the levels of these enzymes are analyzed. Because of that there are many enzymes and effects of different alcohol dosages vary from one to the other, there can be errors in diagnosis frequently [48].

In this study BUPA Liver Disorders dataset from UCI Machine Learning Repository is used [49]. This dataset contains 345 samples that were taken from only single man. The dataset has 6 attributes. These are:

- mcv mean corpuscular volume
- alkphos alkaline phosphotase
- sgpt alamine aminotransferase
- sgot aspartate aminotransferase
- gammagt gamma-glutamyl transpeptidase
- drinks number of half-pint equivalents of alcoholic beverages drunk per day

Classes were coded as 0 and 1. There are 200 samples in class 0 and 145 samples in class 1. The attributes have different range values in the database and these ranges of the data can be seen in Table I.

Table I. Range values and attribute names

Attribute	Range
Mcv	65 - 103
alkphos	23 - 138
sgpt	4 – 155
sgot	5 - 82
gammagt	5 – 297
drinks	0 - 20

To solve the problem, first a neural network was constructed. Each attribute value was coded as a binary string for use as input to the network. The sub-intervals which are used to coding of attribute values are summarized in Table II. Each bit of a string was either 0 or 1 depending on which sub-interval the original value was located. For example, an sgpt value at 45 would be coded as {0,0,1,0,0}. Sub-intervals for the sgpt are: [4, 20] , (20, 30] , (30, 50] , (50, 70] , (70, 155]. So the binary string for sgpt has five bits. Because of $45 \in (30,\ 50]$, third bit of this string equals to 1 and all others equal to 0.

With the coding scheme shown in Table II, we had a total of 33 binary inputs. As the patients were classified into two classes, a single unit of the output layer was sufficient. The target output was 1 if the patient belonged to Class 1, and 0, otherwise. The number of neurons in the hidden layer was taken as five.

Table II. Coding of the attributes

Attribute	No. Of inputs	Subintervals
Mcv	4	[65, 85] , (85, 90] , (90, 95] , (95, 103]
alkphos	9	[23, 50] , (50, 55] , (55, 60] , (60, 65] , (65, 70] , (70, 75] , (75, 80] , (80, 90] , (90, 138]
sgpt	5	[4, 20] , (20, 30] , (30, 50] , (50, 70] , (70, 155]
sgot	5	[5, 20] , (20, 25] , (25, 35] , (35, 45] , (45, 83]
gammagt	5	[5, 15] , (15, 20] , (20, 35] , (35, 70] , (70, 297]
drinks	5	[0, 0.5] , (0.5, 1.5] , (1.5, 3] , (3, 4.5] , (4.5, 20]

The Opt-aiNET algorithm has been applied to solve the equation *C(X)* (see 30) and in order to get the vectors, which maximizes or minimizes that function. Multiplying factor is 0.5, mutation rate is 10. The Opt-aiNET was then run with a population of 20 for 30000 generation for each classification. All parameters were choosen empirically for the best convergence rate between the actual and desired output. This generates 29 rules for Class 0 and 30 rules for Class 1. Both the maximum and minimum of output antibodies has been determined and will be translated into rules.

In summary, for rule extraction the first ANN which classifies the dataset, was designed. Then Opt-aiNET algorithm was executed for extraction of rules from this ANN. Finally, the extracted rules were decoded.

Produced rules diagnosed correctly 192 samples from 200 belong to Class 0 and 135 samples from 145 belongs to Class1. It means system achieve %96 and %93 correctly diagnosis for Class 0 and Class 1 respectively. In summary the system correctly diagnosed %94.8 of whole samples.

The extracted set of rules have been presented in appendix.

VII. CONCLUSION

Mining of classification rules is an important task of data mining. In this paper, an algorithm for extracting comprehensible classification rules for diagnosis of liver disorders has been presented. This algorithm takes into consideration all input attributes and extracts rules from the trained neural network with adaptive activation function efficiently. Neural network has been trained by adaptive activation function in hidden layer and fixed sigmoid activation function in output layer. OptaiNET that is an AIS algorithm used in extracting rules from the trained neural networks. The approach for extracting rules from ANN consists of three phases:

1-Data coding;

2- Classification of coding data;

3- Rule extraction.

