Journal of Mechanics in Medicine and Biology
Vol. 11, No. 4 (2011) 897–915

DOI: 10.1142/S0219519411004101

World Scientific
www.worldscientific.com

ARRHYTHMIA DISEASE DIAGNOSIS USING NEURAL NETWORK, SVM, AND GENETIC ALGORITHM-OPTIMIZED *k*-MEANS CLUSTERING

ROSHAN JOY MARTIS and CHANDAN CHAKRABORTY*

School of Medical Science and Technology
Indian Institute of Technology, Kharagpur, India
**chandanc@smst.iitkgp.ernet.in*

Received 29 July 2010
Revised 5 September 2010
Accepted 12 October 2010

This work aims at presenting a methodology for electrocardiogram (ECG)-based arrhythmia disease detection using genetic algorithm (GA)-optimized *k*-means clustering. The open-source ECG data from MIT-BIH arrhythmia database and MIT-BIH normal sinus rhythm database are subjected to a sequence of steps including segmentation using R-point detection, extraction of features using principal component analysis (PCA), and pattern classification. Here, the classical classifiers viz., *k*-means clustering, error back propagation neural network (EBPNN), and support vector machine (SVM) have been initially attempted and subsequently m-fold ($m = 3$) cross validation is used to reduce the bias during training of the classifier. The average classification accuracy is computed as the average over all the three folds. It is observed that EBPNN and SVM with different order polynomial kernel provide significant accuracies in comparison with *k*-means one. In fact, the parameters (centroids) of *k*-means algorithm are locally optimized by minimizing its objective function. In order to overcome this limitation, a global optimization technique viz., GA is suggested here and implemented to find more robust parameters of *k*-means clustering. Finally, it is shown that GA-optimized *k*-means algorithm enhances its accuracy to those of other classifiers. The results are discussed and compared. It is concluded that the GA-optimized *k*-means algorithm is an alternate approach for classification whose accuracy will be near to that of supervised (viz., EBPNN and SVM) classifiers.

Keywords: ECG; MIT-BIH database; arrhythmia; MIT-BIH normal sinus rhythm; PCA; *k*-means; neural network; support vector machine; genetic algorithm.

1. Introduction

Recently, it is reported that the deaths due to noncommunicable diseases is approximately 32 million, where as more than half of these are attributed (16.7 million) due to cardiovascular diseases (CVD).[1] About more than one-third of these deaths occur in middle-aged adults. Even though CVD rates are declining in high-income and

*Corresponding author.

developed countries, they are increasing virtually every other part of the world.[2] Arrhythmia is one kind of CVD that is preventable and most frequently occurring abnormality of heart. If it is not therapeutically intervened at proper time, it may lead to severe conditions such as fibrillations and flutter that have higher mortality rates. The arrhythmia is not caused by the abnormality of heart muscles, but due to impulse conduction problems.[3] However, such type of abnormality should essentially be detected at its early stage with higher precision to reduce mortality rate. In view of this though the most basic, but important clinical test viz. electrocardiogram (ECG) is given priority for imaging cardiac abnormalities. It provides understanding about the anatomical (structural) as well as physiological (functional) causes for different cardiac diseases. Usually, the cardiologist looks at the ECG and infers the abnormality and diagnoses the cause. However, in developing countries (such as India), due to huge population, there is a need for rapid screening of patients by automatic means, in order to provide diagnostic with consistent accuracy (or efficiency) and reduce the time of the cardiologist. The work presented in this paper attempts to provide such an automated aid for diagnosis by differentiating arrhythmic patients from that of normal patients.

Generally, feature extraction is performed before classification. The features are further classified into different classes using automated classifiers. There are many methods on data classification.[4–6] One of the traditional and ancient algorithms for classification is k-means clustering, which is proposed by MacQueen.[7] It is an unsupervised classification method that uses Euclidean distance as criterion function for optimization. Another classifier is error back propagation neural network (EBPNN)[8] that uses gradient descent algorithm for updating its weights. Another important statistical learning technique called support vector machine (SVM)[9,10] is considered here to provide robust classification, which minimizes the structural risk[16] as well. There are some classifiers that modify their structure heuristically called as evolutionary algorithms.[12] A subclass of evolutionary algorithms whose principle is borrowed from natural genetics is called genetic algorithms (GA).[13] There are many articles[14,15] for a comprehensive survey of GA.

This paper introduces an application of ECG characterization for arrhythmia and normal sinus rhythm. The ECG signal from MIT-BIH arrhythmia and MIT-BIH normal sinus rhythm database are subjected to R-point extraction. From the detected R point, 200 point signal window is chosen for further analysis. The large dimensionality (200 dimensions) is reduced to lower by principal component analysis (PCA). The features obtained from PCA are subjected to classification by different classifiers, viz. k-means, EBPNN, SVM, and GA-optimized k-means clustering. In order to reduce the bias in choosing the training set, m-fold cross validation ($m = 3$) is performed and furthermore average accuracy of all folds is calculated as average classifier accuracy.

The main contribution of this paper is to device a methodology for ECG classification and the use of GA in simple classifier's optimization viz., k-means clustering

so as to increase the performance close to that of other supervised classifiers. Application of GA to other classifier optimization such as fuzzy c-means, Gaussian mixture model (GMM) classifier etc. can be extended.

Section 2 provides materials, Sec. 3 provides methodology, Sec. 4 provides the results and discussions, and Sec. 5 concludes the article.

2. Materials

In the present retrospective study, we have used MIT-BIH normal sinus rhythm and arrhythmia open-source database from www.physionet.org. The MIT-BIH normal sinus rhythm database includes 18 long-term ECG recordings of subjects referred to the Arrhythmia Laboratory at Boston's Beth Israel Hospital (now the Beth Israel Deaconess Medical Center). Subjects were found to have had no significant arrhythmias; they include 5 men, aged 26–45, and 13 women, aged 20–50. The signals are digitized at 250 samples per second.

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. 23 recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well represented in a small random sample. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10-mV range.

