ECG beat classification using machine learning techniques

Shweta H. Jambukia, Vipul K. Dabhi* and Harshadkumar B. Prajapati

Department of Information Technology, Dharmsinh Desai University, Nadiad, Gujarat, India
Email: shweta.it2009@gmail.com
Email: vipul.k.dabhi@gmail.com
Email: harshad.b.prajapati@gmail.com
*Corresponding author

Abstract: An arrhythmia is an abnormality in the heart rhythm, or heartbeat pattern. ECG beats can be classified into six different arrhythmia beat types (left bundle branch block, right bundle branch block, paced beats, premature ventricular contraction, atrial premature beats, and normal rhythm). Early and accurate detection of arrhythmia types is important in detecting heart diseases and choosing appropriate treatment for a patient. This paper proposes a combination of Particle Swarm Optimisation (PSO) and Feed Forward Neural Network (FFNN) for ECG beat classification. We have used MIT-BIH arrhythmia database for data collection and prepared three different datasets. Features, such as R peak sample number and QRS complex, are extracted using Pan-Tompkins algorithm. The extracted features are used as inputs to three different classifiers: Multi-Layer Perceptron Neural Network (MLPNN), Support Vector Machine (SVM), and PSO-FFNN. Results show high classification accuracy of over 97% with either of these three classifiers. The performance comparison of these classifiers is carried out using three measures: sensitivity, specificity, and accuracy. The results suggest that PSO-FFNN shows slightly better performance than MLPNN and SVM in terms of accuracy on all datasets.

Keywords: ECG beat classification; particle swarm optimisation; neural network; support vector machine; RR interval; QRS complex.


Biographical notes: Shweta H. Jambukia received her Master’s degree in Information Technology from Dharmsinh Desai University in 2015. Her research interests include application of machine learning techniques in biomedical field.

Vipul K. Dabhi is an Associate Professor at Department of Information Technology, Dharmsinh Desai University, India. He received his PhD in Computer Engineering at Dharmsinh Desai University, Nadiad, Gujarat, India in year 2014. He received a Bachelor’s degree in Instrumentation & Control Engineering from Gujarat University in 2000 and Master’s degree in Computer Engineering from Dharmsinh Desai University in 2006. His research interests...
ECG beat classification using machine learning techniques

Harshadkumar B. Prajapati is an Associate Professor at Department of Information Technology, Dharmsinh Desai University, India. He received his PhD in Computer Engineering at Dharmsinh Desai University, Nadiad, Gujarat, India in year 2014. He received a Bachelor’s degree in Electronics & Communication Engineering from Gujarat University in 2000 and Master’s degree in Computer Engineering from Dharmsinh Desai University in 2007. His research interests include grid computing, operating system and data analytics. He has published more than 40 research papers in international journals and conferences proceedings.

1 Introduction

An electrocardiogram (ECG) is a diagnostic tool that measures and records the electrical activity of the heart of a patient. ECG is useful in finding the cause of chest pain and detecting abnormal heart rhythm or cardiac abnormalities. Normal healthy hearts have a cardiac ECG shape. Any irregularity in the heart rhythm can change the shape of ECG signal.

Classification of ECG beats plays an important role in detecting various heart diseases. Early and accurate detection of ECG arrhythmia helps doctors to detect various heart diseases. ECG beat classification is a challenging problem due to following reasons (Singh et al., 2012; Bashir et al., 2010; Jambukia et al., 2015): variability in normal ECG waveform of each person, dissimilar signs for one disease on different patients’ ECG waveform, two distinct diseases may have approximately similar effects on different patients’ ECG waveform, lack of standardisation of ECG features, and non-existence of optimal classification rules for ECG beat classification.

Researchers have used different pre-processing techniques, feature extraction techniques, and classification techniques for ECG beat classification. For pre-processing (noise removal), various filters are used such as low pass linear phase filter, linear phase high pass filter, and median filter. For feature extraction, techniques such as Discrete Wavelet Transform (DWT), and Pan-Tompkins algorithm are used. For normalisation of features, Z-score and unity Standard Deviation (SD) techniques are used. The following classifiers are used by research community for ECG classification: Artificial Neural Network (ANN), Support Vector Machine (SVM), Fuzzy C-Means clustering (FCM), and ID3 decision tree.

This paper proposes a combination of Particle Swarm Optimisation and Feed Forward Neural Network (PSO-FFNN) for ECG beat classification. NN randomly initialises the values of weights and biases. This may causes poor mapping of inputs to outputs. PSO is used for optimising the values of weights and biases of NN for accurate ECG classification. Moreover, we also applied Multi-Layer Perceptron Neural Network (MLPNN) and Support Vector Machine (SVM) for classification of ECG beats. We have used MIT-BIH arrhythmia database for collecting raw ECG data. Pan-Tompkins
The proposed algorithm is applied on raw ECG data for noise removal and features extraction. Extracted features are used as inputs to three classifiers (PSO-FFNN, MLPNN, and SVM).

The remainder of this paper is organised as follows: Section 2 gives introduction to ECG waveform and different types of arrhythmia. The proposed approach is presented in Section 3. Section 4 shows experimental setup. The results and discussion are presented in Section 5. The conclusions are given in Section 6.

2 Background knowledge and related work

This section gives introduction to ECG waveform, various features of ECG waveform, and different types of arrhythmia. The related work to ECG beat classification is also presented.