The approach was applied to BUPA Liver Disorders classification problems. The data was obtained from University of California at Irvine (UCI) Machine Learning Repository. The results of comparison experiments show

that the developed approach can generate more accurate rules.

APPENDIX

The extracted set of rules for Class 0.

1. If (mcv $\in[65, 85]$ or mcv $\in(90, 95]$) & (alkphos $\in[23, 50]$ or alkphos $\in(60, 75]$ or alkpho $\in(80, 90]$) & sgpt $\notin(30, 50]$ & sgot >20 & (drinks $\in[0, 0.5]$ or drinks $\in(3, 4.5]$) Then Class 0
2. If mcv ≤ 90 & alkphos ≤ 75 & sgpt $\notin(20, 50]$ & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt ≤ 70 & drinks>3 Then Class 0
3. If mcv ≤ 90 & (alkphos $\in(50, 55]$ or alkphos $\in(60, 75]$ or alkpho $\in(80, 90]$) & sgpt>20 & sgot $\in(20, 45]$ & gammagt>20 & (drinks $\in[0, 1.5]$ or drinks $\in(3, 4.5]$) Then Class 0
4. If mcv $\notin(85, 90]$ & (alkphos $\in(50, 65]$ or alkphos $\in(70, 75]$ or alkpho $\in(90, 138]$) & (sgpt $\in(20, 30]$ or sgpt $\in(50, 70]$) & sgot $\notin(20, 25]$ & gammagt>20 & drinks ≤ 4.5 Then Class 0
5. If mcv>90 & (alkphos $\in(50, 55]$ or alkphos $\in(60, 65]$ or alkphos $\in(75, 138]$) & sgpt $\notin(50, 70]$ & sgot $\notin(20, 25]$ & gammagt $\in(35, 70]$ & drinks>1.5 Then Class 0
6. If mcv $\notin(90, 95]$ & (alkphos $\in[23, 50]$ or alkphos $\in(55, 65]$) & sgot $\in(20, 25]$ & (gammagt $\in[5, 20]$ or gammagt $\in(35, 70]$) & drinks $\notin(1.5, 3]$ Then Class 0
7. If mcv $\in(85, 90]$ & (alkphos $\notin(50, 55]$ & alkphos $\notin(60, 75]$) & sgpt>20 & sgot>25 & gammagt>20 & drinks $\notin(3, 4.5]$ Then Class 0
8. If (mcv $\in(85, 90]$ or mcv $\in(95, 103]$) & (alkphos $\in(55, 60]$ or alkphos $\in(70, 75]$ or alkphos $\in(90, 138]$) & (sgpt $\in[4, 20]$ or sgpt $\in(50, 70]$) & sgot $\notin(25, 45]$ & gammagt $\notin(35, 70]$ Then Class 0
9. If mcv ≤ 95 & alkphos>65 & (sgpt $\in(20, 30]$ or sgpt $\in(50, 155]$) & (sgot $\in(20, 25]$ or sgpt $\in(45, 83]$) & gammagt>20 & (drinks $\in(0.5, 1.5]$ or drinks $\in(4.5, 20]$) Then Class 0
10. If mcv>90 & (alkphos $\notin(60, 65]$ & alkphos $\notin(75, 90]$) & (sgpt $\in[4, 20]$ or sgpt $\in(50, 70]$) & gammagt $\in(15, 35]$ & drinks>1.5 Then Class 0
11. If (alkphos $\in(55, 60]$ or alkphos $\in(65, 75]$) & sgpt $\notin(20, 70]$ & (sgot $\notin(20, 25]$ & sgot $\notin(35, 45]$) & gammagt $\notin(15, 35]$ drinks ≤ 3 Then Class 0
12. If mcv ≤ 95 (alkphos $\notin(55, 60]$ & alkphos $\notin(65, 70]$ & alkphos $\notin(75, 80]$) & (sgpt $\in(20, 50]$ or sgpt $\in(70, 155]$) & sgot $\in[5, 20]$ & gammagt>35 & (drinks $\in(0.5, 1.5]$ & drinks $\in(3, 4.5]$) Then Class 0
13. If mcv $\notin(85, 90]$ & (alkphos $\in[23, 60]$ or alkphos $\in(65, 70]$) & sgpt $\notin(30, 50]$ & (sgot $\in(20, 25]$ or sgot $\in(35, 45]$) & gammagt $\notin(15, 35]$ & drinks>4.5 Then Class 0
14. If mcv>85 & (alkphos $\in(55, 65]$ or alkphos $\in(75, 80]$) & sgpt $\notin(30, 70]$ & (sgot $\in(20, 25]$ or sgot $\in(45, 83]$) & (gammagt $\in[5, 15]$ or gammagt $\in(20, 35]$) & (drinks $\in[0, 0.5]$ or drinks $\in(1.5, 4.5]$) Then Class 0
15. If (alkphos $\in(50, 55]$ or alkphos $\in(65, 138]$) & sgpt ≤ 70 & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt>20 & drinks $\in(3, 4.5]$ Then Class 0
16. If mcv $\notin(85, 95]$ & (alkphos $\in(50, 55]$ or alkphos $\in(60, 65]$ or alkphos $\in(70, 90]$) & (sgot $\in[5, 25]$ or sgot $\in(35, 45]$) & gammagt>15 & (drinks $\in[0, 0.5]$ or drinks $\in(1.5, 4.5]$) Then Class 0
17. If mcv $\notin(90, 95]$ & (alkphos $\in[23, 50]$ or alkphos $\in(55, 60]$ or alkphos $\in(65, 75]$ or alkphos $\in(80, 90]$) & sgpt ≤ 50 & (gammagt $\in[5, 15]$ or gammagt $\in(20, 35]$) & drinks>4.5 Then Class 0