3. Methodology

The methodology presented here consists of QRS detection using Pan Tompkins algorithm and segmentation of ECG about R point into a 200-point window. Based on the segmented 200 samples, PCA is applied on the features and thereafter three classical pattern classifiers viz., k-means, EBPNN, and SVM are compared toward classifying arrhythmia from normal sinus rhythm. Particularly the parameters of k-means algorithm are optimized using GA. It is remarkably observed that the GA-optimized k-means classifier improves its accuracy comparable with other classifiers. Hence, the GA provides new paradigm of using simple unsupervised classifiers for robust classification, instead of supervised classifiers. In this study, GA is suggested as an alternate approach for using neural network and SVM (Fig. 1).

3.1. *R-point detection and segmentation*

Many methods[11,17] have been reported in the literature for the QRS complex detection in the ECG. The most effective is the one proposed by Pan and Tompkins,[17]

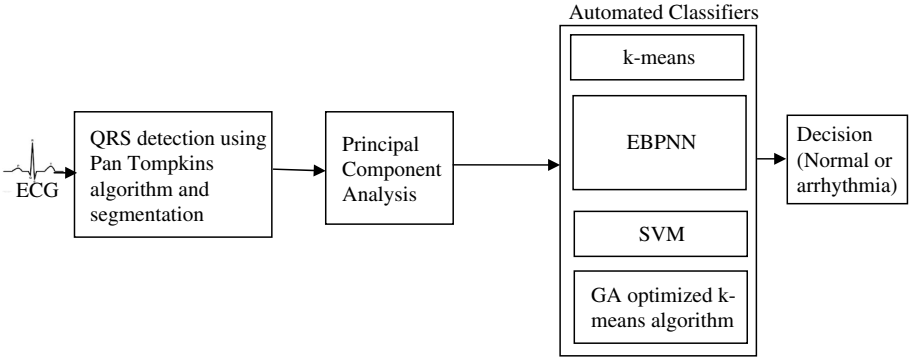

Fig. 1. System overview of the proposed methodology.

which consists of taking derivatives, rectifying the derivatives, and smoothing by moving average system. It makes the use of slope information embedded in the derivatives. High-frequency noise is removed by smoothing operation. The author's earlier works[18–20] made use of R point for registration and subsequently the features are extracted and classified. A simplified version of Pan Tompkins algorithm is given below.

Step 1: Compute the absolute value of first difference of the ECG signal.
Step 2: Smooth the signal by passing through moving average system defined as follows.

$$y(n) = \frac{1}{4}\{x(n) + 2x(n-1) + x(n-2)\}, \tag{1}$$

where $x(n)$ and $y(n)$ are input and output for the moving average system.
Step 3: Compute the derivative of smoothened signal from Step 2 and compute its absolute value.
Step 4: Smooth the signal obtained from Step 3 through a moving average system given by Eq. (1).
Step 5: Sum the two signals obtained from Steps 2 and 4.
Step 6: Threshold the signal with an adaptive threshold to get rectangular pulses; the middle point of these pulses correspond with the R point.

After QRS extraction, 99 samples to the left of R point and 100 samples to the right of R point including R point itself are chosen such that approximately it will include one period of ECG wave.

3.2. *Feature compression using PCA*

Once the ECG data is segmented, its dimensionality is reduced using PCA.[21] The method projects the data in a direction where it is represented best in least square sense. The method involves computation of eigenvalues and eigenvectors of the

covariance matrix of the segmented ECG signal. Since the eigenvectors are orthonormal to each other, they serve as the computational basis for the system. The eigenvectors are sorted in the descending order of corresponding eigenvalues, and the data is projected along the directions of these eigenvectors by taking dot product between the eigenvectors and ECG data vectors. The projected components of data are called as principal components (PCs). The first few PCs will represent most of the data variability and, therefore, only first few PCs are retained and used for subsequent classification. The algorithm consists of following steps.

Step 1: Compute data covariance matrix Σ as

$$\Sigma = \frac{1}{N}\{(d - \bar{d})(d - \bar{d})^T\}, \tag{2}$$

where $\bar{d}$ corresponds to the mean feature vector and d represents the data matrix.

Step 2: Compute the eigenvector matrix V and a diagonal matrix having eigenvalues D such that

$$V^{-1}\Sigma V = D. \tag{3}$$

Step 3: Sort the eigenvectors in the descending order of eigenvalues in D.

Step 4: Project the data into PCs by taking the dot product between the given data and PCs.

Step 5: Choose first few PCs depending on containment of maximum variability of the ECG signal.

3.3. *k-means classification*

The k-means clustering algorithm was proposed by MacQueen.[7] It uses k-seed points as initial k centroids ($k = 2$ in our study). Then, it consists of three basic operations performed iteratively, viz. data assignment to a cluster, centroid (cluster mean vector) computation, and test for algorithm convergence. In the data assignments step based on the current centroid, the data are partitioned into different classes based on minimum mean squared error (MSE) criterion. In the centroid calculation step, based on the data points present in a cluster, the average pattern is computed as the sum of all the patterns in a class divided by the total number of patterns. In each iteration, the algorithm is tested for convergence by computing the difference in the centroids of current iteration with the immediate past iteration. If the difference is more than a predefined threshold the iterations are continued, otherwise it is stopped. The algorithm minimizes the following objective (cost) function.

$$J = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2, \tag{4}$$

where x_j and μ_i indicate the jth pattern in the ith centroid respectively.

3.4. *EBPNN classifier*

We have used an EBPNN in this study, which consists of input layer of neurons, one hidden layer, and an output layer. The weights of each layer are updated using gradient descent algorithm such that the MSE between the actual output of the neural network and the desired output (i.e., network targets or training labels) is minimized. The algorithm aims at minimizing the following objective function.

$$J = \frac{1}{2}\sum_{n=1}^{N}\sum_{i=1}^{k}\{y_i(x_n) - t_i^n\}^2, \tag{5}$$

where k represents the total number of classes present in the data (in our case $k = 2$), $y_i(x_n)$ represents the ith neuron output in the output layer of the neural network for the nth input pattern. Since the data is mean centered after PCA, a bias term is not needed in the network architecture. During training process, the total MSE reduces below a threshold. The testing data is provided to the trained neural network and the network output is noted in order to compute the classification accuracy of the classifier.

