Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 20 – 30

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Performance Analysis of Support Vector Machine and Neural Networks in Detection of Myocardial Infarction

Nitin Aji Bhaskar[a,*]

[a]*Electrical Engineering, Rochester Institute of Technology, Rochester, NY – 14623, USA*

Abstract

One of the most common form of cardiac abnormality is Myocardial Infarction (heart attack) arises when the artery connecting the heart is blocked and there is no sufficient blood or oxygen, which makes the cells present in that region of the heart to die. This paper aims to process and classify an ECG signal as healthy subject or subject diagnosed with Myocardial Infarction (MI) using Artificial Neural Networks (ANN) and SVM (Support Vector Machine). LIBSVM[1] is utilized for the classification with SVM and backpropogation artificial neural networks with varying hidden layers and nodes are also implemented for performance analysis.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Support Vector Machine; Neural Networks; Myocardial Infarction; Principal Component Analysis; Wavelet Transformation

1. Main text

ECG analysis is recognizing the pattern by performing certain pre-processing techniques and then classifying the arrhythmia in real time. Detection and classification of ECG signals have been undertaken through numerous algorithms. Classification of abnormalities like ST-T in ECG helped in the detection of left ventricular strain which Devine and Macfarlane[3] implemented using ANN. Hu et al[4] detected QRS and classified beats with ANN by modeling nonlinear background noise. Multilayer perceptron with an structure was used to augment the QRS complexities to improve detection. Silipo et al.[5] used two methods for classification which was a unsupervised and

* Corresponding author. Tel.: +1-585-690-6533
Email address: fxn6113@rit.edu

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.01.043

supervised learning technique to classify ECG's. Cardiac and ventricular arrhythmias were detected by Sugiura et al. [6] with a fuzzy logic implemented method. Acharya et al.[7] used variability of heart rate as base signal and ANN and fuzzy equivalence did classification for four cardiac arrhythmias. Arrhythmia classification using SVM was proposed by Song et al.[8] where linear discriminant analysis (LDA) was implemented for reduction of feature dimensions. Previous works in this field M. Arif et al [9], demonstrates the usage of Neural Networks in which time domain features are classified with the help of Backpropogation Neural Networks and Resilient back propagation learning algorithm.

In this paper a performance analysis of two different techniques namely Support Vector Machines and Artificial Neural Networks is done through classification of an ECG signal as a healthy subject or subject diagnosed with Myocardial Infarction. The ECG signals were obtained from the PTB (Physikalisch-Technische Bundesanstalt) consisting of numerous patients with varied heart diseases and healthy subjects. This classification focuses only on the patients diagnosed with Myocardial Infarction. One to five records are present for each Subject. Every record consists of the conventional 12 lead ECG signal along with three Frank lead ECG's and all the signals are digitized at 1000 samples per second. An overview of the total number of subjects and records used for the classification is described in Table 1.

Table 1. PTB Diagnostic ECG Database Data.

Type	Number of Subjects	Number of Records
Healthy Control	52	82
MI	148	367

2. The Classification

The different stages used for classification of ECG comprises of pre-processing stage which implements the Pan Tompkins Algorithm, followed by Wavelet Transformation and Principal Component Analysis (PCA). The features obtained from PCA, which forms the dataset for classification is then subjected to Backpropagation Neural Networks and SVM to analyze their performances.

2.1. First Stage: ECG Signal Pre-Processing

The first step in the analysis of the 15 signals (the conventional 12 leads and the 3 Frank Leads) in the PTB[2] database is preprocessing. Pan Tompkins Algorithm is used to detect the QRS complex, which identifies the slope, amplitude and width by removing noise and baseline drift. The algorithm comprises of a series of filters such as high pass and low pass filter followed by methods like differentiation, squaring, and integrating the signal. Fig.1. describes the Pan Tompkins Algorithm methodology

Fig.1. Pan Tompkins Algorithm

The input signal has to be first downsampled from 1000 samples per second to 200 samples per second using a downsampling conversion factor of 5. A Cascaded-Integrator Comb (CIC) Filter is used to do this conversion. The transfer function used for the conversion is as follows

$$H(z) = \frac{(1-z^{-RM})^N}{(1-z^{-1})} \tag{1}$$

The sampling rate change factor R used is 5, the number of samples per stage used is 2 and the number of cascaded stages is 10. After the signal has been downsampled, it is subjected to band pass filtering.