18. If mcv>85 & (alkphos $\in[23, 50]$ or alkphos $\in(55, 60]$ or alkphos $\in(65, 75]$ or alkphos $\in(80, 90]$) & (sgpt $\in[4, 20]$ or sgpt $\in(30, 50]$) & sgot>25 & gammagt>35 Then Class 0
19. If mcv $\in(90, 95]$ & (alkphos $\in(50, 55]$ or alkphos $\in(70, 80]$) & (sgpt $\notin(20, 30]$ & sgpt $\in(50, 70]$) & sgot $\notin(25, 45]$ & gammagt $\notin(15, 35]$ & drinks $\notin(1.5, 4.5]$ Then Class 0
20. If (alkphos $\in[23, 55]$ or alkphos $\in(60, 70]$) & sgpt $\in(20, 30]$ & sgot>25 & gammagt $\notin(20, 35]$ & drinks ≤ 1.5 Then Class 0
21. If mcv ≤ 90 & (alkphos $\in(55, 70]$ or alkphos $\in(75, 138]$) & sgpt ≤ 70 & sgot $\notin(20, 35]$ & gammagt>35 & (drinks $\in(0.5, 1.5]$ or drinks $\in(3, 4.5]$) Then Class 0
22. If mcv $\in(85, 90]$ & (alkphos $\in(50, 65]$ or alkphos $\in(75, 80]$ or alkphos $\in(90, 138]$) & sgpt $\notin(30, 50]$ & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt>20 & drinks $\notin(0.5, 4.5]$ Then Class 0
23. If mcv $\in(85, 90]$ & (alkphos $\in[23, 60]$ or alkphos $\in(75, 138]$) & sgpt $\in(20, 50]$ & sgot $\in(20, 25]$ & gammagt $\in(15, 70]$ & drinks $\in(0.5, 4.5]$ Then Class 0
24. If (mcv $\in[65, 85]$ or mcv $\in(90, 95]$) & (alkphos $\in[23, 60]$ or alkphos $\in(70, 80]$) & sgpt $\in(20, 30]$ & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt $\in(15, 70]$ & drinks>3 Then Class 0
25. If (mcv $\in(85, 90]$ or mcv $\in(95, 103]$) & alkphos $\in(80, 90]$ & sgpt $\in[4, 20]$ & sgot $\notin(20, 35]$ & gammagt $\notin(35, 70]$ & (drinks $\in[0, 0.5]$ or drinks $\in(1.5, 3]$) Then Class 0
26. If mcv ≤ 85 & alkphos $\in(55, 90]$ & sgpt $\notin(20, 30]$ & sgot $\notin(20, 35]$ & (gammagt $\in[5, 15]$ or gammagt $\in(20, 35]$) & (drinks $\in[0, 0.5]$ or drinks $\in(3, 4.5]$) Then Class 0
27. If (alkphos $\in(50, 65]$ or alkphos $\in(70, 80]$) & sgpt $\in[4, 20]$ & (sgot $\in[5, 20]$ or sgot $\in(35, 45]$) & (gammagt $\notin(15, 20]$ & gammagt $\notin(35, 70]$) & (drinks $\in(1.5, 3]$ or drinks $\in(4.5, 20]$) Then Class 0
28. If mcv $\in(90, 95]$ & (alkphos $\notin(60, 70]$ & alkphos $\notin(75, 90]$) & sgpt $\in(30, 50]$ & sgot $\in(20, 35]$ & gammagt>15 & drinks $\in(1.5, 4.5]$ Then Class 0
29. If (mcv $\in[65, 85]$ or mcv $\in(90, 95]$) & (alkphos $\notin(50, 60]$ & alkphos $\notin(80, 90]$) & (sgpt $\in(20, 30]$ or sgpt $\in(50, 70]$) & sgot > 45 & gammagt $\in(20, 70]$ & drinks $\notin(0.5, 3]$ Then Class 0