3.5. *Support vector machine (SVM)*

SVM is a single layer and highly nonlinear network, which minimizes structural risk and has higher generalization ability in the sense that it can classify data correctly. It optimizes the class separation boundary such that the distance from a feature to the class separating hyperplane is maximum simultaneously. Suppose if $(x_i, y_i), i = 1 : N$ are the N observation (or patterns), x_i is the ith input and y_i is the corresponding pattern label, for the two class pattern classification problem, c_+ and c_- are the centroids of the two classes, the classifier response is given by,

$$y_i = \text{sgn}\{(x_i - c)\cdot w\} = \text{sgn}\{x_i \cdot c_+ - x_i \cdot c_- + b\}, \tag{6}$$

where

$$b = \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2). \tag{7}$$

The hyperplane that is optimal in separating the data points into two classes and satisfying condition in Eqs. (6) and (7) will be

$$\underset{w,b}{\text{minimize}}\,\frac{1}{2}\|w\|^2 \quad \text{such that } y_i(w\cdot x_i + b) \geq 1, \quad i = 1, \ldots, N. \tag{8}$$

Equation (8) is a minimization problem, which is also a quadratic optimization problem. In order to solve this problem, one must find the saddle point of the Lagrange function

$$L_p(w, b, \alpha) = \frac{1}{2}\|w\| - \sum_{i=1}^{n}\alpha_i\{y_i(w^T x_i + b) - 1\}, \tag{9}$$

where α_i are called Lagrange multipliers under the constraint $\alpha_i \geq 0$. L_p is minimized in order to find the optimal saddle point with respect to primal variables

w and b. This problem is transformed into the dual form by differentiating L_p with respect to w and b and introducing Karush Kuhn Tucker[22,23] conditions. The transformed dual problem is the minimization problem of the following objective function.

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j,$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{10}$$

For linearly separable data, Eq. (10) is useful. However, when the data is linearly nonseparable, kernel trick is considered to transform the feature space into a higher dimensional space to make the data linearly separable. In practice, we need not have to map the input variables into the high dimensional space directly. Instead the inner product between the features in the kernel space could be used in the optimization problem. The dual problem of optimization with kernel transformation for SVM is

$$\text{Maximize } L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right),$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{11}$$

where $k(x_i, x_j) = \{\varphi^T(x_i) \bullet \varphi(x_j)\}$, which is the dot product between the transformed features. In fact any symmetric function that satisfy the Mercer conditions[10] can be used as a kernel function. Commonly used kernel functions are polynomial, quadratic, and radial basis function (RBF) kernels. The polynomial kernel function of degree d is

$$k(x_i, x_j) = (x_i^T x_j + 1)^d. \tag{12}$$

We have used second- and third-degree polynomial kernels in our analysis. In linear kernel transformation $k(x_i, x_j) = x_i^T x_j$, which is the inner product between the features as can be seen from Eq. (11). The RBF kernel transformation is

$$k(x_i, x_j) = \exp\left\{\frac{\|x_i - x_j\|}{2\sigma^2}\right\}, \tag{13}$$

where σ is the width parameter of RBF kernel.

3.6. *GA optimized k-means approach*

The optimization problem that of k-means classification indicated in Eq. (4), is implemented as a local optimization problem and is a single sample-based optimization. Different initial centroids lead to different final cluster centers due to the fact that the algorithm converges to a local minimum. Consideration of population-based optimization is a good choice and always leads to global minima

of the objective function. GA is a kind of evolutionary algorithm that is population based, derivative-free optimization strategy, which always yields global optimum value. In this study, we have used a GA, which uses three basic operators called selection (or reproduction), crossover, and mutation. The algorithm uses the principles borrowed from natural genetics. Unlike conventional search and optimization algorithms, GA starts its search from a random set of solutions. In the context of GA, the metric used to represent the distance is called as fitness function, which gives relative importance for every population. The GA is depicted in Fig. 2 and explained as follows.

3.6.1. *Coding and decoding of populations*

The GA operates in a binary coded string space than in the actual real values of the objective function. Therefore, the optimization variables (in our case, it is the centroids) are to be coded in binary string form. A fixed number of bits are used to represent the centroid values. According to the 13 features, we have 13 decision variables to be coded into binary form. Since the centroids are floating point values, with $x_i^{\max}$ and $x_i^{\min}$ as the upper and lower bounds of the ith feature, the decoded

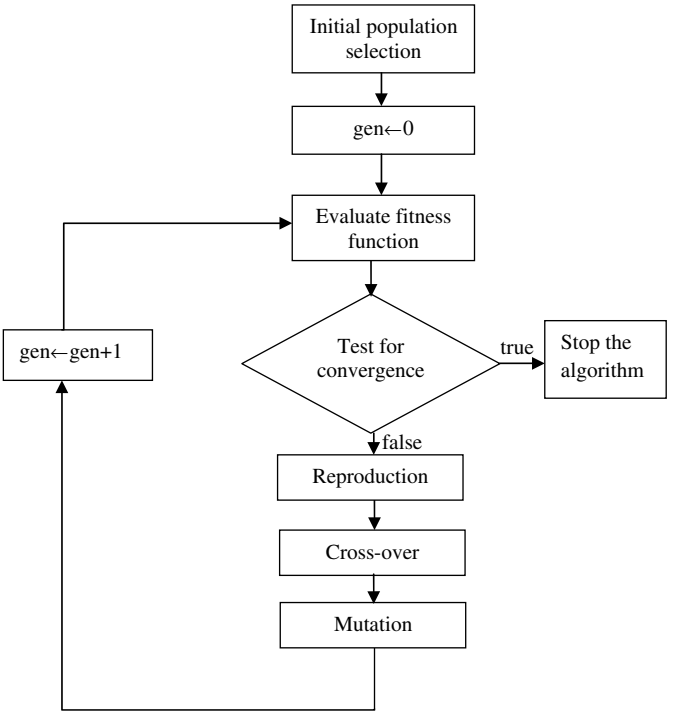

Fig. 2. GA-based optimization of k-means clustering centers.

value of the centroid is given as,

$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^{l_i} - 1} DV(s_i), \tag{14}$$

where l_i represents the string length to code the ith decision variable, $DV(s_i)$ is the decoded value of the string s_i. Different decision variables in an optimization problem can have different number of bits and precisions. The coding of decision variables leads to have a pseudo-chromosomal representation of the solution.

