Soft Computing Techniques to Extraction Knowledge of Cardiac SPECT Diagnosis

Samaher Hussein Ali 1*, Muhannad M. Al-Yasiry 2**

* *Department of Computer Science, University of Babylon, Iraq*
Samaher_hussein@yahoo.com

** *Department of Computer Science, University of Babylon, Iraq*
Muhannad_Al_Yasiry@yahoo.com

Abstract: This paper presents a methodology for extraction knowledge of Cardiac Single Proton Emission Computed Tomography (SPECT) diagnosis with the use of hybrid techniques represented by soft computing to classify patterns from SPECT database. In this paper the searching capability of a Genetic Algorithm (GA) has been exploited for automatically evolving the structure of neural network as well as proper parameters of neural network. This paper concerns with extraction knowledge of the dataset describes diagnosing of cardiac SPECT images, each of the patients is classified into two categories: normal and abnormal. The GA is considered to face contemporaneously the optimization of the design of neural network architecture and the choice of the best learning method. After that, supervised classification algorithm (Kohonen winner-take-all network) determines the class under which each feature vector belongs to was used. At the last stage, (IF-Then rule) to form several rules that govern each class attributes were used. The proposed methodology achieved 95% accuracy and provides fast and adaptive learning for extraction knowledge.

Key words: Extraction Knowledge, Genetic Algorithm, Kohonen Winner-Take-All Network, Soft computing Techniques

1. INTRODUCTION

Extraction knowledge in Data, the step- by- step hands- on solutions of real- world diagnosis problems using widely available soft computing techniques applied to real- world datasets [1].

The term rule generation encompasses both rule extraction and rule refinement. Note that **rule or Knowledge extraction** here refers to extracting knowledge from the artificial neural network, using the network parameters in the process. Rule refinement, on the other hand, pertains to extracting refined knowledge from the artificial neural network that was initialized using crude domain knowledge, rules learned and interpolated for GA reasoning can also be considered under rule generation. It covers, in a wider sense, the extraction of domain knowledge (say, for initial encoding of an artificial neural network) using non connectionist tools like GA.

The reminder of this paper is organized as follows. Section 2 reviews soft computing. Section 3 describes Genetic Algorithm. Section 4 reviews Kohonen Winner-Take-All Network. Section 5 describes the proposed system for Extraction Knowledge of Cardiac SPECT Diagnosis. Section 6 experiments. Finally section 7 presents the conclusions.

2. Soft Computing

In traditional hard computing, the prime desiderata are precision, certainty, and rigor. By contrast, in soft computing the principal notion is that precision and certainty carry a cost and that computation, reasoning, and decision-making should exploit (wherever possible) the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth for obtaining low-cost solutions. This leads to the remarkable human ability of understanding distorted speech, deciphering sloppy handwriting, comprehending the nuances of natural language, summarizing text, recognizing and classifying images, driving a vehicle in dense traffic and, more generally, making rational decisions in an environment of uncertainty and imprecision. The challenge, then, is to exploit the tolerance for imprecision by devising methods of computation that lead to *an acceptable solution at low cost*. This, in essence, is the guiding principle of soft computing [2].

Soft computing is a consortium of methodologies that works synergetically and provides in one form or another flexible information processing capability for handling real life ambiguous situations. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost by seeking for an approximate solution to an imprecisely/precisely formulated problem. The genetic–neural approach, which provides flexible information processing capability by devising methodologies and algorithms on a extraction knowledge system for representation and recognition of real-life ambiguous situations forms, at this juncture, a key component of soft computing. The principal constituents of Soft Computing (SC) are[3]:

Fuzzy Systems (FS), including Fuzzy Logic (FL); Evolutionary Computation (EC), including Genetic Algorithms (GA); Neural Networks (NN), including Neural Computing (NC); Machine Learning (ML); Probabilistic Reasoning (PR). We can Illustrative Applications of Soft Computing. **SC Technologies Applications**: Neural nets (NN), Fuzzy logic (FL), Probabilistic reasoning (PR), Genetic algorithms (GA) and Hybrid systems. **Related Technologies:** Statistics (Stat.) and Artificial intelligence (AI): includes (Case-based reasoning (CBR), Rule-based expert systems (RBR), Machine learning (ML), and Bayesian belief networks (BBN)). **Application:** Classification includes (Monitoring/Anomaly detection, Diagnostics, Prognostics and Configuration/Initialization), Prediction includes (Quality assessment and Equipment life estimation), Scheduling includes (Time/Resource assignments), Control includes (Machine/Process control, Process initialization and Supervisory control) and DSS/Auto-decision making includes (Cost/Risk analysis, Revenue optimization).