The extracted set of rules for Class 1.

1. If mcv $\in(85, 95]$ & (alkphos $\notin(60, 65]$ & alkphos $\notin(70, 75]$ & alkphos $\notin(80, 90]$) & sgpt $\in(20, 50]$ & sgot $\notin(20, 35]$ & gammagt $\notin(20, 70]$ & drinks ≤ 3 Then Class 1
2. If mcv > 90 & (alkphos $\in(60, 70]$ or alkphos $\in(75, 138]$) & sgpt $\notin(30, 70]$ & (sgot $\notin(20, 25]$ & sgot $\notin(35, 45]$) & gammagt $\in[5, 20]$ & (drinks $\notin(0.5, 1.5]$ & sgot $\notin(3, 4.5]$) Then Class 1
3. If mcv > 85 & (alkphos $\in(50, 55]$ or alkphos $\in(60, 90]$) & (sgpt $\in(30, 50]$ or sgpt $\in(70, 155]$) & gammagt ≤ 35 (drinks $\notin[0, 0.5]$ & drinks $\notin(3, 4.5]$) Then Class 1
4. If mcv $\notin(85, 90]$ & (alkphos $\notin(50, 75]$ & alkphos $\notin(80, 90]$) & sgpt $\in(30, 50]$ & sgot $\in(20, 45]$ & (gammagt $\notin(15, 20]$ & gammagt $\notin(35, 70]$) & drinks $\notin(0.5, 1.5]$ Then Class 1
5. If mcv $\in(90, 95]$ & (alkphos $\in[23, 50]$ or alkphos $\in(65, 70]$ or alkphos $\in(75, 90]$) & sgpt>20 & (sgot $\in[5, 20]$ or sgot $\in(25, 35]$) & gammagt $\notin(35, 70]$ & drinks $\notin(3, 4.5]$ Then Class 1