2.1.1 Low Pass Filtering

The filter used in this paper produces attenuation greater than 35dB at 60H*z*. This reduces the high frequency noise produced by the chest and extremity muscles and also suppresses the electrical interference from the power grid if present. In order to reduce the computational complexity, the low pass filter has integer coefficients with the transfer function defined as follows

$$H_{lp}(z) = \frac{1(1-z^{-6})^2}{32(1-z^{-1})^2} \tag{2}$$

Fig.2.(a) Shows the Input ECG signal, which is subjected to Pan-Tompkins Algorithm and Fig.2.(b) Represents the ECG signal after low pass filtering. All figures (2) to figures (7) have voltage on y-axis and seconds on the x-axis

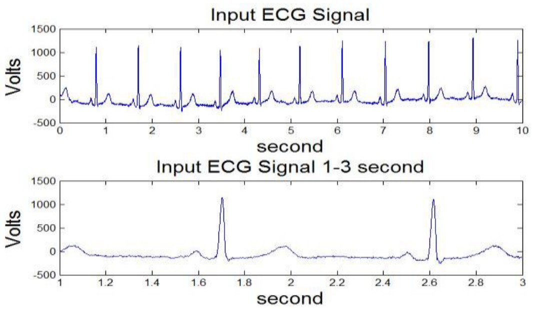

Fig.2.(a) Input ECG Signal

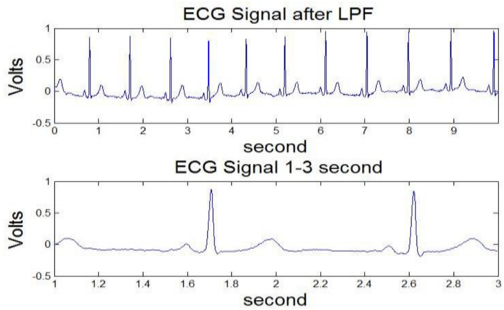

Fig.2.(b) Signal after low pass filtering

2.1.2 High Pass Filtering

This type of filter is utilized for reducing the low frequency noise present in the ECG signal as well as to remove baseline drift. An all-pass filter minus a low pass filter obtains the desired filter. The filter has a transfer function, which is described as follows

$$H_{hp}(z) = z^{-16} - H_{lp}(z) \tag{3}$$

The high pass filter used has a cut off frequency of 5H*z* and also produces a delay of 80ms. Fig.3. shows the filtered ECG signal

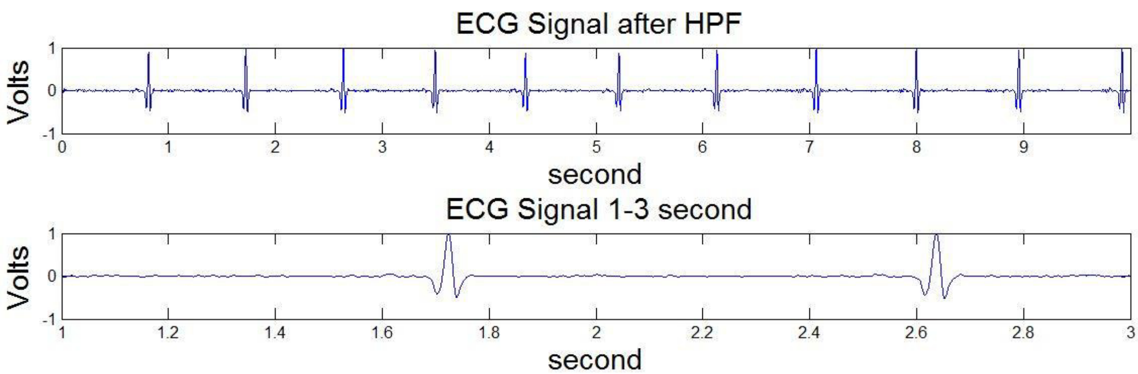

Fig.3 ECG Signal after high pass filtering

2.1.3 Differentiator

The differentiator is used to approximate the d/dt operator up to 30Hz. Here, low frequency components are suppressed and a large gain is provided to the high frequency components occurring from the slopes of the QRS complex. The differentiator used is as follows

