ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

DISCRIMINATION OF HEART RATE VARIABILITY USING DECISION TREES AND MLP NETWORKS

Gomathi.S, Mohan Raj.T, Saranya Sri.R

ABSTRACT- The main objective of the paper is to analyze the heart rate variability (HRV) of various subjects. The ECG signals collected from the public data base is Categorized using the Classification and Regression Tree (CART). The techniques analyzed in this paper is to minimize the fault occurrence during the decision tree construction and optimize the tree size using the Multi layer Perceptrons(MLP) with the help of online node fault injection training algorithm. To find the proof of convergence the probability of the error in the perceptrons is calculated using mean square error method from this efficiency of the algorithm can be identified.

KEYWORDS: Classification and regression Tress (CART), Multi layer Perceptrons (MLP), Heart Rate Variability (HRV), and node fault injection.

I.INTRODUCTION

HEART rate variability (HRV) is a noninvasive measure commonly used to assess the influence of autonomic nervous system (ANS) on the heart. HRV is widely studied in patients suffering from chronic heart failure (CHF). CHF is a pathophysiological condition in which an abnormal cardiac function is responsible for the failure of the heart to pump blood as required by the body. CHF is chronic, degenerative and age related. Early detection is crucial to avoid the condition worsening and to prevent complication of clinical conditions, which may cause higher social costs. Therefore, new noninvasive and low-cost techniques for early assessment of CHF severity,

Manuscript received Feb 10, 2013.

***Gomathi.S**, Computer Science and Engineering, Karpagam University, Coimbatore, India.*
***MohanRaj.T**, Computer Science and Engineering, Karpagam University, Coimbatore, India.*
***SaranyaSri.R**, Computer Science and Engineering, Karpagam University, Coimbatore, India.*

could contribute to contain the number of patients and related costs. The classification of subjects according to the heart rate variability is the problem statement analyzed in this paper. The various possibilities are studied and the different techniques are experiment for the classification of the subjects.

New approach of measuring HRV from short and variable-duration (31–95 s) pulse wave signals produced by an automated oscillometric BP monitor during routine measurements. To characterize HRV, we employ the maximal overlap discrete wavelet transform (MODWT) - based spectral density estimation method introduced by Percival and Walden. We defined accuracy as the quality of the linear regression of HRV with age and BP. The main challenge that we have faced in this study was to manually correct peaks that were falsely identified or missed by the peak detection algorithm before we carried out the variability analysis.

Further Investigation on Cardiovascular Risk Prediction Using Genetic Information is used in identifying inherited genetic variants that are associated with complex disease. Such variants enable a clarification of the disease mechanisms and improve the efficacy of Heart rate variability diagnostics and therapeutics. Genetic information is introduced into risk models to predict the risk of Chronic Heart Failure when it is in its latent or even earlier period. Major disadvantage is more efforts should still be made before integrating genetic information into cardiovascular risk prediction model clinically, because the genes that can fully explain complex heart failure have not yet been identified. The paper deals with the efficient and optimized classification of the ECG signals and construction of the Decision Tree.

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

II.CART ALGORITHM

The CART (Classification and Regression Trees) algorithm is a classification method which uses historical data to construct decision trees. Depending on available information about the dataset, classification tree or regression tree can be constructed. Decision tree can be then used for the classification of various new observations. The fundamental principles included in the tree construction, different splitting algorithms and pruning procedures.

CART method is used for building the decision trees with pre-assigned classes for all observations. Decision trees are represented by set of questions which splits the learning sample into smaller and smaller parts. CART algorithm will search for all possible variables and values in order to find the best split-the questions that split the data into two parts with a maximum homogeneity are considered. The process is then repeated for each of the resulting data fragments. Usually the splitting algorithm will isolate outliers in individual nodes.

An important vital property of CART is that the structure of its classification or regression trees is invariant with respect to monotone transformations of independent variables. One can replace the possible variable with its logarithmic value or the square root values, structure of the tree will not be changed. CART methodology consists of the following three parts:

1. Construction of the maximum tree.

2. Choice of the right tree size.

3. Classification of new data using constructed tree.

The disadvantages of CART may have the unstable decision tree constructed to process the splits by only one variable. Classified as normal or CHF, according to the percentage of excerpts classified as CHF. Measurements: the difference, over the 24 h, between the maximum and the minimum values both for AVNN and LF/HF called, respectively, ΔAVNN and ΔLF/HF. According to the threshold value the ECG signal is measured and the variation in the left and right frequency of the signal measurements results in the accuracy of the data.