6. If (mcv $\in(85, 90]$ or mcv $\in(95, 103]$) & (alkphos $\notin(50, 60]$ & alkphos $\notin(80, 90]$) & (sgpt $\in(20, 30]$ or sgpt $\in(50, 155]$) & sgot $\in(20, 25]$ & gammagt $\notin(15, 35]$ & (drinks $\in[0, 0.5]$ or drinks $\notin(1.5, 3]$) Then Class 1
7. If mcv ≤ 90 & (alkphos $\notin(50, 55]$ & alkphos $\notin(60, 75]$ & alkphos $\notin(80, 90]$) & sgpt ≤ 70 & sgot $\in(25, 45]$ & gammag ≤ 20 & drinks $\notin(0.5, 1.5]$ Then Class 1
8. If mcv ≤ 90 & (alkphos $\in(50, 55]$ or alkphos $\in(65, 70]$ or alkphos $\in(75, 80]$) & sgpt ≤ 30 & (sgot $\in[5, 25]$ or sgot $\in(35, 45]$) & gammagt $\notin(35, 70]$ & drinks $\in(1.5, 3]$ Then Class 1
9. If mcv ≤ 90 & (alkphos $\in(50, 55]$ or alkphos $\in(65, 70]$ or alkphos $\in(75, 80]$) & sgpt ≤ 50 & sgpt ≤ 25 & gammagt ≤ 20 & drinks ≤ 0.5 Then Class 1
10. If mcv $\notin(85, 90]$ & (alkphos $\in(50, 55]$ or alkphos $\in(75, 138]$) & sgpt $\notin(20, 30]$ & (sgot $\in[5, 25]$ or sgot $\in(35, 45]$) & (gammagt $\notin(15, 20]$ & gammagt $\notin(35, 70]$) & (drinks $\notin(1.5, 3]$ or drinks $\notin(4.5, 20]$) Then Class 1
11. If mcv>85 & (alkphos $\in(55, 60]$ or alkphos $\in(65, 75]$ or alkphos $\in(90, 138]$) & sgpt>30 & sgot ≤ 25 & gammagt ≤ 35 & drinks>0.5 Then Class 1
12. If mcv>95 & (alkphos $\in(60, 70]$ or alkphos $\in(75, 80]$ or alkphos $\in(90, 138]$) & sgpt>20 & (sgot $\in(20, 35]$ or sgot $\in(45, 83]$) & (gammagt $\in(15, 35]$ or gammagt $\in(70, 297]$) & drinks $\notin(3, 4.5]$ Then Class 1
13. If mcv>90 & (alkphos $\in(50, 55]$ or alkphos $\in(60, 75]$ or alkphos $\in(90, 138]$) & sgpt $\notin(20, 50]$ & sgot ≤ 45 & gammagt ≤ 70 & drinks>4.5 Then Class 1
14. If mcv ≤ 95 & (alkphos $\in(50, 55]$ or alkphos $\in(60, 70]$) & sgpt>50 & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt $\notin(15, 20]$ & drinks>4.5 Then Class 1
15. If mcv $\notin(85, 90]$ & (alkphos $\in[23, 50]$ or alkphos $\in(65, 70]$) & sgpt $\in(20, 50]$ & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt ≤ 35 & drinks $\notin(3, 4.5]$ Then Class 1
16. If mcv $\in(85, 95]$ & (alkphos $\in(60, 65]$ or alkphos $\in(75, 80]$) & sgot $\in(25, 45]$ & gammagt $\notin(35, 70]$ & (drinks $\in[0, 0.5]$ or drinks $\in(1.5, 4.5]$) Then Class 1
17. If (mcv $\in[65, 85]$ or mcv $\in(90, 95]$) & (alkphos $\in[23, 55]$ or alkphos $\in(60, 65]$ or alkphos $\in(75, 80]$) & (sgpt $\in(30, 50]$ or sgpt $\in(70, 155]$) & sgot ≤ 25 & (gammagt $\in[5, 15]$ or gammagt $\in(35, 70]$) & drinks>1.5 Then Class 1
18. If mcv $\in(90, 95]$ & alkphos $\notin(50, 70]$ & sgpt $\in(20, 50]$ & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & (gammagt $\in(15, 20]$ or gammagt $\in(35, 70]$) Then Class 1
19. If mcv $\in(90, 95]$ & (alkphos $\in(65, 80]$ or alkphos $\in(90, 138]$) & sgpt $\notin(20, 70]$ & (sgot $\in[5, 25]$ or sgot $\in(35, 45]$) & gammagt ≤ 35 & drinks $\notin(3, 4.5]$ Then Class 1
20. If mcv ≤ 90 & (alkphos $\in(65, 70]$ or alkphos $\in(75, 138]$) & sgpt>20 & (sgot $\in[5, 20]$ or sgot $\in(25, 45]$) & gammagt ≤ 70 & (drinks $\in(0.5, 3]$ or drinks $\in(4.5, 20]$) Then Class 1
21. If (mcv $\in(85, 90]$ & mcv $\in(95, 103]$) & alkphos>75 & (sgpt $\notin(20, 30]$ & sgpt $\notin(50, 70]$) & sgot $\in(20, 25]$ & (gammagt $\in(15, 35]$ or gammagt $\in(70, 297]$) & (drinks $\in[0, 0.5]$ or drinks $\in(1.5, 3]$) Then Class 1
22. If mcv $\in(85, 90]$ & (alkphos $\in(55, 75]$ or alkphos $\in(80, 138]$) & (sgpt $\in(20, 30]$ or sgpt $\in(50, 155]$) & sgot ≤ 20 & gammagt ≤ 35 Then Class 1
23. If mcv ≤ 90 & (alkphos $\in(55, 60]$ & alkphos $\in(80, 90]$) & (sgpt $\in[4, 20]$ or sgpt $\in(50, 70]$) & sgot ≤ 35 & gammagt ≤ 70 & drinks $\in(1.5, 3]$ Then Class 1
24. If mcv>90 & (alkphos $\in(50, 60]$ or alkphos $\in(65, 70]$) & sgpt $\in(30, 50]$ &

