

# Automated patient-specific classification of premature ventricular contractions

Turker Ince\*, Serkan Kiranyaz\*\*, Moncef Gabbouj\*\*  
\*Izmir University of Economics, Izmir, Turkey, turker.ince@ieu.edu.tr  
\*\*Tampere University of Technology, Tampere, Finland  
{serkan.kiranyaz, moncef.gabbouj}@tut.fi

## Abstract

*In this paper, we present an automated patient-specific electrocardiogram (ECG) beat classifier designed for accurate detection of premature ventricular contractions (PVCs). In the proposed feature extraction scheme, the principal component analysis (PCA) is applied to the dyadic wavelet transform (DWT) of the ECG signal to extract morphological ECG features, which are then combined with the temporal features to form a resultant efficient feature vector. For the classification scheme, we selected the feed-forward artificial neural networks (ANNs) optimally designed by the multi-dimensional particle swarm optimization (MD-PSO) technique, which evolves the structure and weights of the network specifically for each patient. Training data for the ANN classifier include both global (total of 150 representative beats randomly sampled from each class in selected training files) and local (the first 5 min of a patient's ECG recording) training patterns. Simulation results using 40 files in the MIT/BIH arrhythmia database achieved high average accuracy of 97% for differentiating normal, PVC, and other beats.*

## 1. Introduction

The morphological and temporal characteristics of ECG signals show significant variations not only from patient to patient or for different beat types but even for the same individual and same beat under different timing and situation [1]. This poses major problems for generalization performance of many ECG classification algorithms when presented with new patient's ECG data. Many automated ECG classifier systems employing different techniques for feature extraction from ECG signals and for classification of ECG patterns have been proposed by several researchers. Different solutions to ECG classification problem based on linear discriminant analysis [2], support vector machines [3], artificial neural networks (ANNs) [4], hidden Markov models [5], self-organizing maps [6], filter banks [7], and mixture-of-experts method [8] have been presented in the literature. The ECG signal features used with these classifiers can be mainly categorized into morphological features, timing interval features, statistical features, frequency domain features, and wavelet transform features. Many of these ECG classification algorithms achieved high accuracies (>90%), but they generally lack satisfactory sensitivity and

specificity performance due to significant variation among ECG waveforms and also their classification accuracy and efficiency degrades for a larger database of many patients.

ANNs have become one of the most popular pattern recognition techniques successfully applied to ECG signal classification problem. Hosseini, *et al.* used a multi-stage multilayer perceptron (MLP) neural network classifier that achieved 88.3% accuracy over 10 ECG records of the MIT/BIH arrhythmia database [9]. Recently, Inan, *et al.* combined downsampled wavelet transform of the ECG signal with timing information for the training of a single-hidden layer MLP classifier to achieve 95.16% accuracy in differentiating normal and PVC from other beats over 40 files in MIT/BIH arrhythmia database [10]. For classification of selected ECG features, both groups used MLP classifiers with fixed network structures selected empirically based on performance. In these studies, the MLP classifiers were trained with the backpropagation (BP) algorithm, which is just a gradient descent algorithm on the error space hence the BP may end up in a local minimum (suboptimal solution) making it entirely dependent on initial settings.

In this paper, we present an automated patient-specific classification system designed for robust and accurate detection of PVC beats. The feature extractor in the proposed classifier employs the translation-invariant dyadic wavelet transform and the principal component analysis to extract morphological ECG features, which are then combined with the temporal features to form a resultant efficient feature vector. As the pattern recognition and decision unit, we have utilized the feed-forward ANNs optimally designed by the MD-PSO technique within an architecture space.

## 2. Methodology

The MIT-BIH arrhythmia database [11] is used in this study. The database contains 48 records, each containing two-channel ECG signals for 30 min duration selected from 24-hr recordings of 47 different individuals. Continuous ECG signals are bandpass filtered at 0.1-100 Hz and then digitized at 360 Hz. The database contains annotation for both timing information and beat class information verified by independent experts. In this study, we chose total of 40 records (containing total of 99128 beats) from the database (4 records containing paced heartbeats and 4 other records were excluded in this experiment), using modified-lead II signals in all files and

utilizing the annotation information to locate beats in ECG signals. In the following sections, the suggested feature extraction and classification techniques used in the proposed classifier are presented.