2.1 Background knowledge

One cardiac cycle of normal ECG signal is shown in Figure 1. A ECG signal consists of: (a) P, Q, R, S, T, and U peaks, (b) PR, RR, QRS, ST, and QT intervals, and (c) PR and ST segments, shown in Figure 1. R peak, RR interval, and QRS complex are the basic features of ECG signal. R peak is the second most upward movement of the ECG tracing. RR interval is the interval between two consecutive R peaks. QRS complex normally begins with a downward deflection Q, a larger upward deflection R, and ends with a downward S wave.

Figure 1 Normal ECG waveform

Normal healthy hearts have a cardiac ECG shape. Any irregularity in the rhythm of heart (amplitude, duration, and shape of rhythm) can change the shape of ECG signal. There can be many types of abnormalities present in the ECG signal. An arrhythmia is an abnormality in the heart rhythm, or heartbeat pattern. The heartbeat can be too slow, too fast, have extra beats, or skip a beat. Left Bundle Branch Block (LBBB), Right Bundle Branch Block (RBBB), Premature Ventricular Contraction (PVC), Atrial Premature Beats (APB), and Paced Beats (PB) are different types of arrhythmia. The various ECG features (R peak, QRS interval, RR interval, RR ratio, RR mean), discussed above, are useful to classify ECG beats into these arrhythmia types.
The LBBB arrhythmia has following characteristics compared to normal ECG signal: value of R peak is high, value of RR interval is smaller, and QRS complex is wider, shown in Figure 2(a). The RBBB arrhythmia has following characteristics compared to normal ECG signal: value of R peak is smaller, value of RR interval is larger, and QRS complex is smaller, shown in Figure 2(b). The PVC arrhythmia has following characteristics compared to normal ECG signal: value of R peak is high, value of RR interval is smaller, and QRS complex is wider, shown in Figure 3(a).

The APB arrhythmia has following characteristics compared to normal ECG signal: values of PP interval and QRS complex are smaller, shown in Figure 3(b). APB signal contains sequence of P wave and inverted P wave. In PB, RR interval and QRS complex are wider than normal ECG signal. Also, S wave is deeper and T wave is larger than normal signal. Looking at the characteristics of different arrhythmia, it can be concluded that to classify ECG beats into one of arrhythmia types (LBBB, RBBB, PVC, APB, and PB), we need to extract basic ECG features such as R peak, RR interval, and QRS complex from ECG waveform.

2.2 Related work

Many researchers have worked on classification of ECG signals. They have applied various pre-processing, feature extraction, and classification techniques. Most of the researchers have used MIT-BIH arrhythmia database as a standard ECG database. ECG data can be classified into two ways: (i) ECG beat classification, and (ii) ECG signal classification. Only few researchers have worked on signal classification and it is more difficult compared to beat classification because normal ECG signal may differ for each person, sometimes one disease has dissimilar signs on different ECG signals, and two
distinct diseases may have approximately identical effects on ECG signals. Jambukia et al. (2015) surveyed different databases available for ECG classification and issues involved in ECG classification. Different preprocessing techniques, features extraction techniques, and classifiers are also analysed in Jambukia et al. (2015).

Figure 3  (a) Premature ventricular contradiction arrhythmia (image source: LearntheHeart, 2014a, 2014d), (b) Atrial premature beat arrhythmia (image source: Manual, 2014)

Dallali et al. (2011a) have extracted RR interval using Discrete Wavelet Transform (DWT) technique and used Fuzzy C-Means clustering (FCM) to classify ECG beats. They achieved 99.05% accuracy. Dallali et al. (2011b) applied DWT for extracting the following features: location of R point and RR interval. FCM is used for preclassification and 3-layer MLPNN is used for final beat classification. They achieved accuracy of 99.99% using MLPNN. R peak and RR interval are extracted using DWT in Ayub and Saini (2011). MLPNN is used for ECG beat classification. They achieved Mean Squared Error (MSE) of 0.00621. Khorrami and Moavenian (2010) have extracted RR interval using DWT. Performance of MLPNN and SVM is compared. The obtained results show that MLPNN outperforms SVM on test dataset.
Patra et al. (2010) have manually extracted R peak from the annotation file of the MIT-BIH arrhythmia database. R peak and RR interval are normalised using zero mean and reduced using FCM. Three layers FFNN trained with back propagation algorithm is used for ECG beat classification. Al-Naima and Al-Timemy (2009) have applied DWT at fourth level to extract features of ECG beats. They have used Db4 and Haar wavelets as mother wavelets. They have discarded level 1 and level 2 detail coefficients by arguing that these coefficients contain noise. The extracted DWT coefficients were given as inputs to NN classifier. The sensitivity, specificity and accuracy reported by them are 90%, 90% and 95%. Khazaei (2013) has used both morphological and timing features for classification of ECG beats. The morphological features used are: amplitude of P-peak, amplitude of Q-valley, amplitude of R-peak, amplitude of S-valley and amplitude of T-peak. The used timing features are: position of P-peak, position of Q-valley, position of R-peak, position of S-valley and position of T-peak. Other than these features, they also considered time interval between next and previous beats and time ratio between last beat to the next beat as features. A combination of PSO and SVM was proposed for ECG beat classification. Their proposed approach aimed to improve the performance of SVM in two aspects: optimal selection of feature subset and SVM model parameter optimisation. Moavenian and Khorrami (2010) compared the performance of MLP, trained with backpropagation algorithm, and SVM, trained with Kernel-Adatron algorithm, for ECG signal classification. ECG signals available from MIT-BIH arrhythmia database were used to train these classifiers. Results show that MLP performs good on test dataset while SVM requires less training time and shows good performance on training dataset.