sgot ≤ 45 & (gammagt $\in [5, 15]$ or gammagt $\in (35, 70]$) & drinks ≤ 0.5 Then Class 1

25. If (mcv $\in (85, 90]$ or mcv $\in (95, 103]$) & (alkphos $\notin (60, 65]$ & alkphos $\notin (70, 80]$) & sgpt $\notin (20, 50]$ & sgot $\in (20, 45]$ & gammagt ≤ 20 & (drinks $\in (0.5, 1.5]$ or drinks $\in (3, 4.5]$) Then Class 1
26. If mcv $\in (90, 95]$ & (alkphos $\in (50, 55]$ or alkphos $\in (60, 70]$ or alkphos $\in (80, 138]$) & sgpt $\notin (20, 50]$ & (sgot $\in [5, 25]$ or sgot $\in (35, 45]$) & gammagt $\notin (15, 70]$ & (drinks $\in (0.5, 1.5]$ or drinks $\in (3, 4.5]$) Then Class 1
27. If (mcv $\in (85, 90]$ & mcv $\in (95, 103]$) & (alkphos $\in (60, 65]$ or alkphos $\in (75, 80]$) & (sgpt $\in (20, 50]$ or sgpt $\in (70, 155]$) & sgot $\notin (25, 45]$ & drinks>4.5 Then Class 1
28. If mcv $\in (85, 95]$ & alkphos $\notin (50, 90]$ & (sgpt $\in (20, 50]$ or sgpt $\in (70, 155]$) & sgot $\in (25, 45]$ & gammagt $\notin (15, 70]$ & (drinks $\in (1.5, 3]$ or drinks $\in (4.5, 20]$) Then Class 1
29. If mcv $\in (90, 95]$ & (alkphos $\in (50, 55]$ or alkphos $\in (60, 70]$ or alkphos $\in (90, 138]$) & sgpt $\in (30, 50]$ & sgot $\in (20, 45]$ & (gammagt $\in [5, 20]$ or gammagt $\in (35, 70]$) & drinks ≤ 0.5 Then Class 1
30. If mcv>90 & (alkphos $\in (50, 55]$ or alkphos $\in (75, 138]$) & (sgpt $\in (20, 30]$ or sgpt $\in (70, 155]$) & (sgot $\in (20, 25]$ or sgot $\in (35, 83]$) & gammagt ≤ 35 (drinks $\in [0, 0.5]$ or drinks $\in (3, 4.5]$) Then Class 1