## 2.1. Feature Extraction

The translation-invariant dyadic wavelet transforms (DWTs) pioneered by Mallat and Zhong [12] have been successfully applied to pattern recognition. The DWT has also been shown to be an efficient feature extraction method for ECG signals. Li, *et al.* demonstrated a QRS complex detection technique based on DWT of ECG signal using quadratic spline wavelet that achieved a 99.8% QRS detection rate [13]. The same wavelet decomposition is used by Shyu, *et al.* for feature extraction and classification over a relatively small dataset [14].

The dyadic wavelet transform of  $f(x)$  with a given mother wavelet  $\psi$  at the scale  $2^j$  is defined by the convolution product

$$W_{2^j} f(x) = f * \psi_{2^j}(x) = \int_{-\infty}^{\infty} f(t) \psi_{2^j}(x-t) dt, \quad (1)$$

where  $\psi_{2^j}(x) = (1/2^j) \psi(x/2^j)$  is a dilation of  $\psi$  by the scale factor  $2^{-j}$ . The fast DWT can be implemented using the filter bank structure, which is called the “algorithme a trous.” According to this algorithm, the signal is filtered by the same wavelet low-pass and high-pass filters, but instead of downsampling of the resulting approximation and detail coefficients, the mother wavelet is dilated (by inserting zeros) at each level of the transform.

We used a quadratic spline wavelet, also used by Li, *et al.*, which is bi-orthogonal wavelet with compact support and one vanishing moment. After analysis of DWT transformation of ECG signal at different scales (frequency bands), the DWT at the fourth scale is found to be most appropriate with respect to efficient description of morphological features from ECG waveform. By using the R-peak annotations, based on the range of R-R interval values in the database, we selected [-300ms, +400ms] window about the R peak and extracted DWT feature vector of each heartbeat. Assuming wavelet feature vectors of each heartbeat type as corresponding class samples, after normalization to zero-mean and unity standard deviation, we applied principal component analysis (PCA) to reduce redundancy and dimensionality of DWT feature data. From PCA analysis of DWT feature vectors, five principal components are selected (which account for about 95% of overall variation in the original feature data) to form a resultant compact morphological feature vector for each heartbeat signal. Besides the five principal components, the R-R time interval ( $RR_i = T_i - T_{i-1}$ ) and R-R time interval ratio ( $RR_i / RR_{i+1}$ ) are included as temporal features into the heartbeat feature vector which will be input to the classifier.

## 2.2. Classification

We utilized the feed-forward ANNs for classification of the extracted ECG feature vectors. Network structure and weights are optimized specifically for each patient using the multi-dimensional particle swarm optimization (MD-PSO) technique.

The training data for the ANN classifier include both global (common to each patient) and local (patient-specific) training patterns. The global part contains a total of 150 representative beats randomly sampled from each class in selected 13 training files (100,105-106,119,200-201,203,205,208,212-215) of the MIT-BIH arrhythmia database. The patient-specific training data includes the first 5 min of a patient’s ECG recording. The suggested classifier is then tested using the remaining 25 min of each patient’s ECG record for all 40 files from the database. This section presents the details of the MD PSO technique used in our study to design the (near)optimal patient-specific ANN classifiers.

### 2.2.1. MD PSO Algorithm

The particle swarm optimization (PSO) was introduced by Kennedy and Eberhart in [15], as a population based stochastic search and optimization process. It is originated from the computer simulation of the individuals (particles or living organisms) in a bird flock or fish school, which basically show a natural behavior when they search for some target (e.g. food). In the basic PSO algorithm, the particles are initially distributed randomly over the search space with a random velocity and the goal is to converge to the global optimum of a function or a system. Each particle keeps track of its position in the search space and its best solution so far achieved. This is the personal best value (the so-called *pbest*) and the PSO process also keeps track of the global best solution so far achieved by the swarm with its particle index (the so called *gbest*). So during their journey with discrete time iterations, the velocity of each agent in the next iteration is computed by the best position of the swarm (position of the particle *gbest* as the *social* component), the best personal position of the particle (*pbest* as the *cognitive* component), and its current velocity (the *memory* term). Both *social* and *cognitive* components contribute randomly to the position of the agent in the next iteration. The major drawback of the PSO is that it can only be applied to a search space with fixed dimensions. However, in many of the optimization problems, the optimum dimension is also unknown (e.g. clustering, spatial segmentation, optimization of the dimensional functions, etc.) and should thus be determined within the PSO process. Instead of operating at a fixed dimension  $N$ , the MD PSO algorithm is designed to seek both positional and dimensional optima within a dimension range, ( $D_{\min} \leq N \leq D_{\max}$ ). In order to accomplish this, each particle has two sets of components, each of which has been subjected to two independent and consecutive processes. The first one is a regular positional PSO, i.e. the traditional velocity updates and due positional shifts in  $N$  dimensional search (solution) space. The second one is a dimensional PSO, which allows the particle to navigate through dimensions. Accordingly, each particle keeps track of its last position, velocity and personal best position (*pbest*) in a particular dimension so that when it revisits that the same dimension at a later time, it can perform its regular “positional” fly using this information. The dimensional PSO process of each particle may then move the particle to another dimension where it will remember its positional status and keep “flying” within the positional PSO process in this dimension, and so on. The swarm, on the other hand, keeps track of the *gbest* particles in all dimensions, each of which