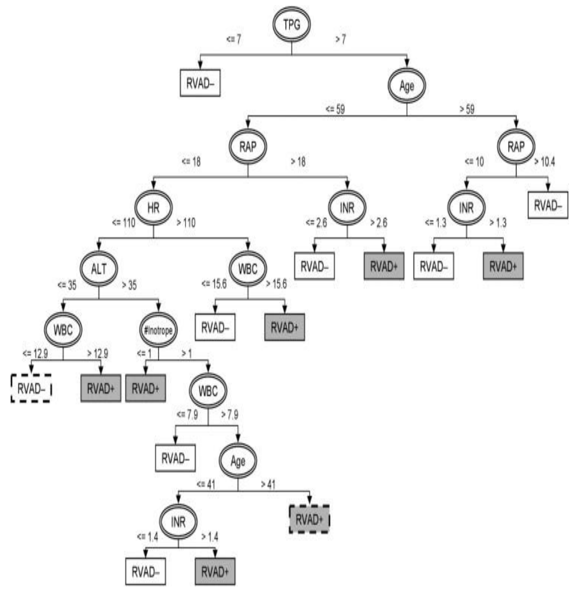


Fig 1. Decision Tree

Steps followed for building decision trees is

- Check for base cases
- For each attribute *a*
  - Find the normalized information gain from splitting on *a*
- Let *a_best* be the attribute with the highest normalized information gain
- Create a decision *node* that splits on *a_best*
- Recurse on the sublists obtained by splitting on *a_best*, and add those nodes as children of *node*.

When there is increase in the data collection the CART becomes tedious in mining the record and fault occurrence in the tree construction need to be studied and problem need to be solved in a optimized way.

III.MLP CONSTRUCTION

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. The expert can then be used to provide projections given new situations of interest and answer "what if" questions. Neural network are prone to be fault tolerance the objective is to identify and improve it. The fault tolerance in the decision tree had been studied and various training algorithm had been proposed. Neural network contains nodes and forms a network with many interconnections among the nodes with specific weight between the nodes. The MLP network has 3 layers input layer, output layer, hidden layer.

In the initial stage the nodes are in supervisied learning phase where the different samples are been studied. The input nodes with the sum of the weights are computed and processed in the hidden layer. The output of the network is compared with possible combination of the inputs. Input layer is a vector predictor variable values is presented to the input layer. It standardizes these values so that the range of each variable is -1 to 1. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse.

Hidden layer arriving at a neuron network is, the value from the each input neuron is multiplied by its weight (wji) and the resulting values are added together producing a combined value Uj. Output layer arriving at a neuron is, the value from the each hidden layer neuron is multiplied by a weight (wji) and the resulting weighted values are added together producing a combined value Vj.

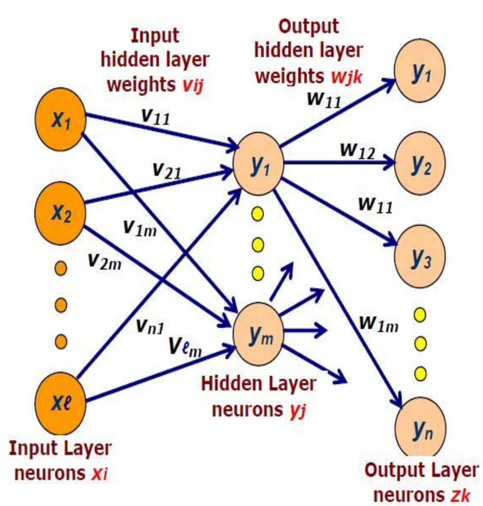


Fig 2. MLP network for HRV

IV.RELATED WORK

We assume that the training data set

$D = \{(x_k, y_k)\}_{k=1}^{N}$ is generated by an unknown system, where $x_k \in R^n$ is the input and $y_k \in R$ is the output.