Vishwa et al. (2011) segmented continuous ECG signal into individual ECG beats. For segmentation, they utilised the annotation file provided by MIT-BIH database. The beats were centred on R peak and have 300 sample data. The different structures of ANN were trained to classify arrhythmic and non-arrhythmic data. They reported an accuracy of 96.77% on MIT-BIH database and 96.21% on NSR database. In Korürek and Dogan (2010), R peaks of the ECG beats were detected using Pan-Tompkins algorithm. The ECG beats were collected by considering 128 points from each side of R peaks. Four morphological features: $t_{RR}$ interval (current RR interval at t time), $t_{RR+1}$ interval (next RR interval at t+1 time), $QR_{S}$ (QRS complex height), and QRS-width were extracted from ECG beats. A combination of PSO and RBFNN was proposed for classification of six types of beats (Korürek and Dogan, 2010). Sensitivity and specificity achieved using PSO-RBFNN classifier for ECG beat classification are 96.25% and 99.10% respectively.

A combination of Independent Component Analysis (ICA) and neural networks (Probabilistic and back-propagation) is proposed for ECG beat classification in Yu and Chou (2008). ICA is used to decompose the ECG signal into basic components which are statistically independent to one another. ICA-based features along with RR interval are used as input feature vector to neural networks. Experiments were performed on eight ECG beat types, selected from MIT-BIH arrhythmia database. Yu and Chou (2008) concluded that PNN gives better results than BPNN in terms of accuracy and robustness to number of ICA-bases. A combination of wavelet transform and probabilistic neural network (PNN) is proposed in Yu and Chou (2007) to classify six different types of ECG beats. The original QRS complex signals are decomposed into different subbands using two-level discrete wavelet transform. Haar wavelet is selected as the mother wavelet. Three categories of features, energy, coherence and morphological, are extracted from
the signal in each subband. The obtained classification accuracy of 99% for all type of
ECG beats indicates effectiveness of their proposed approach. The problem of reducing
the dimensionality of ECG classification problem is addressed in Mar et al. (2011). Mar
et al. (2011) noted that the computational requirements of search methods (which try to
select the best subset of available features) increase with increase in the number of
features. The Sequential Forward Floating Search (SFFS) algorithm was applied to find
optimal subset of features. The optimal subset of features was given as input to
multilayer perceptron (MLP).

Many researchers have used Pan-Tompkins algorithm for ECG beat detection (QRS
complex detection). Faezipour et al. (2009) proposed the modification in the original
Pan-Tompkins algorithm to reduce computational complexity of the algorithm. The
modified algorithm requires only one set of threshold, derived from denoised ECG
signal, as compared to two sets of thresholds required by the original algorithm.
Faezipour et al. (2009) reported an accuracy of 99.51% on MIT-BIH arrhythmia
database. A Swarm based Support Vector Machine (SSVM) technique is proposed for
detecting Premature Ventricular Contraction (PVC) in Nuryani et al. (2014). PSO is used
to optimise the parameters of SVM. The width and gradient of QRS wave were given as
inputs to SSVM. Nuryani et al. (2014) concluded that SSVM with polynomial kernel
outperforms SSVM with other kernels like RBF, Sigmoid and linear.

A combination of DWT and Random Forests (RF) classifier is proposed in Alickovic
and Subasi (2016) for diagnosis of heart arrhythmia. DWT is used to extract features
from ECG signal by decomposing the signal into different components at different
frequency bands. Total 27 statistical features were extracted and used as inputs to RF
classifier. The experiments were carried out on two different databases: five ECG signal
patterns from MIT-BIH database and four ECG signal patterns from St. Petersburg
Institute of Cardiological Technics 12-lead arrhythmia database. Their proposed
approach obtained 99.33% accuracy for MIT-BIH database and 99.95% for St.
Petersburg Institute of Cardiological Technics 12-lead arrhythmia database (Alickovic
and Subasi, 2016). An expert system, based on DWT and fuzzy classifier, is designed for
ECG arrhythmia detection in Mahmoodabadi et al. (2010). Daubechies (Db6) wavelet is
used as mother wavelet. The features extracted using DWT are used as inputs of fuzzy
classifier for arrhythmia detection. Mahmoodabadi et al. (2010) claimed that fourteen
types of arrhythmias and abnormalities can be detected using their proposed expert
system.

3 Proposed approach

This section presents the proposed approach for ECG beat classification. It explains the
implementation steps of the proposed approach in detail.