respectively indicates the best (global) position so far achieved and can thus be used in the regular velocity update equation for that dimension. Similarly the dimensional PSO process of each particle uses its personal best dimension in which the personal best fitness score has so far been achieved. Finally, the swarm keeps track of the global best dimension,  $dbest$ , among all the personal best dimensions. The  $gbest$  particle in  $dbest$  dimension represents the optimum solution and dimension, respectively. Due to space limitation, we skipped the details of MD PSO algorithm.

### 2.2.2. Evolving ANNs by the MD PSO

As a stochastic search process in multi-dimensional search space, MD PSO seeks (near-) optimal networks in an architecture space, which can be defined by any type of ANNs with any properties. All network configurations in the architecture space are enumerated into a (dimensional) hash table with a proper hash function, which basically ranks the networks with respect to their complexity, i.e. associates higher hash indices to networks with higher complexity. MD PSO can then use each index as a unique dimension of the search space where particles can make inter-dimensional navigations to seek for an optimum dimension ( $dbest$ ) and the optimum solution on that dimension.

In this work, we apply MD PSO technique for evolving fully-connected, feed-forward ANNs or the so-called MLPs. The reasoning behind this choice is that MLP is the most widely used in this field and it offers conventional methods such as BP for training. Suppose for the sake of simplicity, a range is defined for the minimum and maximum number of layers,  $\{L_{\min}, L_{\max}\}$  and number of neurons for hidden layer  $l$ ,  $\{N_{\min}^l, N_{\max}^l\}$ . Without loss of generality, assume that the size of both input and output layers is determined by the problem and hence fixed. This yields that the architecture space can now be defined only by two range arrays,  $R_{\min} = \{N_I, N_{\min}^1, \dots, N_{\min}^{L_{\max}-1}, N_O\}$  for minimum and  $R_{\max} = \{N_I, N_{\max}^1, \dots, N_{\max}^{L_{\max}-1}, N_O\}$ , for maximum number of neurons allowed for each layer of a MLP. The size of both arrays is naturally  $L_{\max} + 1$  where corresponding entries define the range of the  $l^{\text{th}}$  hidden layer for all those MLPs, which can have an  $l^{\text{th}}$  hidden layer. The size of input and output layers,  $\{N_I, N_O\}$ , is fixed and same for all configurations in the architecture space within which any  $l$ -layer MLP can be defined providing that  $L_{\min} \leq l \leq L_{\max}$ .  $L_{\min} \geq 1$  and  $L_{\max}$  can be set to any value meaningful for the problem encountered. The hash function then enumerates all potential MLP configurations into hash indices, starting from the simplest MLP with  $L_{\min} - 1$  hidden layers, each of which has minimum number of neurons given in  $R_{\min}$ , to the most complex network with  $L_{\max} - 1$  hidden layers, each of which has maximum number of neurons given in  $R_{\max}$ . Let  $N_h^l$  be the number of hidden neurons in layer  $l$  of a MLP with input and output layer sizes  $N_I$  and  $N_O$ , respectively. The input neurons are merely fan-out units since

no processing take place. Let  $F$  be the activation function applied over the weighted inputs plus a bias, as follows:

$$y_k^{p,l} = F(s_k^{p,l}) \text{ where } s_k^{p,l} = \sum_j w_{jk}^{l-1} y_j^{p,l-1} + \theta_k^l \quad (2)$$

where  $y_k^{p,l}$  is the output of the  $k^{\text{th}}$  neuron of the  $l^{\text{th}}$  hidden/output layer when the pattern  $p$  is fed,  $w_{jk}^{l-1}$  is the weight from  $j^{\text{th}}$  neuron in layer  $l-1$  to  $k^{\text{th}}$  neuron in layer  $l$ , and  $\theta_k^l$  is the bias value of the  $k^{\text{th}}$  neuron of the  $l^{\text{th}}$  hidden/output layer, respectively. The training mean square error,  $MSE$ , is formulated as,

$$MSE = \frac{1}{2PN_O} \sum_{p \in T} \sum_{k=1}^{N_O} (t_k^p - y_k^{p,O})^2 \quad (3)$$

where  $t_k^p$  is the target (desired) output and  $y_k^{p,O}$  is the actual output from the  $k^{\text{th}}$  neuron in the output layer,  $l=O$ , for pattern  $p$  in the training set  $T$  with size  $P$ , respectively. At a time  $t$ , suppose that the particle  $a$  in the swarm,  $\xi = \{x_1, \dots, x_a, \dots, x_S\}$ , has the positional component formed as,