$$y(n) = \frac{1[2x(n) + x(n-1) - x(n-3) - 2x(n-4)]}{8} \tag{4}$$

The output obtained after the differentiator operation is in Fig.4.

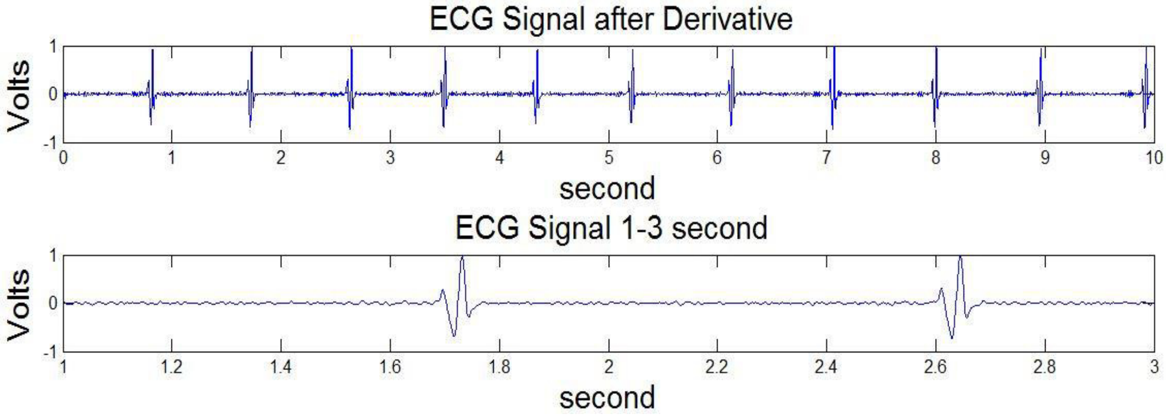

Fig.4. ECG Signal after Differentiator

2.1.3 Squaring

In order to enhance the high frequency component related to signal present in the QRS complex squaring operation is performed. Squaring makes the result positive and focuses on the large difference arising from the QRS complex. Output after the squaring operation is shown in Fig.5. The squaring operation is performed as follows

$$y(n) = [x(n)]^2 \tag{5}$$

This also helps in suppressing differences, which are small and present in P and T waves.

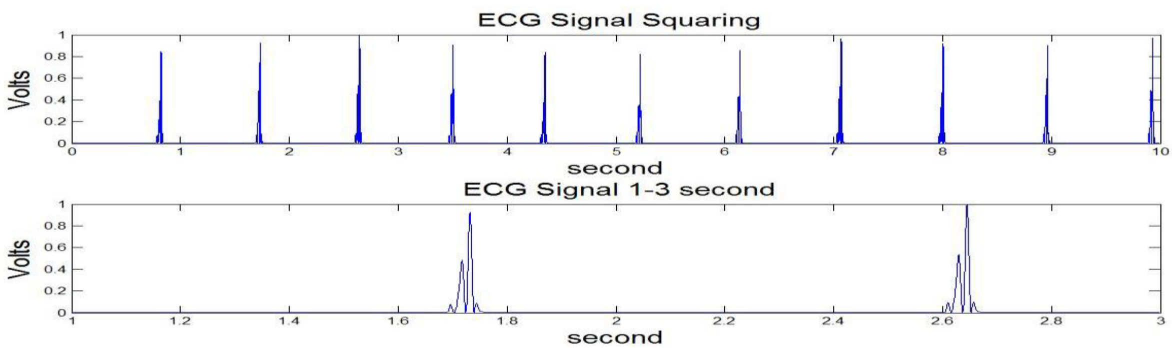

Fig.5. ECG Signal after Squaring

2.1.3 Integrating

The output of the derivative operator exhibits multiple peaks in QRS complex present in a single duration. In order to smooth the output a moving window is used as follows