3.1 Proposed system for ECG beat classification

ECG beat classification consists of following four steps: data collection, pre-processing,
features extraction, and classification. The proposed approach is tested using MIT-BIH
arrhythmia database. For pre-processing and feature extraction, Pan-Tompkins algorithm
is used. PSO-FFNN is used as a classifier for classifying ECG data into various
arrhythmia types. Figure 4 shows proposed system for ECG arrhythmia classification.
Firstly, raw ECG signal, collected from MIT-BIH arrhythmia database, is passed through Pan-Tompkins algorithm. Pan-Tompkins algorithm removes noise and baseline wander from raw ECG signal. Then, the algorithm extracts the QRS duration and R peak sample number. R peak is obtained from the signal.dat file of MIT-BIH database using R peak sample number as an index. Then RR interval is calculated from the timings given in signal.dat file. From RR interval, RRR interval, RR mean, RR ratio, and average 10 RR interval are calculated. These seven features are given as inputs to the PSO-FFNN classifier. PSO-FFNN classifies the ECG data into different classes (arrhythmia types) based on these seven input features.

\[ R_{\text{mean}} = RR(i) - \overline{RR} \]  
\[ R_{\text{ratio}}(i) = \frac{RR(i)}{RR(i-1)} \]  
\[ RRR_{i-1}(i) = RR(i) - RR(i-1) \]

The architecture that integrates PSO and Artificial Neural Network (ANN) for ECG classification is shown in Figure 5. In the proposed system, PSO algorithm is used for optimisation of weights and biases of FFNN. The architecture comprises the following main steps: PSO is initialised with random position and velocity vectors. Now for each particle’s position, fitness (Mean Square Error) is evaluated. MSE is the squared difference of generated output (output generated by NN for each particle’s position according to current global best (gbest) values of that particle - current best weights and biases values) and desired output. Personal best fitness is set as pbest and best of all pbest is set as global best (gbest). Particles’ velocity and position are updated according to its personal and social experience as given in equations (4) and (5).

\[ v_{x_{d}}(t+1) = \omega \cdot v_{x_{d}}(t) + c_1 R_1 (p_{x_{d}}(t) - x_{d}(t)) + c_2 R_2 (p_{\text{gbest}}(t) - x_{d}(t)) \]  
\[ x_{d}(t+1) = x_{d}(t) + v_{x_{d}}(t+1) \]

where, \( v_{x_{d}}(t+1) \) is the updated velocity, \( \omega \) is the inertia weight, \( R_1 \) and \( R_2 \) are random numbers in range of \([0...1]\), which introduce useful randomness for the search strategy. \( c_1 \) is the cognitive positive constant weight parameter, \( c_2 \) is the social positive constant weight parameter, \( p_{x_{d}}(t) \) is the historically best position of the \( i \)-th particle in the \( d \)-th dimension, \( p_{\text{gbest}}(t) \) is the global best position in the \( d \)-th dimension, \( x_{d}(t) \) is the current position of the \( i \)-th particle in the \( d \)-th dimension.
This procedure is repeated till maximum number of iterations is reached. The optimised values obtained from gbest are used as values of weights and biases of NN as shown in equation (7) to train the FFNN. The trained FFNN is tested on test dataset.

3.2 Feature extraction technique: Pan-Tompkins algorithm

Pan-Tompkins algorithm (Pan and Tompkins, 1985) consists of following five steps: band-pass filtering, differentiation, squaring, moving window integration, and thresholds adjustment. Figure 6 shows these steps.

Band pass filter reduces noise (like baseline wander and T wave interference) from ECG signal. It is composed of cascaded high pass and low pass Butterworth IIR filters. Derivative operator finds the high slopes that normally distinguish the R peak from other ECG waves (P, Q, S, T and U wave) and suppresses the low frequency components such as P and T waves. Squaring operation performs point by point squaring of ECG signal. It makes the result positive and emphasise large differences resulting from QRS complexes. Squaring is used for further enhancing high frequency components and suppressing the small differences arising from P and T waves. Integration sums the area under the squared waveform over a suitable interval. It advances one sample interval and integrates the new predefined interval window. It extracts the slope of the R wave. Signal to noise ratio increases after the ECG signal has passed from the band pass filter. So, threshold adjustment is done and sensitivity of the algorithm is improved.
Pan-Tomkins algorithm has following advantages compared to other available feature extraction techniques: (i) sensitivity and efficiency of Pan-Tomkins algorithm is more than 99% (Pan and Tompkins, 1985), (ii) required computational efforts are less. Moreover, Pan-Tomkins algorithm has inbuilt steps for noise removal and baseline wander removal.

3.3 ECG classification models

This section describes PSO-FFNN, MLPNN, and SVM classifiers used for ECG beat classification.

3.3.1 PSO-FFNN classifier

Particle Swarm Optimisation is a population-based search strategy that finds optimal solutions using a set of particles. The velocities of these particles are adjusted according to their historical performance and performance of their neighbours, in the search space (Ahmed and Glasgow, 2012). PSO is suitable to solve problems whose solutions can be represented as a set of points in an n-dimensional solution search space. The term particles refer to population members, which are described as the swarm positions in the n-dimensional solution space. Each particle is set into motion through the solution space with a velocity vector representing the particle’s speed in each dimension. Each particle has a memory to store its historically best position. Therefore, the moving particles, at each iteration, evaluate their current position with respect to the fitness function to be optimised. The moving particles compare their current position with their historically best position, as well as to the positions of other individuals of the swarm. Then, each particle updates its experience (if the current position is better than its historically best one), and adjusts its velocity according to global best particle by moving closer towards it. The position of global best particle is updated before the end of each iteration of PSO. If the position of any of particle in the entire swarm is better than the current position of the global best particle then the index of the swarm’s global best particle is updated. The parameter ‘inertia weight’ is used to control the global exploration ability of PSO and to provide a balance between the global and local search abilities.