3.6.2. *Evaluation of fitness function*

While the centroid value (the decision variable) is represented by a binary string, it is required to evaluate the solution, in the light of the underlying objective and constraint functions. Since we do not have any constraints in this case, the fitness of a string is assigned a value that is a function of the solution's objective function value. Generally, in the unconstrained optimization, the fitness is made equal to the objective function value. From the given populations, the objective function of Eq. (4) is computed after decoding the binary strings of populations. Since the optimization problem is the minimization of actual fitness function, the function to be maximized is

$$M = \frac{1}{J}. \tag{15}$$

3.6.3. *Reproduction*

The objective of reproduction is to duplicate the fitter strings and discard the less fitter or bad solutions, while keeping the population size constant. This is done by identifying the fitter solutions, making multiple copies of these strings, discard the bad solutions in the population and replace them with the fitter solutions. In analogy with the natural genetics, the fitter individuals only survive in the environment. Same thing is true with the centroids represented as strings in this case. There are many methods used in reproduction. Here, proportionate selection method is used. Here, the solutions are assigned copies, the number of which is proportional to their respective fitness values. If the average fitness of all the populations in a generation is f_{avg}, and f_i is the fitness of one particular solution, then the strings of this solution will get an expected number of solutions equal to f_i/f_{avg} number of copies. This is implemented using a roulette wheel mechanism. The wheel is divided radially into different regions depending on the fractional fitness of average fitness. The total area of the wheel is average fitness and individual regions will have an area equal to $f_i * f_{\text{avg}}$. The new generation strings are generated by sampling the roulette wheel on probabilistic basis. On the basis of different areas, a probability density function is estimated and accordingly a random number is generated, and accordingly reproduction operation is performed.

3.6.4. *Cross-over*

Since reproduction operator does not create any new solutions, it only copies the solutions in the population having higher fitness. Cross-over and mutation are the two operators, which generate newer populations. In cross-over operation, two strings are randomly chosen and substrings are exchanged between strings to generate newer strings. We have used single point cross-over operation. A crossing site in the strings is chosen randomly and the string is broken into two parts at the cross-over site. Two pieces of strings from two different original strings are merged to get a new string in the next generation.

3.6.5. *Mutation*

In this operation, some random changes in the bits of the strings of solutions are done. Usually, the mutation probability is kept very low as compared to the natural genetics. The need for mutation is to keep diversity in the population. The mutation operator alters a string locally to hopefully create a better string. A random number is generated for every bit in a string, and accordingly the bits are flipped with a given mutation probability.

3.6.6. *Termination*

The three operators are used iteratively until the fitness function becomes steady in the sense that its value does not change in the new iterations. In this case, the GA is said to be converged. Another criterion used for termination is that if the GA exceeds a particular number of iterations. The solution provided by GA at its convergence will be close to the global optimum of the objective function.

3.7. *m-Fold cross validation*

We have employed m-fold cross validation[25] with $m = 3$. Here, the total number of samples are sub-sampled into three (m) sets, one set is used for testing whereas the other two sets are used for training the classifier. The process is repeated for two more times such that every sub-partition is used as testing set and the rest are used for classifier training. The three accuracies are averaged to estimate the average classifier performance. Using m-fold cross validation, the bias in choosing the samples from the population can be overcome.

4. Results and Discussion

In order to apply the presented methodology, a two-class ECG classification problem is formulated based on MIT BIH arrhythmia and MIT-BIH normal sinus rhythm dataset (described in Sec. 2). Using Pan Tompkin's algorithm, the R point in the ECG is detected. The exact position of R point is tuned by compensating the

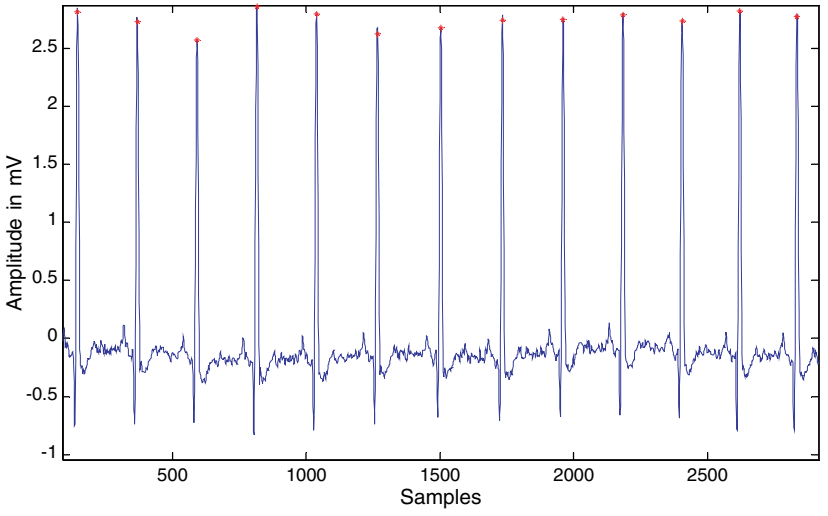

Fig. 3. The detection of R point in normal sinus rhythm signal, R point is highlighted in red asterisk.

group delays of the involved filters in Pan Tompkins algorithm. The choice of Pan Tompkins method is due to its simplicity and accurate identification of beats. The detection of R point is shown in Fig. 3, the detected R point is shown with red asterisk. It can be seen from the results that Pan Tompkins algorithm detects the R point with good precision. In fact Pan Tompkins algorithm is a multistage filtering (difference, smoothing, etc.) along with a nonlinear operation (rectification) in the algorithmic steps.

Once the R point is detected, 100 samples to the right of R point, 99 samples to the left of R point, and the R point itself are chosen to form one segment of ECG beat for subsequent feature extraction and classification. These 200 samples in the data are reduced to lesser number of samples using PCA technique. PCA projects the data on to the directions of maximum variability. The highest eigenvalue corresponds to the variance of the data in the direction of maximum variability. The subsequent eigenvalues are lower in magnitude and eventually the magnitude reduces with the data dimension. It is observed from Fig. 4 that 13 PCs contribute 99.7% variability of the data, whereas the rest of the 0.03% variability is contributed by the other 187 directions. Therefore, first 13 PCs are chosen for subsequent pattern classification.