A. MLP with Single Linear Output Nodes

The unknown system is approximated by a MLP showed in Fig. 1 with n input nodes, m hidden nodes, and one linear output node. The output is given by

$$f(x_k, d, A, c) = d^T z(x_k, A, c) = \sum_{i=1}^{m} d_i z_i(x_k, a_i, c_i) \quad (1)$$

$$z_i(x_k, a_i, c_i) = \frac{1}{1 + \exp(-(a_i^T x_k + c_i))} \quad (2)$$

for $i = 1, 2 \ldots m$, where $A = [a1, a2, \ldots, a_m] \in R^{n \times m}$ is the input-to-hidden weight matrix, $a_i \in R^n$ is the input weight vector associated with the i^{th} hidden node, $c = (c1 \ldots cm)^T \in R^m$ is the input-to-hidden bias vector, $d \in R^m$ is the transfer function of the i^{th} hidden mode, where

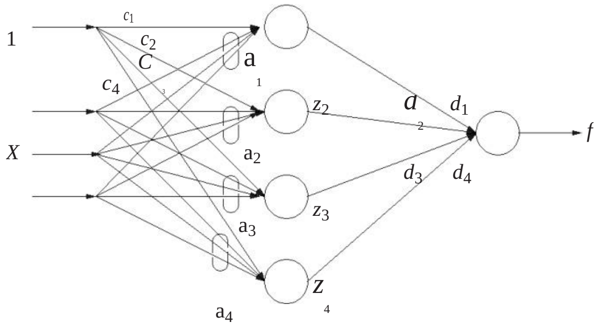
$$w_i = (d_i, a_i^T, c_i^T) \quad . \quad (3)$$

Besides, we let w be a parametric vector augmenting all the parametric vectors
$w_1, w_2, \ldots, w_m$

$$w = (w_1^T, w_2^T, \ldots, w_m^T)^T . \quad (4)$$

In other words, the output of the network can be denoted as $f(x_k, w)$. Throughout this paper, we call $w_1, w_2 \ldots w_m$ and w the weight vectors.

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013



$$w_1 = \begin{bmatrix} d_1 \\ a_1 \\ c_1 \end{bmatrix}, w_2 = \begin{bmatrix} d_2 \\ a_2 \\ c_2 \end{bmatrix}, w_3 = \begin{bmatrix} d_3 \\ a_3 \\ c_3 \end{bmatrix}, w_4 = \begin{bmatrix} d_4 \\ a_4 \\ c_4 \end{bmatrix}$$

Fig 3. Structure of a linear output MLP with four sigmoid hidden nodes.

V. CONVENTIONAL TRAINING METHODS

In weight decay training, a sample is randomly drawn from the dataset D at each training iteration. We denote the sample being selected at the t^{th} iteration as $\{x^t,y^t\}$.Let δ1(t), . . . , δN (t) be N binary random variables such that

$$\sum_{k=1}^{N} ä_k(t) = 1 \qquad (5)$$

for all t ≥ 0. Besides, random vectors

(δ1 (t1) . . . δN (t1))T and

(δ1(Tt2), . .T. , δN (t2) are independent for all t1 ≠ t2.

Then, (x_t, y_t) can be defined as

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \sum_{k=1}^{N} ä_k(t) \begin{bmatrix} x_k \\ y_k \end{bmatrix} \qquad (6)$$

The input xt is thus fed in the MLP, and the output is calculated by (1) and (2). To shorten the length of the equations, we denote zi (xt, ai (t), ci (t)) in (2) by zi (t) from now on.

The updated equations of the weight vectors w_i $i = 1, 2, \ldots, m$) can then be written as follows:

$$w_i(t+1) - w_i(t) = ì(t)\,\{(y_t - f(x_t, w(t)))g_i(x_t, w(t)) - áw_i\, t)\} \qquad (7)$$

Where

$$\mathbf{g}_i(\mathbf{x}_t, \mathbf{w}(t)) = \frac{\partial}{\partial \mathbf{w}_i} f(\mathbf{x}_t, \mathbf{w})|_{\mathbf{w}=\mathbf{w}(t)} = \begin{bmatrix} z_i(t) \\ d_i(t)z_i(t)(1-z_i(t))\mathbf{x}_t \\ d_i(t)z_i(t)(1-z_i(t)) \end{bmatrix}. \qquad (8)$$

ì(t) > 0 is the step size at the t^{th} iteration

is the decay constant controlling the amount of decay of the weight vector $w_i(t)$ in each update iteration. Conventionally, the initial weight vector w(0) is set to a small random vector with elements around zero.