In proposed system, PSO algorithm is used for optimisation of weights and biases of FFNN. Figure 7 shows the architecture of FFNN with 7 inputs and 3 outputs. Perceptrons are arranged in layers, with the first layer (input layer) taking in inputs and the last layer (output layer) producing outputs. The middle layer is known as the hidden layer. Each
perceptron of one layer is connected to every other perceptron of the next layer. Each node is connected to one or more other nodes by real-value weights, but not to nodes in the same layer. No feedback loop is present in feed forward neural network. The following equations are used to calculate output of hidden and output layers:

\[
Q_h = \frac{1}{1 + e^{-I_h}}
\]

(6)

\[
I_h = \sum_{i=1}^{n} w_{ij} x_i + b_h
\]

(7)

where, \( w_{ij} \) is the weight between input and hidden layer or between hidden and output layer, \( x_i \) is the input feature vector, and \( b_h \) is the bias.

**Figure 7** Architecture of FFNN used in proposed PSO-FFNN classifier

To evaluate the performance of PSO-FFNN classifier, the following measures are used: sensitivity, specificity, and accuracy. Sensitivity is the ratio of true positive beats to total of true positive and false negative beats. Specificity is the ratio of true negative beats to total of true negative and false positive beats. The overall accuracy is the ratio of total number of true negative and true positive beats to total number of beats.

\[
Sensitivity = \frac{TP}{TP + FN}
\]

(8)

\[
Specificity = \frac{TN}{TN + FP}
\]

(9)

\[
Accuracy = \frac{TN + TP}{TN + FN + FP + TP}
\]

(10)
3.3.2 MLPNN classifier

MLPNN trained with back-propagation algorithm is also used as a classifier for ECG beat classification. MLPNN is same as the FFNN, but it has two or more number of hidden layers. MLPNN is trained with Levenberg-Marquardt back-propagation algorithm for experimental purpose. The network training process is performed by providing input-output data to the network, which targets to minimise the error by optimising the network weights. The neural network toolbox of MATLAB is used for performing experiments of MLPNN. We have measured the classification accuracy given by MLPNN.

3.3.3 SVM classifier

SVM is a kind of supervised learning method used for classification and regression. SVM is primarily a classifier that performs classification task by constructing hyperplane in a multidimensional space that separates different class labels. SVM aims at minimising the empirical classification error and maximising the geometric margin. SVM map input vector to a higher dimensional space where a maximal separating hyperplane can be constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separate the data. The separating hyperplane is the hyperplane that maximise the distance between the two parallel hyperplanes. SVM uses the idea of kernel substitution which is broadly referred as kernel methods. There are many kernel functions available in SVM. However, selection of good kernel function is a challenging task. However, there are some popular kernel functions such as linear kernel, RBF kernel and sigmoid kernel. Performance of SVM classifier depends on choice of kernel function and configuration parameters. For the implementation of SVM classifier, LibSVM (Libsvm, 2014) library on MATLAB platform is used. For the evaluation of SVM classifier, we have used classification accuracy as performance measure.

\[
\text{Accuracy} = \frac{\text{number of correctly classified samples}}{\text{Total number of samples}}
\]  

(11)

3.4 Implementation of proposed work

A user-friendly GUI for feature extraction and ECG beat classification is developed in MATLAB environment, which is shown in Figure 8. To load the raw ECG signal in GUI, go to File > open menu. After loading raw ECG signal, press ‘Denoise’ button to remove baseline wander and noises from the ECG signal. To extract features using Pan-Tompkins algorithm, go to Feature extraction > Pan-Tompkins menu. It extracts the R peak sample number and QRS duration as shown in the middle panel of GUI in Figure 8. The seven (R peak, QRS interval, RR interval, RRR interval, RR mean, RR ratio, and average 10 RR interval) different features can be generated by pressing ‘Generate feature vector’ button, as shown in Figure 8. The GUI provides checkboxes for selection of features, to be used as inputs of classifier, from the generated features. The selected features (feature vector) are loaded into final feature vector table. After this, the output vector is loaded for classification using ‘load output vector’ button as shown in Figure 8. In experiments, outputs (classes) is represented in vector form. For example, class 1 output vector should be [1,0,0] and class 2 output vector should be [0,1,0]. After loading final feature vector and output vector, run classifiers from Classifier > PSO – FFNN menu.
Figure 8  GUI developed on MATLAB showing Final input feature vector and output vector for ECG classification

Figure 9  GUI used for setting different parameters of PSO-FFNN classifier

To set the PSO and FFNN parameters, go to ‘Classifier > PSO – FFNN’ menu and to run PSO-FFNN classifier, use ‘Run classifier’ button, as shown in Figure 9. Output of a classifier is shown in Figure 10. Figure 11 shows MSE versus iteration graph for PSO-FFNN for training dataset. It also presents confusion matrix for training and test datasets and overall confusion matrix. Class wise sensitivity, specificity, and accuracy is presented using bar chart, shown in Figure 11.
4 Experimental setup

This section discusses about available ECG database, feature extraction techniques and ECG classification.
4.1 ECG database

The MIT-BIH arrhythmia database (Physionet, 2014) is used for the experimental purpose. The database contains 48 recordings. Each record has duration of 30 minutes and includes two leads namely modified limb lead II (MLII) and any one out of the modified leads V1, V2, V4 or V5. There are three files namely signal.dat, annotation.atr and header.hea. The signal.dat is a binary file and comprises the ECG signal. The annotation.atr file reports events within the recording, such as heart beats.