$$y(n) = \frac{1}{N}[x(n-(N-1)) + x(n-(N-2))... + x(n)] \tag{6}$$

The window size N has to be chosen considering the following, large value of N would cause the outputs of the QRS complex and T wave to merge. A small value could lead to several peaks occurring for a one QRS complex. For ECG signal sampled at 200 samples/sec N=30 was found to be reasonable. Fig.6. shows the ECG signal after the moving window integration

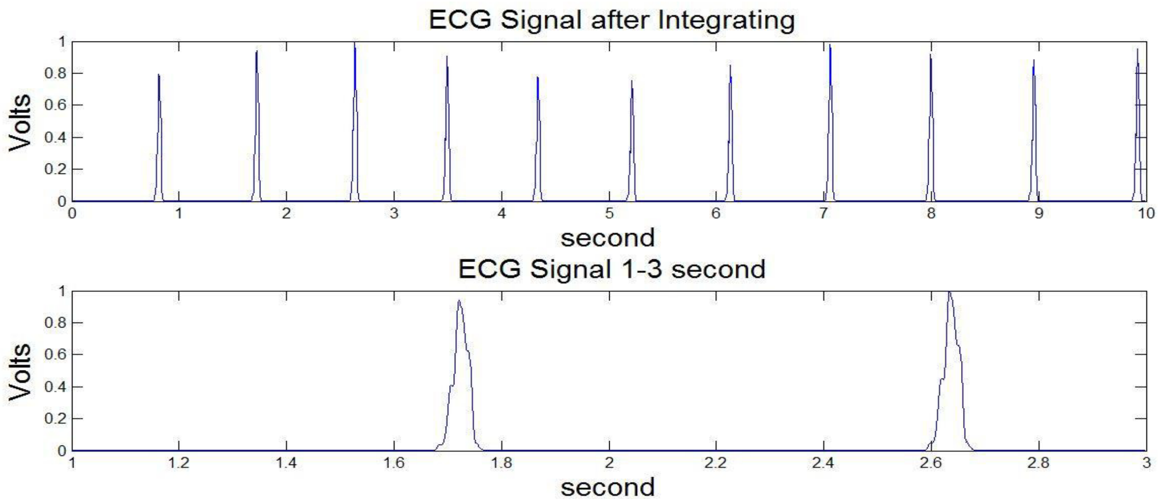

Fig.6. ECG Signal after Integrating

Adaptive Thresholding is used to determine the QRS complex and whenever the final output changes direction in an interval a peak is detected. Fig 7 shows the detection of the QRS complex after the Integrator stage of the Pan Tompkins Algorithm