ACKNOWLEDGMENT

This study is supported by the Scientific Research Projects Unit of Selcuk University.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001
[2] V. Brtka, E. Stokic & B. Srdic, "Automated extraction of decision rules for leptin dynamics—A rough sets approach", *Journal of Biomedical Informatics*, 41 pp.667–674, 2008.
[3] K. C. Tan, Q. Yu, C. M. Heng and T. H. Lee, "Evolutionary computing for knowledge discovery in medical diagnosis", *Artificial Intelligence in Medicine*, 27 pp.129–154, 2003.
[4] K. Rajasethupathy, A. Scime, K. S. Rajasethupathy and G. R. Murray, "Finding ''persistent rules": Combining association and classification results". *Expert Systems with Applicat*ions, doi:10.1016/j.eswa.2008.06.090, 2008.
[5] F. Palacios, X. Li and L. E. Rocha, "Data Mining based on CMAC Neural Networks", *3rd International Conference on Electrical and Electronics Engineering*, 2006.
[6] W. Au, K. C. C. Chan, and X. Yao, "A Novel Evolutionary Data Mining Algorithm With Applications to Churn Prediction", *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 6, 2003.
[7] Y. Qiang Y. Guo, X. Li Q. Wang H. Chen and D. Cuic, "The Diagnostic Rules of Peripheral Lung Cancer Preliminary Study Based on Data Mining Technique", *Journal of Nanjing Medical University*, 21(3) pp.190-195, 2007.
[8] C. C. Bojarczuk, H. S. Lopes, A. A. Freitas and E. L. Michalkiewicz, "A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets", *Artificial Intelligence in Medicine*, 30, 27–48, 2004.
[9] S. Wang, "Classification with incomplete survey data: A Hopfield neural network approach", *Computers & Operations Research*, 32, pp. 2583–2594, 2005.
[10] A. ŻAK, "Neural Model of Underwater Vehicle Dynamics", *International Journal of Mathematics And Computers in Simulation*, Volume 1 (2), pp. 189-195, 2007.
[11] L. Nikolaos, "Radial Basis Functions Networks to hybrid neuro-genetic RBFNs in Financial Evaluation of Corporations", *International Journal of Computers,* Volume 2 (2), pp. 176-182, 2008.
[12] S. Mukhopadhyay, C. Tang, J. Huang, M. Yu and M. Palakal, **"**A comparative study of genetic sequence classification algorithms**",** *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 57 – 66, (2002).
[13] W. Duch, R. Adamczak and K. Grabczwski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules", *IEEE Transactions on Neural Networks*, Vol 11, No 2, 2000.
[14] S.H. Huang and H. Xing, "Extract intelligeble and concise fuzzy rules from neural networks", *Fuzzy Sets and Systems 132*, pp.233-243. 2002.
[15] R. Andrews, J. Diederich and A.B. Tickle, "A survey and critique of techniques for extracting rules from trained artificial neural Networks", *Knowledge-Based Systems*, 8, pp. 373-389, 1996.
[16] A.B. Tickle, R. Andrews, M. Golea, and J. Diederich, "Rule extraction from trained artificial neural Networks", *Neural Networks analysis, architectures and applications*, pp. 61-69, 1997.
[17] E.W. Saad and D.C. Wunsch II, "Neural Network explanation using inversion", *Neural Networks* 20, pp. 78-93, 2007.
[18] E. Keedwell, A. Narayanan and D. Savic, "Evolving rules from neural networks trained on continuous data", *Proceedings of the 2000 Congress on Evolutionary Computation*, 2000.
[19] S.I. Gallant, "Connectionist expert systems", *Communications of the ACM*, Vo1.31, No.2, pp. 152-169, 1988.
[20] G.G. Towell and J. Shavlik, "Extracting refined rules from knowledge based neural networks", *Machine Learning*, 13, pp.71-101, 1993.
[21] J.A. Alexander and M.C. Mozer, "Template-based algorithm for connectionist rule extraction", in:G. Tesauro, D. Touetzky, T. Leen (Eds.), *Advances in Neural Information Processing Systems*. Vol.7 MIT Pres, Cambridge, MA, 1995.
[22] I. Sethi and J. Yoo, "Multi-valued logic mapping of neurons in feed-forward Networks", *Eng. Intel. Syst*, 4 (4), pp.243-153, 1996.
[23] H. Lu, R. Setiono and H. Liu, "Effective data mining using neural Networks", *IEEE Trans Knowledge Data Eng* 8 (6), pp. 957-961, 1996.
[24] R. Setiono and K. Leow, "FERNN: an algorithm for fast extraction of rules from neural Networks", *Appl. Intel*, 12 (1-2), pp.15-25, 2000.
[25] V. Palade, D. Neagu and G. Puscasu, "Rule extraction from neural Networks by interval propagation", *Proceedings of the Fourth IEEE International Conference on Knowledge-Based Intelligent Engineering Systems*, Brighton-UK, pp.217-220, 2000.
[26] A.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.
[27] E.R. Hruschka and N.F.F. Ebecken, "Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach", *Neurocomputing* 70, pp.384-397, 2006.
[28] L. Brito Goncalves, M. M. Bernardes Rebuzzi Vellasco, M.A. Cavalcanti, M.A. Pacheco and F. Joaquim de Souza, "Inverted Hierarchical Neuro-Fuzzy BSP System:A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases", *IEEE Transactions On Systems*, Man And Cybernetics—Part C: Applications And Reviews, Vol. 36, No. 2, 2006.