We have used online PhysioBank ATM (ATM, 2014a) service for converting signal.dat and annotation.atr files into text form. The 'Show samples as text' and 'Show annotation as text' options of ATM service are used to get signal and annotation files into text form. However, the ATM service puts a limit on the amount of data that you can obtain in a single request. The limit is 60,000 samples (about a minute data). So, the other option is to obtain data into .mat format. The wfdb toolbox (ATM, 2014b) for MATLAB is used to obtain data in .mat file. The following commands of wfdb toolbox are used for converting signal.dat and annotation.atr file into .mat format: rdsamp and rdann.

We have taken a dataset which contains six different classes: normal (N), left bundle branch block (L), right bundle branch block (R), paced beat (PB), premature ventricular contraction (V), and atrial premature beat (A). We prepared three different datasets. Dataset-1 contains ECG beats from three different classes: N, L, and R. Dataset-2 contains ECG beats corresponding to four different classes: N, L, R, and PB. Dataset-1 and Dataset-2 contains whole 30 minutes of MLII signal (approximately 2000 beats). Dataset-3 contains ECG beats corresponding to six different classes: N, L, R, PB, V, and A. Dataset-3 contains 500 beats of each class. Table 1 shows arrhythmia classes, database files, and number of beats used to prepare three different datasets.

**Table 1**  
Arrhythmia Classes, Database Files and Number of Beats Used for Preparing Three Datasets. N,L,R,PB, V and A stands for normal, left bundle branch block, right bundle branch block, paced beat, premature ventricular contraction and atrial premature beat

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classes</th>
<th>MIT-BIH Files</th>
<th>No. of Beats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset I</td>
<td>N,L,R</td>
<td>100,109,118</td>
<td>2273, 2527, 2279</td>
</tr>
<tr>
<td>Dataset II</td>
<td>N,L,R,PB</td>
<td>100,109,118, 102</td>
<td>2273, 2527, 2279, 2187</td>
</tr>
<tr>
<td>Dataset III</td>
<td>N,L,R,PB,V,A</td>
<td>100,109,118, 102,200,209</td>
<td>500 beats from each class</td>
</tr>
</tbody>
</table>

4.2 Methodology

Methodology used for ECG beat classification comprises following steps:

- Removal of baseline wander and noise from the raw ECG signal using Pan-Tompkins algorithm (Pre-processing step).
- Extracting R peak sample number and QRS complex using Pan-Tompkins algorithm (Feature extraction step).
The following five features are calculated from R peak sample number: RR interval, RRR interval, RR mean, RR ratio, and average 10 RR interval.

Seven features (R peak, QRS interval, RR interval, RRR interval, RR mean, RR ratio, and average 10 RR interval) are given as inputs to three different classifiers (PSO-FFNN, MLPNN, and SVM).

Train three classifiers using training dataset. For each of three created datasets, we chosen 70% of dataset to be training data.

Apply the trained classifier on test dataset and compare the performance of these classifiers using the following measures: sensitivity, specificity, and accuracy. For each of three datasets, we chosen 30% of dataset to be test data.

5 Results and discussion

Following seven features are extracted to classify ECG beats into different classes: R peak, QRS duration, RR interval, RR mean, RR ratio, RRR interval, average of 10 RR interval. Out of these seven features, two (R peak, QRS duration) are morphological and remaining five are timing features. These features capture shape and temporal aspects of ECG beats. Generally, the selection of number of neurons in the hidden layer of NN depends on the type and complexity of classification task and size of training data. The selection of large number of neurons in the hidden layer may lead to over-fitting problem, which may result in failing to classify new input data. Therefore, we performed different experiments by varying the number of hidden neurons. PSO-FFNN, MLPNN, and SVM are experimentally tested for ECG beat classification. Comparison of these classifiers based on obtained results is presented in this section.

5.1 Experiment-1: PSO-FFNN classifier

Parameters values set for PSO-FFNN classifier, used for ECG beat classification, are as follows:

- **Processing**
  - Normalise features into range [-1,1]
  - Optimise weights and biases of FFNN using PSO
  - Classify data into different classes using FFNN

- **PSO parameters**
  - Adaptive linearly decrease PSO: We selected Adaptive linearly decrease PSO model which starts with a high inertia weight for global exploration and linearly decreases the inertia weight to allow exploitation (local explorations) of promising regions in later iterations.
  - Number of particles: 30
  - Value of cognitive positive constant $C_1$: 2.0
Value of social positive constant $C_2$: 2.0