3. Genetic Algorithm

The genetic algorithm (GA) is a heuristic used to find approximate solutions for difficult to solve problems through application of the principles of evolutionary biology to computer science. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination (or crossover). Genetic algorithms are a particular class of evolutionary algorithms.

Genetic algorithms are typically implemented as a computer simulation in which a population of abstract representations (called **chromosomes**) of candidate solutions (called **individuals**) to an optimization problem evolving toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but different encodings are also possible. The evolution starts from a population of completely random individuals and happens in generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), modified (mutated or recombined) to form a new population, which becomes current in the next iteration of the algorithm.

3.1. Basic Description

Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved.

Algorithm is started with a **set of solutions** (represented by **chromosomes**) called **population**. Solutions from one population are taken and used to form a new population. This is motivated by a hope that the new population will be better than the old one. Solutions which are selected to form new solutions (**offspring**) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

This is repeated until some condition (for example a number of populations or improvement of the best solution) is satisfied [4].

3.2. Steady State Genetic Algorithm (ssGA)

In a GA having overlapping generations, only fractions of the individuals are replaced in each generation. The steady state algorithm is illustrated in Figure (1). In each generation, two different individuals are selected as parents, based on their fitness. Crossover is performed with a high probability, Pc, to form offspring. The offspring are mutated with a low probability, Pm and inverted with probability PI, if necessary. A duplicate check may follow, in which the offspring are rejected without any evaluation if they are duplicates of some chromosomes already in the population. The offspring that survive the duplicate check are evaluated and are introduced into the population only if they are better than the current worst member of the population, in which case the offspring replaces the worst member. This completes the generation. In the steady state GA, the generation gap is minimal, since only two offspring are produced in each generation [5].

Advantages of using Steady-State GA (ssGA)

Simple genetic algorithm SGA replaces the entire parent population with the children; one tactic that is often added to a selection strategy is elitism. In elitism the fittest individuals pass unchanged from the parent population to the children one. This is to ensure that the fittest genes are not lost to the next generation.

Steady state GA (ssGA) replaces few individuals, while a standard GA aims at finding a single Global optimum, the application of the concept of the Niching methods (subpopulation) provide a set of solutions rather than only one solution.

In most real life situations the structure of neural network is not known a priori. The real challenge in this situation is to be able to automatically evolve a proper value of neural network parameters as well as providing the appropriate structure of neural network. In this paper, we propose soft computing techniques (neural network, genetic algorithm and attribute classes statically) which can automatically extract the appropriate knowledge of the dataset describes diagnosing of cardiac SPECT images.