It is seen from Table 1 that first eigenvalue corresponds to 86.23% variability in the data, whereas second eigenvalue corresponds to 10.94% of variability in the data. The contribution by other directions is even lesser and in the 13th component, the energy is 0.02% of the total signal energy. Hence, first 13 features are considered.

The features after PCA are used for further pattern classification using four classifiers, viz., k-means, EBPNN, SVM, and GA-optimized k-means classifiers. The

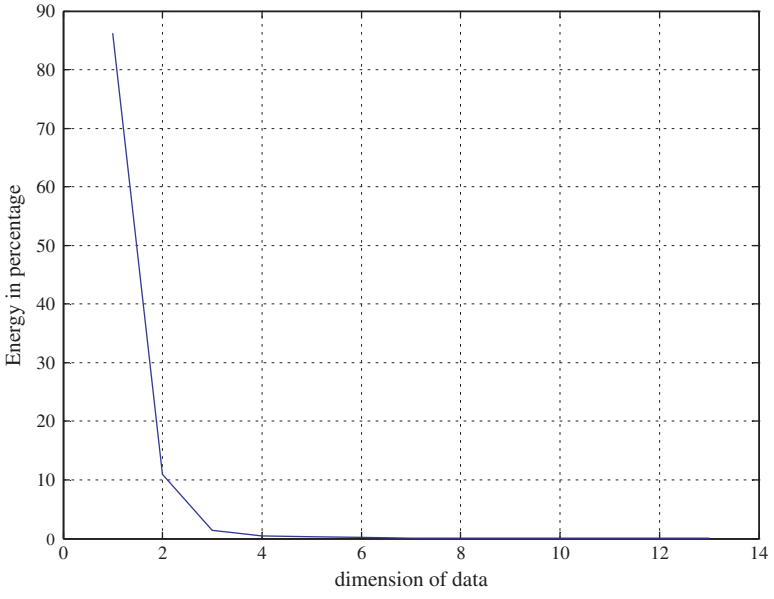

Fig. 4. The energy profile of PCs with respect to data dimension.

Table 1. Energy profile of PCs.

PC Index	Eigenvalues	Percentage of energy contained
1	13.1985	86.2342
2	1.6750	10.9438
3	0.2021	1.3204
4	0.0591	0.3864
5	0.0403	0.2636
6	0.0235	0.1533
7	0.0163	0.1067
8	0.0118	0.0773
9	0.0102	0.0663
10	0.0084	0.0546
11	0.0054	0.0350
12	0.0052	0.0342
13	0.0042	0.0272

reduction of MSE by the neural network is shown in Fig. 5. The neural network is used in batch mode, when all the samples in the training data are fed to the neural network, it is said to be one epoch. As the epochs iteratively progressed, the error is propagated backwards so as to update the network weights such that the MSE decreases. In Fig. 5, we can observe that the MSE is decreasing with the epochs. A predefined threshold on MSE is defined and if the MSE reduces below this threshold, the iterations are stopped and the algorithm is said to be converged. The neural network converges in 53 epochs and the threshold chosen is 10^{-6}.

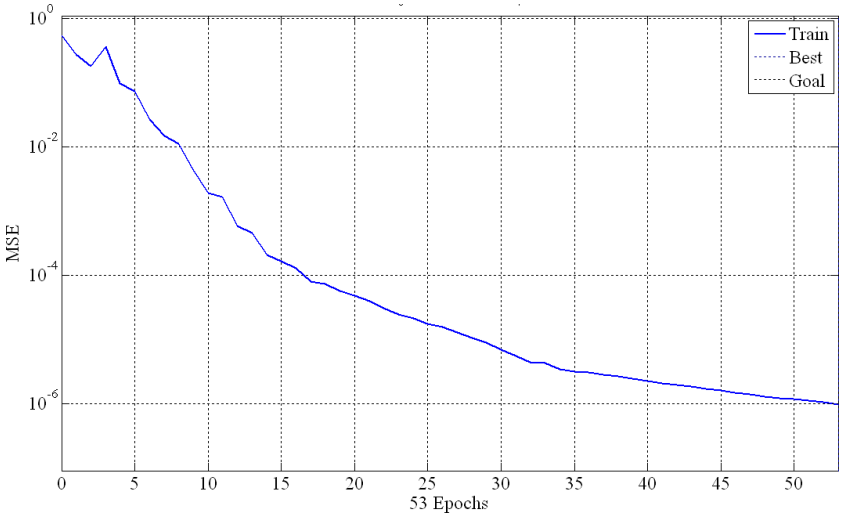

Fig. 5. Training of EBPNN classifier, MSE is decreasing with iterations.

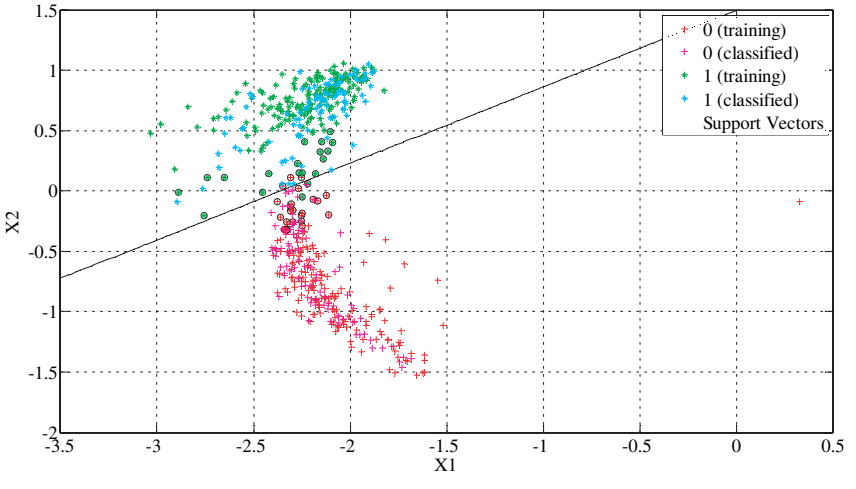

Fig. 6. SVM classification using linear kernel.