Similar to the notation w in (4), we can define a vector g which augments all the vector functions $g_1 \ldots g_m$, given by

$$g = (g_1^T, g_2^T, \ldots, g_m^T)^T \qquad (9)$$

From (7)

$$w(t+1) - w(t) = ì(t)\,\{(y_t - f(x_t, w(t)))g(x_t, w(t)) - áw(t)\}. \qquad (10)$$

Conventionally, the initial weight vector w (0) is set to a small random vector with elements around zero. By using notation $\delta_k(t)$ in (5) and (7) can be rewritten as

$$w_i(t+1) - w_i(t) = \mu(t)\{\sum_{k=1}^{N}\delta_k(t)(y_k - f(x_k, w(t)))g_i(x_k, w(t)) - \alpha w_i(t)\}. \qquad (11)$$

In, the objective function of the weight decay training algorithm (7) was derived as probability.

VI. FAULT INJECTION TRAINING

The nodes in the MLP may have the fault the algorithm which we propose online node fault injection algorithm is embedded with the nodes in the network. We consider three cases for multi layer perceptrons (MLP):

1) MLP with single linear output node. One input node and one output node in the layers.

2) MLP with multiple linear output node and output to many nodes.

3) MLP with single sigmoid output node.

One output but many input. The hidden nodes randomly produce zero during the training stage. When the output value is not zero fault is in encounted and the fault is rectified by a counter value so the fault identification becomes a easier task since the nodes are in on among the three cases. The possible combination of the inputs are anlaysed with the sum of their weight in the hidden layer to match with the required output.The process flow for the fault tolerence are:

1)formulation of neural network

2)input and hidden layer formation

3)output rule for the proposed network

4)adaptive rule formulation

5)supervisied learning of the network

6)training and graphical implementation of the precptrons

7)visualization

VII.CONVERGENCE ANALYSIS

To find the proof of convergence the probility of the error in all the three types of precptrons is calculated using mean square error (MSE) method from this efficency of the algorithm is identified. MSE is the average of the square of the "error".The error is the amount by which the value implied by the estimator differs from the quantity to be estimated.For the random sample of size n X1,X 2,…Xn. It is possible to significantly reduce the false negatives, improving the sensitivity of the classifier and enhancing the subject classifier with the tree, which bases on the two nonstandard measures ΔAVNN and ΔLF/HF.

The execution of the nodes after the training phase is stimuated as the graph and the error occurance and the performance of the fault tolerence algorithm is studied by the result.The accuracy of the data classfication is increase to due the error identification and correction.The thresold value of the root can be adjused as per the requirement and the symptoms analysis.Moreover, we demonstrated that it is possible to enhance the discriminative power of these measures by adding a few nonstandard measures of HRV. The results show that it is possible to discriminate normal subjects from CHF ones by using short-term HRV measures, extracted from 24-h Holter registrations.

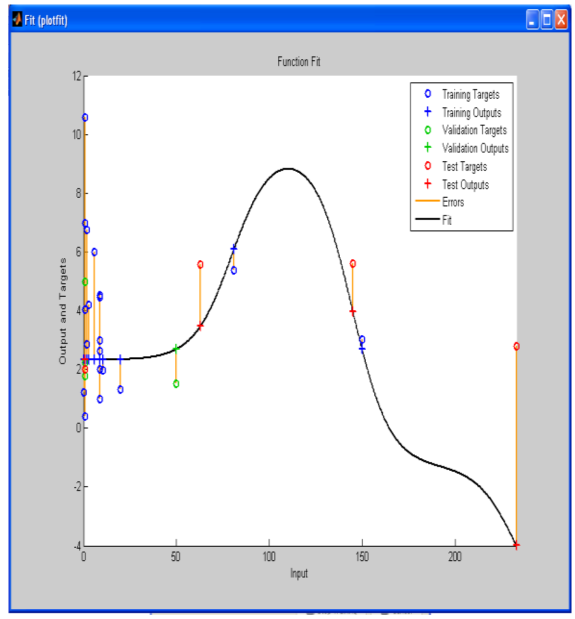


Fig 4.Fault Tolerance Graph.