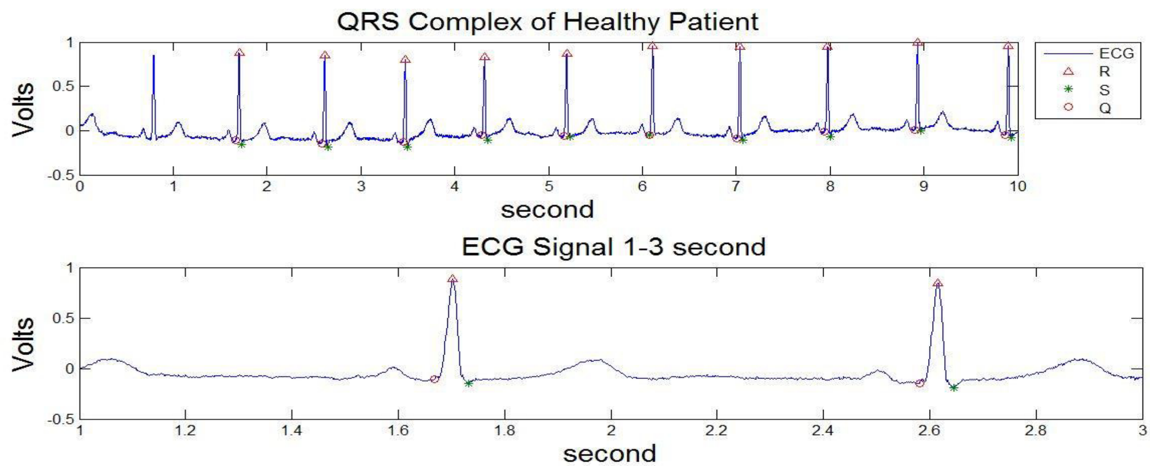

Fig.7.(a) QRS Complex of Healthy Subject after Integrating

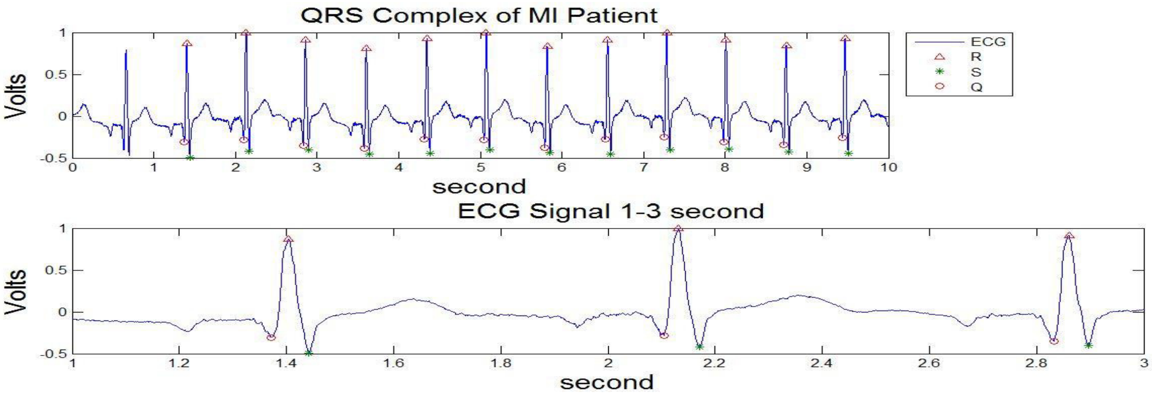

Fig.7.(b) QRS Complex of MI subject after Integrating

2.2. Second Stage: Wavelet Transformation

A time-frequency representation of the signal is done by wavelet transformation. Discrete Wavelet Transform (DWT) is used as it reduces the computation time and is easy to implement. After using digital filtering techniques on the signal a time-scaled representation is obtained. The ECG signal is passed through filters with various cut-off frequencies at numerous stages. The coefficients of the wavelets are realized by iteration of filters with rescaling. The detail information present is obtained by filtering operations. The scale is determined by upsampling or downsampling. Successive low pass and high pass filtering of discrete time-domain signal, which produces the detail and approximate coefficients through the Mallat-Tree decomposition, computes DWT. This paper utilizes four detail and approximate coefficients.

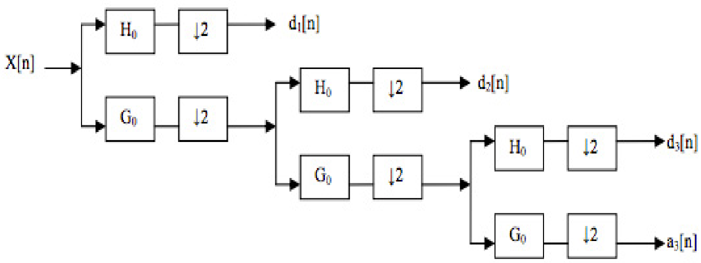

Fig.8 Mallat Tree Decomposition

Fig.8 describes an example of the Mallat Tree Decomposition in which three detail and approximate coefficients are obtained. In this paper, ECG signal is represented by X[n] sequence. G_0 represents a low pass filter and high pass filter is represented by H_0. For each level the high pass filter yields detail information d[n] and low pass filter related with the scaling function yields coarse approximation denoted by a[n]. For every decomposition level, the half band filters yields signals covering only half the frequency band. The coefficients are then subjected to statistical analysis where in median, mean, max, min, standard deviation, range, variance, mean absolute deviation are computed. Each record has 15 leads, and each lead has four detail and approximate coefficients, and for each coefficient eight statistical features are present. Since the total number of records is 449, an approximate and detail coefficient matrix of size (449 x 480) is formed.