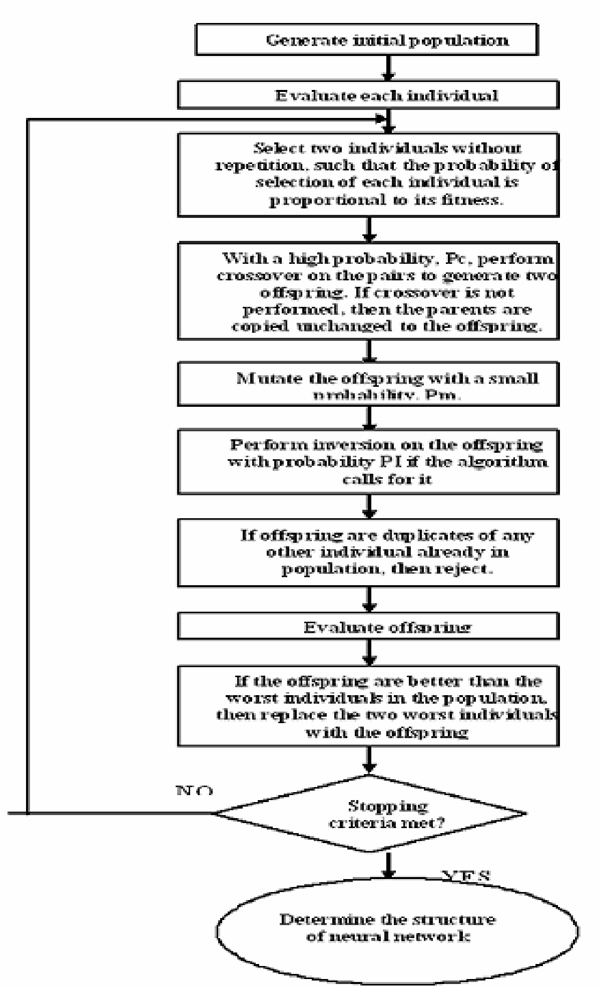

Figure (1): Steady-State Genetic Flowchart

4. Kohonen Winner-Take-All Network

A neural network is a mathematical or computational model for information processing based on a connectionist approach to computation. The original inspiration for the technique was from examination of bioelectrical networks in the brain formed by neurons and their synapses. In a neural network model, simple nodes (or "neurons", or "units") are connected together to form a network of nodes - hence the term "neural network"[6].

Most researchers today would agree that artificial neural networks are quite different from the brain in terms of structure. Like the brain, however, a neural net is a massively parallel collection of small and simple processing units where the interconnections form a large part of the network's intelligence; however, in terms of scale, a brain is massively larger than a neural network, and the units used in a neural network are typically far simpler than neurons. Nevertheless, certain functions that seem exclusive to the brain such as learning have been replicated on a simpler scale, with neural networks.

A typical *feedforward* neural network will consist of a set of nodes. Some of these are designated *input nodes*, some *output nodes*, and those in between *hidden nodes*. There are also connections between the neurons, with a number referred to as a *weight* associated with each connection. When the network is in operation, a value will be applied to each input node - the values being fed in by a human operator, or from environmental sensors, or perhaps from some other program. Each node then passes its given value to the connections leading out from it, and on each connection the value is multiplied by the weight associated with that connection. Each node in the next layer then receives a value which is the sum of the values produced by the connections leading into it, and in each node a simple computation is performed on the value - a sigmoid function is typical. This process is then repeated, with the results being passed through subsequent layers of nodes until the output nodes are reached. Early models had a fixed number of layers. But in this work, genetic algorithms are used to evolve the neural structure.

Kohonen winner-take-all network also known as self-organizing neural networks, are networks which incorporate a topology scheme, i.e., take into account the topological structure among units.

The input signals are n-tuples and there is a set of c database attributes. At each step in the training phase, the weights that best match the input pattern is elected the winner (usually in a minimum Euclidean distance sense).

This winning unit and a neighborhood around it are then updated in such a way that their internal weights be closer to the presented input. The adopted updating factor is not equal for all neurons, but stronger near the winning unit, decreasing for more distant units. Figure 2 shows the basic structure of Kohonen winner-take-all. There are input components connected to all cluster units. The cluster units can assume any spatial distribution, which are usually linear or planar arrays. Weights are associated to each connection. With time, the gain factor must be reduced and also the neighborhood decreases in size.During the learning phase the node weights are changed in an ordered manner, in such a way that the main database features tend to be organized according to a topological distribution in the network. Adjacent nodes respond similarly, while distant nodes respond diversely. The convergence of the features in the winner-take-all occurs considering some limitations on the gain factor while updating the weights.