The SVM classification of ECG using a linear kernel is shown in Fig. 6. The SVM is a highly nonlinear statistical classifier that uses a kernel function to map the data into a high dimensional space where it is assumed to be linear. Here, the class separating hyperplane is optimized such that the distance of the data sample points to the discriminant hyperplane gets maximized simultaneously. Since we have used a linear kernel, the discriminant hyperplane is looking like a straight

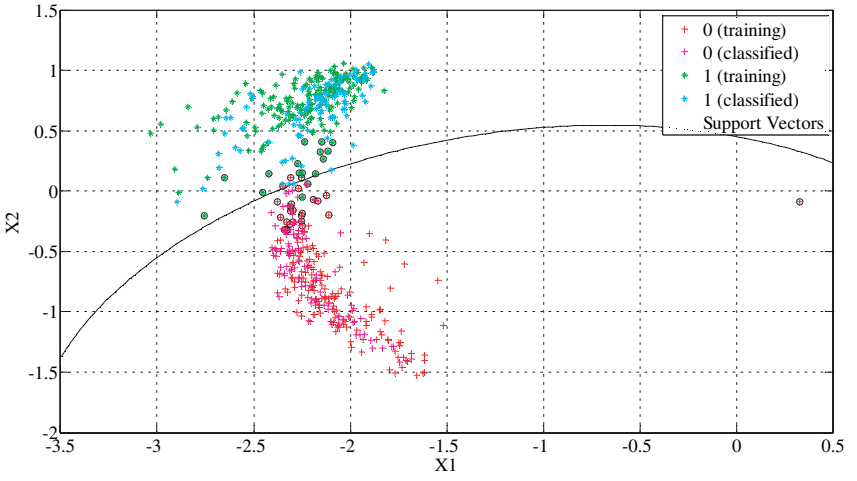

Fig. 7. SVM classification with second-degree polynomial kernel.

line as seen from Fig. 6. From this and as discussed in Sec. 3.5, the discriminant hyperplane is linear. Figure 7 shows SVM classification of ECG data using second-degree polynomial kernel. Since the kernel function is quadratic, it can be seen that the discriminant hyperplane is looking like a parabola. Since parabola is more nonlinear than a straight line, we can conclude that the discriminant hyperplane is more complex. Figure 8 shows SVM classification of ECG data using third-order polynomial kernel function. Since the polynomial degree is higher, we can see that

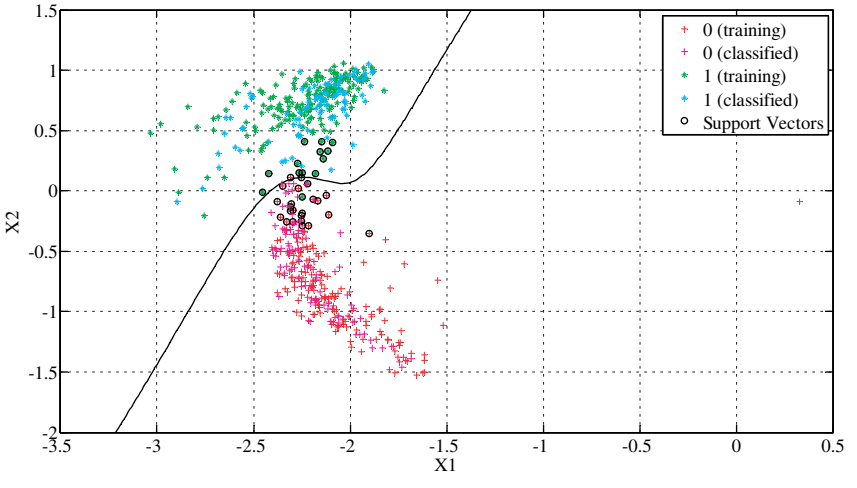

Fig. 8. SVM classification with third-degree polynomial kernel.

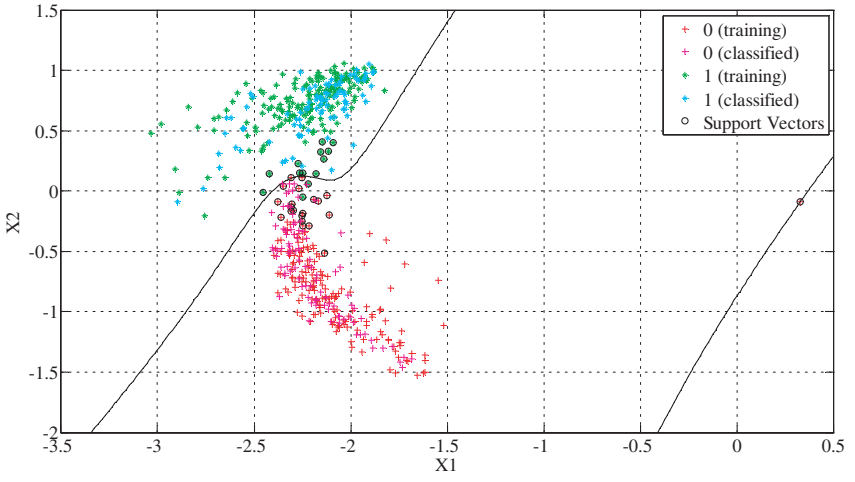

Fig. 9. SVM classification with fourth-degree polynomial kernel.

the discriminant hyperplane is more complex since it is having more number of maxima and minima (i.e., one maxima and one minima as against only one maxima in case of second-degree polynomial) as can be seen from the results. Figure 9 shows SVM classification with fourth-degree polynomial, where discriminant hyperplane is further complex. In the same way, Fig. 10 shows SVM classification with fifth-degree polynomial, where the discriminant hyperplane is still more complex as can be seen from the many maxima and minima in the discriminating curve.