Value of maximum inertia weight: 0.9

Value of minimum inertia weight: 0.6

Maximum number of iterations: 500

**FFNN parameters**

- Number of hidden layer: 1
- Activation function: sigmoid

Table 2 shows results obtained for PSO-FFNN classifier. We have performed different experiments by varying the number of hidden neurons (5, 10, 15) for each datasets. As shown in Table 2, PSO-FFNN gives better performance for 10 hidden neurons in terms of Sensitivity (Se), Specificity (Sp) and Accuracy (Acc). Accuracy given in the Table 2 is average of 10 runs for each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Hidden Neurons</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Se(%)</td>
</tr>
<tr>
<td>Dataset I</td>
<td>5</td>
<td>99.45</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>99.56</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>99.37</td>
</tr>
<tr>
<td>Dataset II</td>
<td>5</td>
<td>97.89</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>98.42</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>98.41</td>
</tr>
<tr>
<td>Dataset III</td>
<td>5</td>
<td>91.46</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>91.94</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>90.26</td>
</tr>
</tbody>
</table>

### 5.2 Experiment-2: MLPNN classifier

Parameters values set for MLPNN classifier, used for ECG beat classification, are as follows:

- **Processing**
  - Normalise features into range $[-1,1]$
  - Classify data into different classes using MLPNN

- **MLPNN Parameters**
  - Number of hidden layers: 1, 2, 3
  - Activation function: sigmoid (logsig)
  - Training algorithm: Levenberg-Marquardt back propagation algorithm
ECG beat classification using machine learning techniques

- Value of learning rate: 0.3
- Value of momentum: 0.2
- Number of epochs: 500

Table 3 presents results obtained using MLPNN classifier. We have performed experiments by considering different values for (a) number of hidden layers and (b) number of nodes within hidden layer. As shown in Table 3, MLPNN gives the best performance for single hidden layer comprising of ten hidden neurons in terms of Accuracy. Accuracy given in Table 3 is the average of 20 runs for each dataset.

<table>
<thead>
<tr>
<th># Hidden Layers</th>
<th># Hidden Neurons</th>
<th>Dataset I Acc(%)</th>
<th>Dataset II Acc(%)</th>
<th>Dataset III Acc(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>95.67</td>
<td>96.88</td>
<td>96.67</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>97.86</td>
<td>97.55</td>
<td>97.26</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>94.74</td>
<td>97.14</td>
<td>96.81</td>
</tr>
<tr>
<td>2</td>
<td>4,4</td>
<td>96.93</td>
<td>96.53</td>
<td>95.46</td>
</tr>
<tr>
<td></td>
<td>5,3</td>
<td>96.97</td>
<td>96.50</td>
<td>95.91</td>
</tr>
<tr>
<td></td>
<td>8,8</td>
<td>97.58</td>
<td>96.8</td>
<td>96.37</td>
</tr>
<tr>
<td>3</td>
<td>4,4,3</td>
<td>95.58</td>
<td>91.13</td>
<td>93.89</td>
</tr>
<tr>
<td></td>
<td>8,8,3</td>
<td>94.17</td>
<td>94.79</td>
<td>94.17</td>
</tr>
</tbody>
</table>

5.3 Experiment-3: SVM classifier

Parameters values set for SVM classifier, used for ECG beat classification, are as follows:

- **Processing**
  - Normalise features into range [-1,1]
  - Classify data into different classes using SVM

- **SVM Parameters**
  - Kernel function: rbf, sigmoid

We have performed number of experiments by varying values of C and Gamma parameters to see the effect of high and small regularisation of the classification model, and fat as well as thin kernels. We implemented SVM classifier with two kernel functions: Radial Basis Function (RBF) and Sigmoid. Table 4 shows experiment results obtained for SVM classifier. It is observed from the table that the classification accuracy achieved by SVM based on RBF kernel is better than those achieved by SVM based on sigmoid for all values of C and Gamma parameters.
Table 4  Experimental results obtained for SVM classifier

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Parameters</th>
<th>Dataset I</th>
<th>Dataset II</th>
<th>Dataset III</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c</td>
<td>gamma</td>
<td>Acc(%)</td>
<td>Acc(%)</td>
</tr>
<tr>
<td>Radial Basis</td>
<td>11179</td>
<td>0.0030</td>
<td>97.22</td>
<td>95.82</td>
</tr>
<tr>
<td></td>
<td>25844</td>
<td>0.0035</td>
<td>97.50</td>
<td>96.58</td>
</tr>
<tr>
<td></td>
<td>2880</td>
<td>0.0072</td>
<td>97.12</td>
<td>96.43</td>
</tr>
<tr>
<td></td>
<td>23269</td>
<td>0.0089</td>
<td>96.93</td>
<td>96.69</td>
</tr>
<tr>
<td></td>
<td>28992</td>
<td>0.0084</td>
<td>97.50</td>
<td>96.94</td>
</tr>
<tr>
<td></td>
<td>3326.2</td>
<td>0.0071</td>
<td>96.47</td>
<td>97.08</td>
</tr>
<tr>
<td></td>
<td>3658.7</td>
<td>0.0089</td>
<td>96.84</td>
<td>96.61</td>
</tr>
<tr>
<td></td>
<td>10886</td>
<td>0.0043</td>
<td>97.08</td>
<td>96.25</td>
</tr>
<tr>
<td></td>
<td>3576.5</td>
<td>0.0056</td>
<td>96.89</td>
<td>96.43</td>
</tr>
<tr>
<td></td>
<td>3655.9</td>
<td>0.0078</td>
<td>96.98</td>
<td>96.11</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>11179</td>
<td>0.0030</td>
<td>97.03</td>
<td>95.86</td>
</tr>
<tr>
<td></td>
<td>25844</td>
<td>0.0035</td>
<td>97.03</td>
<td>96.64</td>
</tr>
<tr>
<td></td>
<td>2880</td>
<td>0.0072</td>
<td>96.84</td>
<td>96.54</td>
</tr>
<tr>
<td></td>
<td>23269</td>
<td>0.0089</td>
<td>96.51</td>
<td>95.82</td>
</tr>
<tr>
<td></td>
<td>28992</td>
<td>0.0084</td>
<td>97.36</td>
<td>96.07</td>
</tr>
<tr>
<td></td>
<td>3326.2</td>
<td>0.0071</td>
<td>97.13</td>
<td>95.75</td>
</tr>
<tr>
<td></td>
<td>3658.7</td>
<td>0.0089</td>
<td>96.89</td>
<td>96.58</td>
</tr>
<tr>
<td></td>
<td>10886</td>
<td>0.0043</td>
<td>96.52</td>
<td>95.89</td>
</tr>
<tr>
<td></td>
<td>3576.5</td>
<td>0.0056</td>
<td>96.84</td>
<td>96.51</td>
</tr>
<tr>
<td></td>
<td>3655.9</td>
<td>0.0078</td>
<td>96.61</td>
<td>96.43</td>
</tr>
</tbody>
</table>