[29] T. Raicharoen and C. Lursinsap, "Critical Support Vector Machine Without Kernel Function", *Proceedings of the 9th International Conference on Neural information Processing* (ICONIP'OZ) , Vol. 5, 2002.

[30] A.M. Bagirov and J. Ugon, "Supervised Data Classification via Max-min Separability", *Mathematical Modelling of Biosystems,* Springer Berlin Heidelberg, 2008.

[31] L.P. Cordella, C. De Stefano, F. Fontanella, and A. Marcelli, "A Novel Genetic Programming Based Approach for Classification Problems", *Lecture Notes in Computer Science, Image Analysis and Processing – ICIAP 2005.*

[32] H. Kahramanli and N. Allahverdi, "Rule extraction from trained adaptive neural networks using artificial immune systems, *Expert Systems with Applications,* Volume 36 (2), pp. 1513-1522, 2009.

[33] S. Xu and M. Zhang, "Justification of a neuron-adaptive activation function", In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks*, Vol. 3, pp. 465–470, 2000.

[34] G. Tezel and Y. Özbay, "A new approach for epileptic seizure detection using adaptive neural network", *Expert Systems with Applications,* 2008, doi:10.1016/j.eswa.2007.09.007.

[35] A. Zilouchian and M. Jamshidi, *Intelligent Control Systems Using Soft Computing Methodologies*, CRC Press, 2001.

[36] M. Rocha, P. Cortez and J. Neves, "Evolution of neural networks for classification and regression", *Neurocomputing,* 70, pp.2809–2816, 2007.

[37] S. Xu and M. Zhang, "Data Mining – An Adaptive Neural Network Model for Financial Analysis". *Proceedings of the Third International Conference on Information Technology and Applications,* 2005.

[38] D. E. Rumelhart and J. L. McClelland, *Parallel distributed computing: exploration in the microstructure of cognition*, Cambridge, MA: MIT Pres, 1986.

[39] A. Watkins and L. Boggess, "A New Classifier Based on Resource Limited Artificial Immune Systems", *Proceedings of the Congress on Evolutionary Computation* , 2002.

[40] F. Seredynski and P. Bouvry, "Anomaly detection in TCP/IP Networks using immune systems paradigm", *Computer Communications* 30 pp.740-749, 2007.

[41] A. Kalinli and N. Karaboga, "Artificial immune algorithm for IIR fitler design", *Engineering Applications of Artificial Intelligence,* 18 pp. 919-929, 2005.

[42] P. Musilek, A. Lau, M. Reformat and L. Wyard-Scott, "Immune Programming", *Information Sciences* 176. pp. 972-1002, 2006.

[43] L.N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer. UK, 2002

[44] J. Timmis, M. Neal and J. Hunt, "An artificial immune system for data analysis", *BioSystems,* 55, pp. 143–150, 2000.

[45] T. Hou, C. Su and H. Chang, "Using neural Networks and immune algorithms to find the optimal parameters for an IC wire bonding process". *Expert System with Applications* 34. pp. 427-436, 2008.

[46] R.R de Attux, L.T. Duarte, R. Ferrari, C.M. Panazio, L.N.de Castro, F.J. Von Zuben and J.M.T. Romano, "MLP-Based Equalization and Pre-Distortion Using An Artificial Immune Network". *IEEE Workshop on Machine Learning for Signal Processing*, pp. 177-182, 2005.

[47] C.K. Loo, "Accurate and Reliable Diagnosis and Classification Using Probabilistic Ensemble Simplified Fuzzy ARTMAP", *IEEE Transactions on Knowledge and Data Engineering*, Vol 17, No.11, 2005.

[48] M. Yalçın and T. Yıldırım, Karaciğer bozukluklarının yapay sinir ağları ile teşhisi, *Biyomedikal Mühendisliği Ulusal Toplantısı,* Istanbul, Türkiye, pp. 293–297, 2003.

[49] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, UCI Repository of machine learning databases, (1998), [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, last access: 11th August 2008.