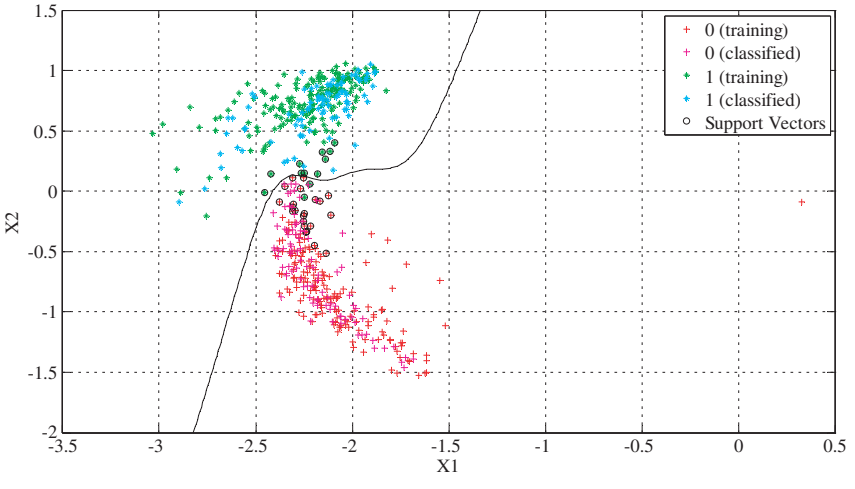

Fig. 10. SVM classification with fifth-degree polynomial kernel.

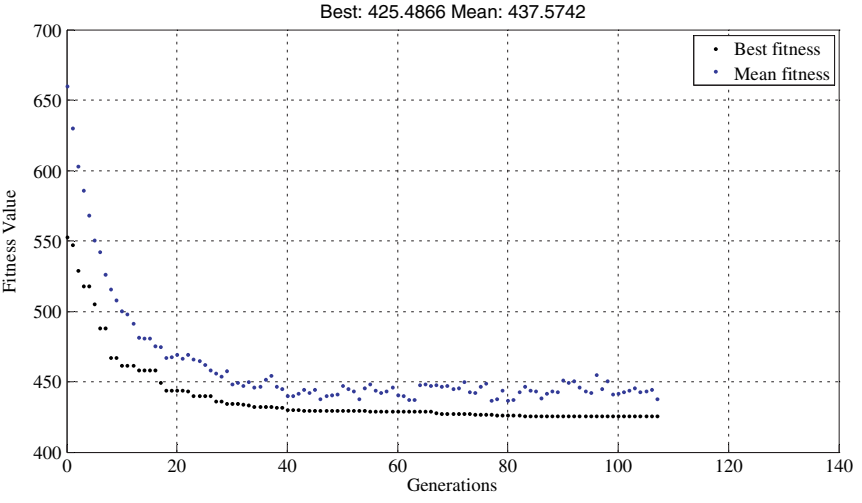

Fig. 11. GA classification, fitness value is shown decreasing with the generations.

The GA-optimized k-means classifier is used to classify the ECG, whose results are shown in Figs. 11–13. Figure 11 shows the best and mean fitness values in each generation. It is observed that both best and mean fitness value decrease as progression of generations, since our optimization problem of classification is minimization problem of the objective function. Figure 12 shows the average distance between the individuals in every generation. It is expected that as the generations

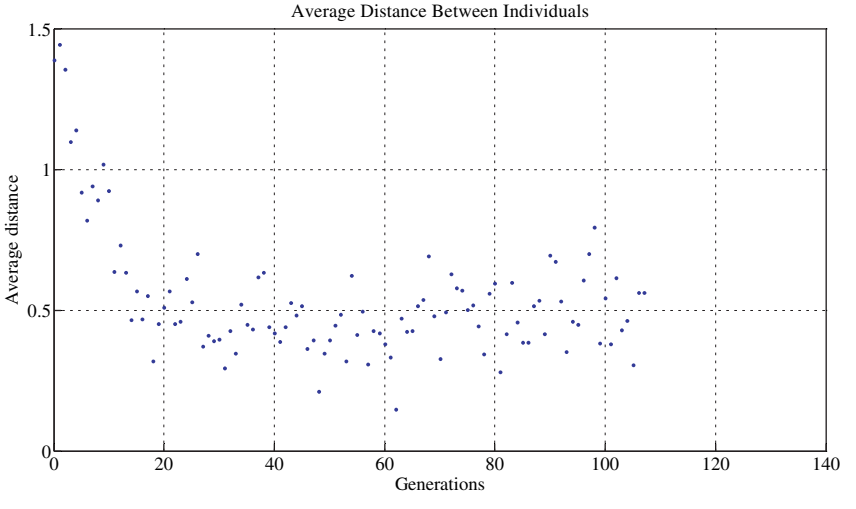

Fig. 12. GA classification, average distance between individuals shown decreasing with generations.

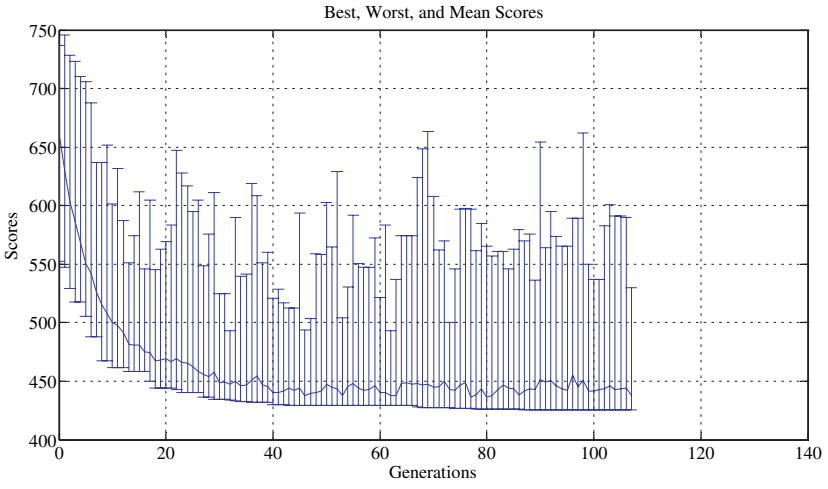

Fig. 13. The best, worst, and average scores in every generation in GA classification.

Table 2. Classification accuracy of various classifiers.