5.4  Comparison of result of PSO-FFNN, MLPNN, and SVM

PSO-FFNN, MLPNN and SVM are experimentally tested for ECG beat classification. Comparison of these classifiers based on obtained results are shown in Table 5. Figure 12 shows this comparison using a bar chart. As shown in Table 5 and Figure 12, PSO-FFNN outperforms MLPNN and SVM in terms of accuracy. The correct choice of kernel parameters (e.g. gamma g) and penalty parameter C is crucial for obtaining good results while using SVM. This suggests that an extensive search must be conducted on the parameter search space before experimentation or optimisation of these parameters needs to be done using other optimisation techniques. MLPNN trained with back propagation (BP) algorithm suffers from slow convergence to local and global minima. The random initialisation of weights may makes poor mapping of inputs to outputs. Back propagation algorithm gets stuck in local minima and has slow speed of convergence. These drawbacks of BP-MLPNN can be overcome by PSO as it optimises the weights and calculate fitness accordingly. These optimised weights and biases are then used to train the FFNN. Therefore, PSO-FFNN performs better than MLPNN and SVM.
Table 5  Comparison of PSO-FFNN, MLPNN, and SVM classifier

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset I Acc(%)</th>
<th>Dataset II Acc(%)</th>
<th>Dataset III Acc(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO-FFNN</td>
<td>99.41</td>
<td>98.68</td>
<td>98.69</td>
</tr>
<tr>
<td>MLPNN</td>
<td>97.86</td>
<td>97.55</td>
<td>97.26</td>
</tr>
<tr>
<td>SVM</td>
<td>97.50</td>
<td>97.08</td>
<td>97.77</td>
</tr>
</tbody>
</table>

Figure 12  Comparing performances of three classifiers on three different datasets

6 Conclusions

Three classifiers, PSO-FFNN, MLPNN with back propagation and SVM were designed and implemented for ECG arrhythmia classification. Three different datasets were formed which vary in terms of number of classes (arrhythmia types) and number of beats. The raw ECG data was denoised and features were extracted from denoised ECG signal using Pan-Tompkins algorithm. Pan-Tompkins algorithm extracted R peak sample number and QRS duration. From R peak sample number and time duration of R peak other features such as RR interval, RRR interval, RR mean were calculated. All these features were fed as inputs to three classifiers. The trained classifiers were applied on test data and obtained results were compared.

The PSO-FFNN classifier achieved accuracy of 99.41%, while MLPNN and SVM classifiers achieved accuracy of 97.86% and 97.50% for Dataset-1 (3 classes dataset). The PSO-FFNN classifier achieved accuracy of 98.68%, while MLPNN and SVM classifier achieved accuracy of 97.55%, and 97.08% for 4 classes dataset. The PSO-FFNN classifier achieved accuracy of 98.69%, while MLPNN and SVM classifier achieved accuracy of 97.26% and 97.77% for 6 classes dataset. Out of three models, PSO-FFNN gave better performance in terms of accuracy on all three datasets. SVM demands optimised values for kernel and penalty parameters to achieve good classification accuracy. These parameters can be optimised manually or by other optimisation techniques. Back propagation algorithm gets stuck in local minima and has slow speed of convergence. Due to these reasons, PSO-FFNN performs better than MLPNN and SVM.
In the current study, we considered two shape related and five time related beat features. However, one can extend the input feature vector by considering other shape related beat features (such as amplitude of Q-valley, position of Q-valley, amplitude of P-peak etc.) and test its impact on the classification accuracy. Moreover, we have not considered all arrhythmia types in our experiments. We tested three different datasets: Dataset-I constituting three classes, Dataset-II constituting four classes, and Dataset-III constituting six classes. However, our future work aims at classifying all ECG arrhythmia types. In our future research, we intend to focus on following points: (i) classification of arrhythmia beats by considering wavelet based features (DWT coefficients as input features), (ii) optimal selection of feature sub-set to reduce the training time (computational requirements), (iii) achieving high classification accuracy with small sized input feature vector and limited training dataset.

References


ECG beat classification using machine learning techniques