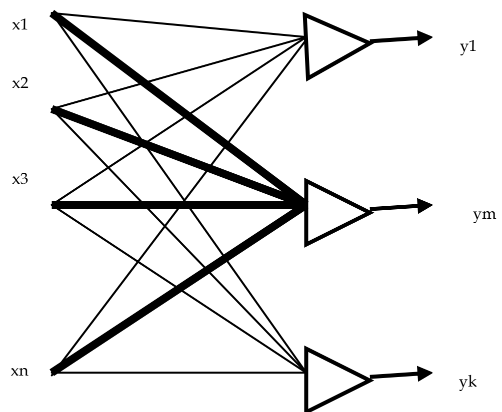

Figure (2): Structure of Kohonen Winner-Take All

5. The Structure of Proposed System

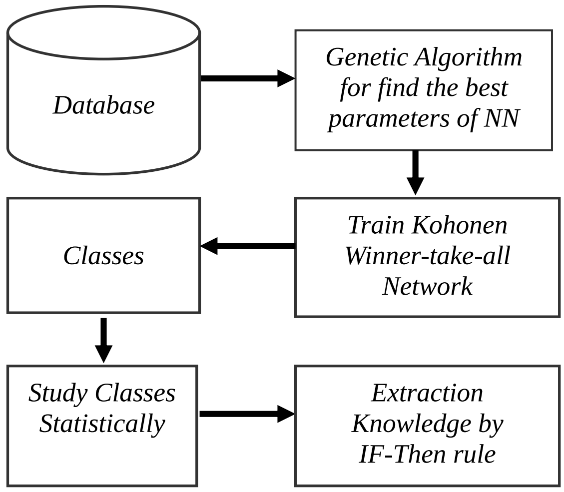

Figure (3): A Block Diagram of overall System

5.1. Cardiac SPECT Database

The dataset describes diagnosing of cardiac SPECT images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature patterns were created for each patient. The pattern was further processed to obtain 22 binary feature patterns.

5.2. Genetic Algorithm to find the Best Parameters of NN

The basic idea is provide an automatic procedure to find the most appropriate neural network structure for extraction knowledge of Cardiac SPECT Diagnosis. Generally speaking, the optimization of a Kohonen winner-take-all network which must be trained to solve a given problem is characterized by the need to determine:

- The architecture

(a) the number of hidden layers NHL

(b) The number of nodes NHI for each layer I

- The activation function f^{l}_{i} to be used for each layer I.
- Absence of the bias b
- The most appropriate technique for weight change.
- The training coefficient η
- The momentum coefficient α.

We have utilized heuristic methods to reduce the search time and probability of getting stuck in local minima. Namely, GA has been used to determine the appropriate set of parameters listed above.

The aim is to provide the most general possible technique to determine the network structure.

5.2.1. Encoding

The neural network is defined by a genetic encoding in which the genotype codes for the different characteristic of the Kohonen Winner-take-all Network and the genotype is the Kohonen Winner-take-all Network itself. Therefore, the genotype contains the parameters related to the network architecture, i.e., the number of hidden layers(NHL) and the numbers of neurons in each layer l (NHl), and other genes representing the activation function type for the output layer f_i^{out} , the parameters related to Kohonen Winner-take-all Network (η and α), the number of neurons in the input layer N^{in}, absence on bias b. For our evaluation algorithm, the chromosome structure y= (y1,…,y_l), constituted by l=8, is reported in Figure(4). Each allele is defined in the subset A_i, i€{1,…,8} reported in the third row of the table 1.

Table 1: The genetic encoding

Locus	y1	y2	y3	y4	y5	y6	y7	y8
Gen	NHL	η	α	b	N^{in}	NH1	NH2	f_i^{out}
Set	A1	A2	A2	A3	A4	A5	A5	A6

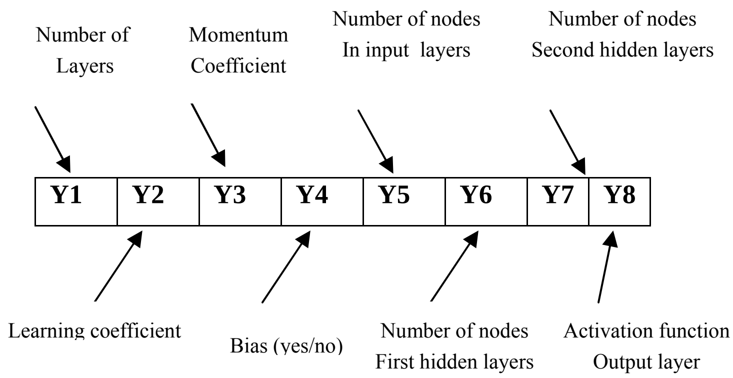

Figure (4): The graphical representation of the genetic encoding

$$\begin{cases} A_1 = \{1,2\} \\ A_2 = [0,1] \\ A_3 = \{0,1\} \\ A_4 = \{1,90\} \\ A_5 = \{1,30\} \\ A_6 = \{1,2,3\} with \{1 \equiv f1, 2 \equiv f2, 3 \equiv f3\} \end{cases}$$