2.3. Principal Component Analysis

The Principal Component Analysis (PCA) uses an orthogonal transformation to transform the data by a linear projection onto a lower dimensional space that preserves as much data variation as possible. The first principal component results for as much of the variability in data, and remaining variability on subsequent components. One of the main objectives of PCA is dimensionality reduction. The algorithm used for PCA is as follows

Table 2. Principal Component Analysis Algorithm

Algorithm: Principal Component Analysis	
1.	Calculate the mean and subtract mean from the input ECG data
2.	Compute the Covariance Matrix from computed matrix in Step 1
3.	Perform Singular Value Decomposition (SVD) on the Covariance Matrix (**C**) [**U**,**S**,**V**] = svd(**C**)
4.	Compute the eigenvalues of the Covariance Matrix and arrange in descending order and store in a matrix (**Eigval**).
5.	Limit the number of components to that which accounts for certain fraction of the total variance. (In this case 0.9 is used)
6.	Define variables temp and **PCA**
7.	Compute: while(temp/sum(Eigval) <= 0.9 && i<=size(V,2)) temp = Eigval(i,1) + temp PCA = [PCA U(i,1)] i = i + 1 end
8.	Project the original data by matrix multiplication of input with matrix **PCA**

The matrix approximate and detail coefficient are significantly reduced from (449 x 480) to (449 x 28) and (449 x 480) to (449 x 23) respectively.

2.4. Support Vector Machine (SVM)

SVM is a supervised learning model, which analyzes data and recognizes patterns, which is used for classification. [1] LIBSVM is used for the implementation of SVM on MATLAB. It takes an input and calculates for every given input, which of the two likely classes (Myocardial Infarction or Healthy Control) forms the output. The label indicator used for the classification is 1 for Healthy Control and 0 for Myocardial Infarction. [1] The training vectors $\mathbf{y}_i$ are represented into a superior dimensional space with the function ϕ. SVM computes a linear separating hyperplane with a maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Also,

$$K(y_i, y_j) = \phi(y_i)\phi^T(y_j) \tag{7}$$

is called the Kernel Function. The kernel used in this paper is the radial basis function kernel defined as follows, where γ is known as the Gaussian parameter

$$K(y_i, y_j) = \exp(-\gamma \| y_i - y_j \|^2), \gamma > 0 \tag{8}$$

The effectiveness of the SVM depends on the kernel parameter γ and the penalty parameter C. Grid Search is used to compute the best combination of C and γ with exponentially growing sequences [2^{-5}, 2^{-4},…, 2^4, 2^5] to obtain maximum accuracy. After the best combination of C and γ has been obtained, a 2-fold cross validation is performed to train and test the input data. For training the SVM 224 records are utilized and the remaining 225 records are used for testing the obtained model. The results of the SVM are described in Table 3, 4.

2.5. Neural Networks

Artificial Neuron consists of inputs that are multiplied with weights and then computed with the help of a mathematical function, which governs the activation of the neuron. The output of the artificial neuron is computed by a different function. Artificial Neural Networks combines Artificial Neuron to compute information. Fig.9. represents a model of an artificial neuron.

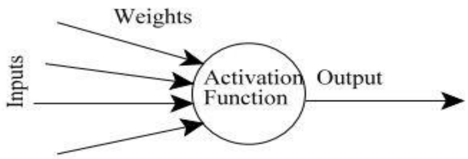

Fig.9 Artificial Neuron

Higher the weight of an artificial neuron, input is multiplied stronger. Computation of the neuron will be distinctive depending upon the weights. To obtain the output of a specific input the weights have to be adjusted. The method of regulating the weights is known as training. This paper focuses on feed-forward backpropagation algorithm. The network obtains inputs from the neurons in the input layer and the output of the network is provided to the output layer. Intermediate hidden layers may be present which is present between the layers of input and output.