	Accuracy (%)			
	Fold 1	Fold 2	Fold 3	Average
k-Means	91.2088	92.3077	90.1099	91.2088
Neural network	96.1538	97.2527	95.0549	96.1538
SVM-linear kernel	96.1538	97.8022	95.6044	96.5201
SVM-polynomial-2	95.7875	96.1538	96.5201	96.1538
SVM-polynomial-3	96.7033	97.2527	97.8022	97.2527
SVM-polynomial-4	97.2527	96.1538	97.2527	96.8864
SVM-polynomial-5	96.7033	97.8022	95.0549	96.5201
GA-optimized k-means	96.1538	96.1538	95.6044	95.9707

progress, it should lead fitter generations and, therefore, the mean distance between the individuals should decrease. Hence, the GA will converge. Figure 13 shows the best, worst, and average scores of the fitness function in every generations of the GA. As the generations evolve, the scores will decrease since our problem is the minimization problem of the objective function.

The classification accuracy of various classifiers is shown in Table 2. It is observed that neural network and SVM provide higher accuracy in comparison with basic k-means algorithm. If the objective function of k-means algorithm is optimized by GA, then we observe that its performance can reach to near to that of neural network and SVM classifiers.

5. Conclusion

This paper provides a methodology for the classification of ECG belonging to normal sinus rhythm and arrhythmia classes. Different classifiers are used for

classification. It is observed that k-means algorithm provides lower accuracy whereas supervised classifiers (viz., EBPNN and SVM with different kernels) provide higher accuracy. It is also seen that if k-means algorithm is optimized by GA it yields performance near to that of supervised classifiers. In addition, different polynomial kernel functions are used in SVM classification. All polynomial kernel functions (i.e., second, third, fourth, and fifth degree) yield almost similar results. Polynomial kernel of third degree provides slightly higher accuracy, but the difference is very feeble so as to compare. Again as seen from k-fold cross-validation, different runs of classifier training yield different accuracy. In some runs, it provides more and some runs it provides slightly lower accuracy. Therefore, we cannot say which polynomial kernel provides the highest accuracy since the difference is less than 1% of accuracy. We conclude that EBPNN, SVM with different polynomial kernels, and GA-optimized k-means algorithm all provide almost similar results.

As a future direction, one can optimize other unsupervised classifiers such as fuzzy c-means classifier and GMM in the same methodology using GA and one can see how the accuracy changes by the incorporation of GA. In addition, some of the new and efficient variants of GA can be used in order to reduce the computational burden. Moreover, as a future direction, various new and novel operators can be used in order to catalyze the convergence of GA.

Acknowledgment

Authors are extremely grateful to VECC, Kolkata for financial support and SRIC, IIT Kharagpur for valuable aid.

References

1. Park K, *Park's Textbook of Preventive and Social Medicine*, 18th edn., Banarsidas Bhanot Publishers, 2005.
2. Fauci AS, Braunwald E, Kesper DL, Hauser SL, Longo DL, Jamesonn JL, Loscalzo J, *Harrison's Principles of Internal Medicine*, 17th edn., McGraw Hill, 2008.
3. Guyton AC, Hall JE, *Textbook of Medical Physiology*, 11th edn. W. B. Saunders Co., 2006.
4. Jain AK, Dubes RC, *Algorithms for Clustering Data*, Eagle-Wood Cliffs, NJ, Prentice-Hall, 1988.
5. Jain AK, Murthy MN, Flynn PJ, Data clustering: a review, *ACM Comput Surv* **31**:264–323, 1999.
6. Xu R, Wunsch D, Survey of clustering algorithms, *IEEE Trans Neural Networks* **16**(3):645–678, 2005.
7. MacQueen JB, Some methods for classification and analysis of multivariate observations, *Proceedings of* 5th *Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, University of California Press, Berkeley, CA, USA, 1967, pp. 281–297.
8. Bishop C, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
9. Vapnik VN, *Statistical Learning Theory*, Wiley Interscience, September, 1998.
10. Christianini N, Taylor JS, *An Introduction to Support Vector Machines and Other Kernel Based Learning Methods*, Cambridge University Press, 2000.

11. Gunn S, *Support Vector Machines for classification and Regression*, Technical Report, University of Southampton, May, 1998.
12. Hruschka ER, Campello RJGB, Freitas FA, Carvalho ACPLF de, A survey of evolutionary algorithms for clustering, *IEEE Trans. Syst Man Cybern C: Appl Rev* **39**(2): 2009.
13. Goldberg DE, *Genetic Algorithms in Search, Optimization and Machine Learning*, 3rd Impression, Pearson Education, 2008.
14. Kalyanmoy Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Ltd., 2001.
15. Naldi MC, Carvalho ACPLF de, Campello RJGB, Hruschka ER, Genetic clustering for data mining, In Maimon O, Rokach L, (eds.), *Soft Computing for Knowledge Discovery and Data Mining*, New York, Springer-Verlag, pp. 113–132, 2007.
16. Martinez JP, Almeida R, Olmos S, Rocha AP, Laguna P, A wavelet based ECG delineator: evaluation on standard databases, *IEEE Trans Biomed Eng* **51**(4):570–581, 2004.
17. Pan J, Tompkins WJ, A real time QRS detection algorithm, *IEEE Trans Biomed Eng* **32**(3):230–236, 1985.
18. Banik S, Martis RJ, Nayak D, Code excited linear prediction for electrocardiogram, *Proc IEEE Eng Med Biol Soc* **1**:160–163, 2004.
19. Martis RJ, Chakraborty C, Ray AK, A two stage mechanism for registration and classification of ECG using Gaussian mixture model, *Pattern Recognit* **42**(11):2979–2988, 2009.
20. Martis RJ, Krishnan MM, Chakraborty C, Pal S, Sarkar D, Mandana KM, Ray AK, Automated screening of arrhythmia using wavelet based machine learning techniques, *J Med Syst* 2010 (In Press).
21. Duda R, Hart P, Stork D, *Pattern Classification*, 2nd edn., John Wiley & Sons, 2001.
22. Chong EKP, Zak SH, *An Introduction to Optimization*, 2nd edn., Wiley Interscience, 2001.
23. Ravindran A, Ragsdell KM, Reklaitis GV, *Engineering Optimization Methods and Applications*, 2nd edn., Wiley Interscience, 2006.
24. Gunn AM, Gupta MK, Dasgupta B, *Fundamentals of Statistics*, 4th edn., World Press Private Ltd., Vols. I and II, 2008.
25. Schneider J, Cross validation, 1997, Available at: http://www.cs.cmu.edu/∼schneide/tu5/node42.html, Accessed on: July 10, 2010.