Where A_6 means that the activation function used for the output layer can assume the value f1 for the sigmoid, f2 for tanh and f3 for the combination of three sigmoid function. The best solutions always contained the combination function for both hidden layer. We can represent the activation function used in this paper, as follow

F1= $\dfrac{1}{1+\exp(-net)}$ (1)

F2= $\dfrac{\exp(net)-\exp(-net)}{\exp(net)+\exp(-net)}$ (2)

F3= $s(x+3)-2s(x)+s(x-3)$ (3)

Where

S = F1= $\dfrac{1}{1+\exp(-net)}$

5.2.2. Fitness Function

Whenever an ANN is used, the total pattern set is divided into two subsets: the training set and the verifying set. The former is used to train the network, while the latter is used to evaluate the generalization ability of the trained network.

To evaluate the goodness of an individual, the parameter which seems to describe better its goodness is the normalized mean square error (NMSE) on the training set, E_t. we define NMSE as :

$$NMSE = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - o_i)^2}}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}} \quad (4)$$

Since the ANN generalization ability is computed with reference to the verifying set, it seems natural to take into account also the normalized mean square error on this latter set, Ev. Specifically the fitness function we use is the following:

F(Y) $=E_t+E_v$ (5)

5.3. Train Winner-take-all Network

After we get the best parameters expected to training network (i.e., the best structure of ANN) using genetic algorithm we can train Kohonen winner-take all network.

The network is composed of an orthogonal grid of cluster units, each associated to several internal weights for the features data. The initial values for weights are set to small random values before the learning phase. When an input sample is presented to the network, a search is made to choose the winning unit *c*, as stated in (6). The input vector **x** at time *t* is compared to each of the weight vectors $\mathbf{m}_i \in R^n$ and the minimum Euclidean distance between the input signal and the neuron weights determines the closest match[7].

$$\|\mathbf{x}(t)-\mathbf{m}_c(t)\| = \min_i\{\|\mathbf{x}(t)-\mathbf{m}_i(t)\|\} \quad (6)$$

Then, the weights are updated considering a circular neighborhood around the winning unit, as showed in (7). A scalar gain function $\alpha = \alpha$ (t) is employed to establish how the updating will be done inside the neighborhood *Nc*. Outside this neighborhood the weights remain unchanged. The optimal gain function values are getting by GA,

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}_i(t) - \mathbf{m}_i(t)] \quad \forall i \in N_c(t) \quad (7)$$

5.4. Study Classes Statistically

After classification has been made through Kohonen winner-takeall network, one can inspect each class of the pattrens acquired. There are several methods of inspections. The easiest one is statistical anlysis of each class. Using central tendency and dispertion statistical measures one can form several rules that govern each class attributes[8].

5.4.1. Measures of Central Tendency*:* are measures which are representative of a sample or population. They enable one to be more objective when drawing conclusions or making inferences. These measures identify the *center* or *middle* of a set of values and best characterize the distribution. The typical measures of central tendency are:

• **Mode**: Value which occurs most often. It is the most typical category

• **Median:** Value corresponding to the middle case or middle observation

• **Mean**: Arithmetic average

Mean= sum_of_all_values / number_of_all_values

5.4.2. Measures of Dispersion*:* Another important characteristic of a data set is how it is distributed, or how far each element is from some measure of central tendency (average). There are several ways to measure the variability of the data. Although the most common and most important if which is the standard deviation, which provides an average distance for each element from the mean, several others are also important, and are hence discussed here.

• **Range**: is the difference between the highest and lowest data element. Symbolically, range is computed as *x*max-*x*min, although this is very similar to the formula for midrange. This is not a reliable measure of dispersion, since it uses only two values from the data set. Thus, extreme values can distort the range to be very large while most of the elements may actually be very close to each other. Recently, it has been notes that afew books define statistical range the same as its more mathematical usage. Thus, instead of being a single number it is the interval over which the data occurs. Such books would state the range as [xmin,xmax] or xmin to xmax.