The backpropagation is a supervised learning, which has inputs and outputs being provided to compute the network, and the difference between the actual and result (error) is computed. The main usage of the backpropagation algorithm is to decrease the error until the Artificial Neural Network (ANN) realises the training data. The aim is to regulate the weights so that the error is negligible wherein the training begins with random weights. A weighted sum used as the activation function of the Artificial Neuron in ANN is used to implement the backpropagation algorithm. The weighted sum is described as the sum of inputs xi multiplied with individual weights wij as follows

$$A_j(\overline{x},\overline{w}) = \sum_{i=0}^{n} x_i w_{ij} \tag{9}$$

Initial uniform weights are randomly assigned between -1 and 1 with total number of epochs set to 3000. The output function utilized in this paper is a sigmoid function, which is described as

$$O_j(\overline{x},\overline{w}) = \frac{1}{1+e^{A_j(\overline{x},\overline{w})}} \tag{10}$$

The sigmoid function has the advantage that it is near to 1 for large positive numbers, close to 0 for negative numbers and 0.5 for 0. The error function for the output is defined as

$$E(\overline{x},\overline{w},d) = (O_j(\overline{x},\overline{w}) - d_j)^2 \tag{11}$$

where d_j is the desired output. The square of the error is taken as it will be positive always and bigger in case the error is big and lesser if small. Error present in the network is the summation of all error given by

$$E(\overline{x},\overline{w},d)=\sum_j (O_j(\overline{x},\overline{w})-d_j)^2 \quad (12)$$

Using gradient descendent method, mentioned in Equation 13, the adjustment of weights are obtained after the back propagation computes how the error varies on the outputs, inputs and weights.

$$\Delta w_{ji}=-\eta\frac{\partial E}{\partial w_{ji}} \quad (13)$$

The constant eta (η) depends on the influence of the weight to error function. The momentum for weight update is set to 0.5. The Table 5 describes the performance analysis of the neural network with varying hidden layer and nodes

3. Results

The accuracy of classification in (%) during the training and testing phase of SVM is as shown below in Table 3

From the classification results it can be observed that the approximate coefficients obtained from the ECG give a slightly better classification result compared to the Detail Coefficient. The cross validation accuracy obtained during the training phase for different values of the penalty parameter C and the Gaussian kernel ɣ is described in Table 4. The overall accuracy obtained during the training phase for the Approximate Coefficient is 78.93722% ± 2.015554 whereas for the Detail Coefficient it is 82.34594% ± 2.26224.

Table 3. SVM Classification Results

Phase	Number of records	Approximate Coefficients	Detail Coefficients
Training	224	82.1429	86.6071
Testing	113	84.9558	83.1858
Validating	112	91.0714	90.1786
Optimum C		8	2
Optimum ɣ		0.5	2
Number of Support Vectors		86	106

The performance of Neural Networks is obtained by computing the Sensitivity, Specificity and Overall Accuracy.

Sensitivity (SE) is calculated by the equation

$$SE=\frac{TP}{TP+FN} \quad (14)$$

Table 4. Comparison of Classification Accuracy with varying C and ɣ

Penalty Parameter C	Gaussian Parameter ɣ	Approximate Coefficient Accuracy (%)	Detail Coefficient Accuracy (%)
0.0625	0.0625	77.2321	81.6964
0.0625	16	77.2321	81.6964
0.125	0.125	77.2321	81.6964
0.125	16	77.2321	81.6964
0.25	0.0625	77.2321	81.6964
0.5	0.5	77.2321	82.1429
1	1	80.3571	82.5893
1	2	80.3571	83.0357
2	1	80.8036	85.2697
2	2	81.25	86.6071
8	0.5	82.149	77.6786
Mean		78.93722	82.34594
Standard Deviation		2.015554	2.26224