VIII.CONCLUSION

The terminal nodes for CHF class are on the left of their parent nodes, reflecting the fact that CHF subjects have a depressed HRV. In this paper, we have presented the convergence analyses and the objective functions of the On-line node fault injection training algorithm for MLPs with single linear output node, multiple linear output nodes, and single sigmoid output node. With the objective functions presented, we are now able to compare the similarities and differences between the algorithms developed based on fault/noise injection during training and those developed based on the idea of regularization.

ACKNOWLEDGMENT

Thanks to the experts who have contributed for the development of discrimination of heart rate variability using decision trees and MLP networks.

REFERENCES

[1] M.Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A.Malliani, A. J. Moss,and P. J. Schwartz, “Heart rate variability: Standards of measurement, physiological interpretation, and clinical use,” *Eur. Heart J.*, vol. 17,no. 3, pp. 354–381, Mar. 1996.

[2] J. P. Saul, Y. Arai, R. D. Berger, L. S. Lilly, W. S. Colucci, and R. J. Cohen, “Assessment of autonomic regulation in chronic congestive heart failure by heart rate spectral analysis,” *Am. J. Cardiol.*, vol. 61, no. 15, pp. 1292–1299, Jun. 1988.

[3] J. T. Bigger, J. L. Fleiss, R. C. Steinman, L. M. Rolnitzky,W. J. Schneider, and P. K. Stein, “RR variability in healthy, middle-aged persons compared with patients with chronic coronary heart-disease or recent acute myocardial-infarction,” *Circulation*, vol. 91, no. 7, pp. 1936–1943, Apr. 1995.

[4] G. C. Casolo, P. Stroder, A. Sulla, A. Chelucci, A. Freni, and M. Zerauschek,“Heart-rate-variability and functional severity of congestiveheart- failure secondary to coronary-artery disease,” *Eur. Heart J.*, vol. 16, no. 3, pp. 360–367, Mar. 1995.

[5] C. T. Chiu, K. Mehrotra, C. K. Mohan, and S. Ranka, “Modifying training algorithms for improved fault tolerance,” in *Proc. Int. Conf. Neural Netw.*, vol. 4. Orlando, FL, Jun. 1994, pp. 333–338.

[6] C. S. Leung and J. Sum, “Analysis on generalization error of faulty RBF networks with weight decay regularizer,” in *Proc. Int. Conf. Neural Inf. Process.*, vol. 5507. Dec. 2009, pp. 316–323.

[7] P. Edwards and A. Murray, “Fault tolerance via weight noise in analog VLSI implementations of MLPs-a case study with EPSILON,” *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, vol. 45, no. 9,pp. 1255–1262, Sep. 1998

[8] A. F. Murray and P. J. Edwards, “Synaptic weight noise during multilayer perceptron training: Fault tolerance and training improvements,” *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 722–725, Jul. 1993.

[9] A. F. Murray and P. J. Edwards, “Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training,” *IEEE Trans. Neural Netw.*, vol. 5, no. 5, pp. 792–802, Sep. 1994.

[10] R. Mulvaney and D. Phatak, “Generalized heart DWT and transformations between decision trees and neural networks,” *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 81–93, Jan. 2006.

[11] D. S. Phatak and I. Koren, “Complete and partial fault tolerance of feedforward neural nets,” *IEEE Trans. Neural Netw.*, vol. 6, no. 2, 446–456, Mar. 1995.

[12] D. Deodhare, M. Vidyasagar, and S. S. Keerthi, “Synthesis of fault-tolerant feedforward neural networks using minimax optimization,”*IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 891–900, Sep. 1998.

[13] C. Neti, M. H. Schneider, and E. D. Young, “Maximally fault tolerant neural networks,” *IEEE Trans. Neural Netw.*, vol. 3, no. 1, pp. 14–23, Jan. 1992.

Gomathi.S received her B.E degree in Karunya University in 2011 and currently doing her M.E in Karpagam University.

Saranya Sri.R received her B.C.A degree in Sri Krishna college of Arts and Science in 2008 , M.C.A in Bharathiyar University in 2011 and currently doing her M.E in Karpagam University .

Mohan Raj.T received his M.E in Kumaraguru College of Techonology coimbatore currently working as Assistant Professor in Karpagam University.