• **Standard Deviation:** The Standard deviation is another way to calculate dispersion. This is the most common and useful measure because it is the average distance of each score from the mean. The formula for standard deviation is as follows.

$$\sigma = \sqrt{\frac{\Sigma(\mu - x_i)^2}{N}} \quad (8)$$

• **Variance:** Variance is the third method of measuring dispersion. In fact variance is just the square of the standard deviation

$$\sigma^2 = \frac{\Sigma(\mu - x_i)^2}{N} \quad (9)$$

5.5. Extraction Knowledge

There are several methods to **Extraction Knowledge.** One of these methods depends on use IF-Then Rule, such as images are described in terms of many characteristics and a rule is given which specifies the attributes that determine membership of the target class [9]. This paper proposes a soft computing (SC) system for Extraction simple classification rules in the following format: IF (a-certain combination -of-attribute values-is-satisfied) THEN (predict-a-certain-class). Each pattern represents a set of these IF-THEN rules. This rule format has the advantage of being intuitively comprehensible for the user. Hence, he/she can combine the knowledge contained in the extracted rules with his/her own knowledge, in order to make intelligent decisions about the target classification problem.

6. Experiment

6.1. Result of Test:

- **Title of database**: SPECT heart data
- **Sources:** Krzysztof J. Cios, Lukasz A. Kurgan

University of Colorado at Denver, Denver, CO 80217, U.S.A.
Krys.Cios@cudenver.edu
Lucy S. Goodenday Medical College of Ohio, OH, U.S.A.

- **Relevant Information**:

The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature pattern was created for each patient. The pattern was further processed to obtain 22 binary feature patterns. The CLIP3 algorithm was used to generate classification rules from these patterns. The CLIP3 algorithm generated rules that were 84.0% accurate (as compared with cardiologists' diagnoses).

- **Number of Attributes**: 23 (22 attribute plus 1 binary class)
- **Attribute Information:**

1. OVERALL_DIAGNOSIS: 0,1 (class attribute, binary)
2. F1: 0,1 (the partial diagnosis 1, binary)
3. F2: 0,1 (the partial diagnosis 2, binary)
4. F3: 0,1 (the partial diagnosis 3, binary)
5. F4: 0,1 (the partial diagnosis 4, binary)
6. F5: 0,1 (the partial diagnosis 5, binary)
7. F6: 0,1 (the partial diagnosis 6, binary)
8. F7: 0,1 (the partial diagnosis 7, binary)
9. F8: 0,1 (the partial diagnosis 8, binary)
10. F9: 0,1 (the partial diagnosis 9, binary)
11. F10: 0,1 (the partial diagnosis 10, binary)
12. F11: 0,1 (the partial diagnosis 11, binary)
13. F12: 0,1 (the partial diagnosis 12, binary)
14. F13: 0,1 (the partial diagnosis 13, binary)
15. F14: 0,1 (the partial diagnosis 14, binary)
16. F15: 0,1 (the partial diagnosis 15, binary)
17. F16: 0,1 (the partial diagnosis 16, binary)
18. F17: 0,1 (the partial diagnosis 17, binary)
19. F18: 0,1 (the partial diagnosis 18, binary)
20. F19: 0,1 (the partial diagnosis 19, binary)
21. F20: 0,1 (the partial diagnosis 20, binary)
22. F21: 0,1 (the partial diagnosis 21, binary)
23. F22: 0,1 (the partial diagnosis 22, binary)

dataset is divided into:

-- training data ("SPECT.train" 80 instances)

-- testing data ("SPECT.test" 187 instances)

- **Missing Attribute Values**: None
- **Class Distribution**:

-- Entire data

Class	# examples
0	55
1	212

-- Training dataset

Class	# examples
0	40
1	40

-- Testing dataset

Class	# examples
0	15
1	172

Apply genetic algorithm to finding the structure of ANN (i.e. find best structure of Kohonen winner-take all network). Before this, we need to determine some of parameters relate to GA as explains in procedure genetic algorithm (in this experiment, pop size=75, pc=1, pm=0.2, no of generation=120*).*

Train Kohonen Winner-take-all neural network on the feature vectors for each pattern to determine the correct class for each feature vector. As a result the network enables from classification of all database patterns.