Specificity (SP) is calculated by the equation

$$SP = \frac{TN}{TN + FP} \tag{15}$$

The overall Accuracy (ACC) is calculated by the equation

$$ACC = \frac{TN + TP}{TN + TP + FN + FP} \tag{16}$$

where is TN (True Negative), TP (True Positive), FN (False Negative), FP (False Positive). The results of the neural network are provided in table 5 wherein the network is trained for different number of hidden layer, varying the number of nodes and analyses of the overall accuracy, Specificity and Sensitivity is performed. MATLAB 7.1 was used to perform the analysis. In this case the analysis was performed on two and three hidden layers by varying the number of nodes in each layer. The accuracy of classification for the detail and approximate coefficients is provided in table 5.

4. Conclusion

The overall performance of the both the SVM and the neural Networks show that the SVM performs better classification of the subjects with the approximate coefficients providing 91.0714 % overall accuracy during validation and 90.1786% overall accuracy for detail coefficient. The neural network overall accuracy for the classification compared to the SVM is comparatively low with a maximum overall accuracy of 82.14% for a network with 3 hidden layers for the approximate coefficient and 78.1% for detail coefficient.

Table 5. Neural Network Performance Analysis

		Approximate Coefficient			Detail Coefficient		
Hidden Layer	Number of Nodes	ACC	SE	SP	ACC	SE	SP
2	30 40	81.2	83.9	64.5	75.8	77.8	68.2
	50 60	77.2	78.2	70.9	74.1	75.	68.9
	70 80	79.1	79.7	74.19	74.55	76.5	65.8
3	16 10 4	82.14	83.93	70.94	75.44	76.5	70.7
	30 40 50	77.67	78.23	74.19	75.44	75.9	73.1
	50 40 30	78.1	79.7	67.7	78.1	79.7	70.7

Acknowledgements

This research was undertaken as part of the Artificial Intelligence and Exploration coursework at Rochester Institute of Technology. Valuable guidance and support from Eyup Cinar and Dr. Ferat Sahin helped in the successful completion of Project.

References

1. Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Goldberger AL, Amaral LAN, Glass L, Hausdroff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit and PhysioNet: Components of a New Research Resource for Complex Physiologic Signal. Circulation 101(23):e215-e220 2000 (June 12) PMID: 10851218;doi: 10.1161/01.CIR.101.23.e215.
3. B. Devine and P. W. Macfarlane, “Detection of Electrocardiographic left ventricular strain using neural nets,” Med. Biol. Eng. Comput, 31, pp. 343-348, 1993.
4. Y.H. Hu, W. J. Tompkins, J. L. Urristi, and X. A. Valtino, “Application of artificial neural networks for ECG signal detection and classification,” Journal of Electrocardiology, 26, pp. 66-73, 1993.
5. R. Silipo, G.Bortolan, and C.Marchesi, “Supervised and unsupervised learning for diagnostic ECG classification,” IEEE Eng. in Medicine & Biology, pp.1054, 31 Oct.-3 Nov., Amsterdam, The Netherlands, 1996
6. Sugiura, T. Hirata, H. Harada, Y. and Kazui, T., Automatic discrimination of arrhythmia waveforms using fuzzy logic”, Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 20, no. 1, 1998
7. Acharya U R, Subbanna Bhat P, Iyengar S S, Rao A and Dua S,“Classification of heart rate using artificial neural network and fuzzy equivalence relation”, Pattern Recognit., vol. 36, pp. 61–68, 2003
8. M.H. Song, J.Lee, S.P. Cho, K.J. Lee, and S.K. Yoo, “Support Vector Machine Based Arrhythmia Classification Using Reduced Features,”International Journal of Control, Automation, and Systems, vol. 3, no.4, pp. 571-579, Dec. 2005
9. Arif, M.; Malagore, I.A.; Afsar, F.A., "Automatic Detection and Localization of Myocardial Infarction Using Back Propagation Neural Networks," Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on , vol., no., pp.1,4, 18-20 June 2010