Production rules have been widely used to represent knowledge in expert systems and they have the advantage of being easily interpreted by human experts because of their modularity, i.e., a single rule can be understood in isolation and does not need reference to other rules. The prepositional like structure of such rules has been described earlier but can summed up as if-then rules. After calculating these statistical measures for each class, one can form one production rule depending on one or more measure. For example we have made the following rule depending on mean and variance of class Mo, such that if variance of attribute1 and attribute 2 is less than some value, say 0.02(small variance more effective attribute), we can make this rule:

IF attrib1 is M1 and attrib2 is M2 then class Mo

Where: *M1, M2* are the mean of attribute 1 and attribute 2 respectively. *Mo* is the attribute of largest mode.
Rules been generated for each class are as follow:

IF (F1 is 0) AND (F2 is 1) AND (F3 is 1) AND (F4 is 1) AND (F5 is 1) AND (F6 is 0) AND (F7 is 0) AND (F8 is 1) AND (F9 is 1) AND (F10 is 1) AND (F11 is 0) AND (F12 is 0) AND (F13 is 1) AND (F14 is 1) AND (F15 is 0) AND (F16 is 1) AND (F17 is 0) AND (F18 is 1) AND (F19 is 1) AND (F20 is 1) AND (F21 is 0) AND (F22 is 0) **THEN** Class is normal

IF (F1 is 0) AND (F2 is 0) AND (F3 is 1) AND (F4 is 1) AND (F5 is 0) AND (F6 is 0) AND (F7 is 0) AND (F8 is 1) AND (F9 is 1) AND (F10 is 0) AND (F11 is 0) AND (F12 is 0) AND (F13 is 1) AND (F14 is 1) AND (F15 is 0) AND (F16 is 0) AND (F17 is 0) AND (F18 is 0) AND (F19 is 0) AND (F20 is 0) AND (F21 is 1) AND (F22 is 0) **THEN** Class is abnormal

7. Conclusion

This paper introduces a fully automatic approach for the extraction knowledge of SPECT heart data set and the result explains: The proposed method will be higher classification accuracy than that of pervious method such as(The CLIP3 algorithm generated rules that were 84.0% accurate (as compared with cardiologists' diagnoses)and Other results: CLIP4 algorithm achieved 86.1% accuracy, ensemble of CLIP4 classifiers achieved 90.4% accuracy And Predicted attribute: OVERALL _DIAGNOSIS (binary))

To receive an acceptable classification result, the structure of neural network needs to be spectrally determined. This can be done with chosen parameters of ANN randomly or depend on using trail and error principle, and the proposed methodology verifies this by applying genetic algorithm. The randomly estimation of the number of the NHL, η.α.b. N^{in}, NH1, NH2and f_i^{out} that are used in the neural network may lead to error in classification process. Therefore, the proposed methodology can solve this problem by determining it automatically. Forming several rules that govern each class attributes by using IF-Then Rule format makes the system more precise because the features extracted from each class which are used to train Kohonen winner-take-all network are congruent to the conditions of this rule, while the resultant classes from Kohonen network are congruent to the actions of this rule. In other words the extracted rules are used to justify the inferred decisions.

REFERENCES

[1] Daniel T. Larose, "Discovering knowledge in Data: An Introduction to Data Mining," John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[2] Andrew.K"Soft Computing Industrial Application" , Intelligent Systems Laboratory Lowa University, 2003. Site: http: //www.icaen.uiowa.edu\~ankusiak

[3] Angela.B, "Soft Computing", England, 2003. Site:http://www.tessella.com

[4] From Wikipedia, the free encyclopedia," Genetic algorithm", GNU Free Documentation license, June 30, 2005.

[5] P. M. Elizabeth and others, "Genetic Algorithm for VLSI Design ,layout & test Automation ", 1999, prentice Hall , PTR USA.

[6] Xin.Y,"Evolving Artificial Neural Networks", Proceedings of the IEEE,Vol. 87, No.9, PP. 1423-1447, September 1999.

[7] Nabeel H. and Samaher Hussein,"Generating Rules from Trained Neural Network using FCM for Satellite Images Classification", IEEE, 4rth International Conference: Sciences of Electronic, Technologies of Information andTelecommunications, TUNISIA, March 2007.

[8] Mahdi A.," Extracting Rules from Databases Using Soft Computing", M.Sc Thesis, University of Babylon, 2005.

[9] Sushmita.M and Yoichi.H.,2000. *Neuro-Fuzzy Rule Generation: Survery in Soft Computing Framework*, IEEE Trnasaction on Neural Network,Vol. 11,No. 3.