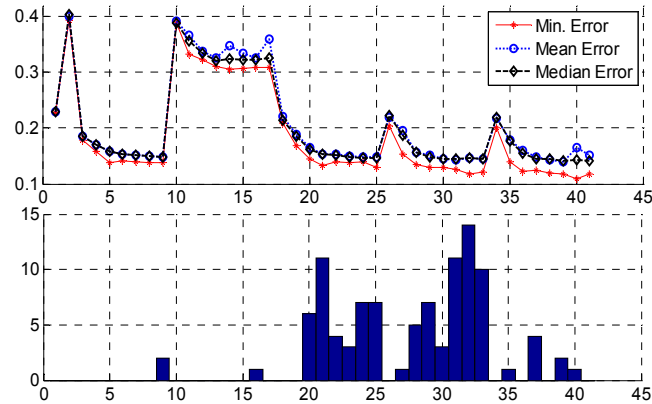
$$xx_a^{xd_a(t)}(t) = \left\{ \begin{array}{l} \{w_{jk}^0\}, \{w_{jk}^1\}, \{\theta_k^1\}, \{w_{jk}^2\}, \{\theta_k^2\} \\ \dots, \{w_{jk}^{O-1}\}, \{\theta_k^{O-1}\}, \{\theta_k^O\} \end{array} \right\} \quad (4)$$

where  $\{w_{jk}^l\}$  and  $\{\theta_k^l\}$  represent the sets of weights and biases of the layer  $l$ . Note that the input layer ( $l=0$ ) contains only weights whereas the output layer ( $l=O$ ) has only biases. By means of such a direct encoding scheme, the particle  $a$  represents all potential network parameters of the MLP architecture at the dimension (hash index)  $xd_a(t)$ . As mentioned earlier, the dimension range,  $D_{\min} \leq xd_a(t) \leq D_{\max}$ , where MD PSO particles can make inter-dimensional jumps, is determined by the architecture space defined. Setting  $MSE$  in (3) as the fitness function enables MD PSO to perform evolutions of both network parameters and architectures within its native process. Due to space limitation, we skipped further details of evolutionary ANNs by the MD PSO algorithm.

## 3. Experimental Results

In order to determine which network architectures are optimal for a given problem, we apply exhaustive BP training over every network configuration in the architecture space defined. As mentioned earlier, BP is nothing but a gradient descent algorithm and thus for a single run, it is susceptible to get trapped to the nearest local minimum. However, performing it several times with randomized initial parameters eventually increases the chance of converging to (a close vicinity of) the global minimum in the error space, and therefore, it is performed by significantly large amount of times (e.g.  $K=500$ ) so that for a particular architecture, the minimum error obtained among all trials, can then be assumed to be the (near) optimal score achieved with that network. Note that even though  $K$  is kept quite

high, there is still no guarantee of converging to the global optimum with BP; however, the idea is to obtain the “trend” of best performances achievable with every configuration under equal training conditions. In this way the optimality of the networks evolved by MD PSO can be justified. In order to show the optimality of the network configurations evolved by MD PSO, we first of all, use a “limited” architecture space ( $R^1: R^1_{\min} = \{N_I, 1, 1, N_O\}$  and  $R^1_{\max} = \{N_I, 8, 4, N_O\}$ ) containing the simplest 1, 2 or 3 layers MLPs with  $L^1_{\min} = 1$  and  $L^1_{\max} = 3$ . It contains 41 networks.



**Figure 1: Error (MSE) statistics from exhaustive BP training (top) and *dbest* histogram from 100 MD PSO evolutions (bottom) for patient record 214.**

In this experiment, for patient record 214, we perform 100 MD PSO runs with 250 particles, each of which terminates at the end of 400 epochs (iterations). Figure 1 shows *dbest* histogram and the error statistics plot from the exhaustive BP training the data of patient record 214 where  $N_I = 7$ ,  $N_O = 3$  used for all networks. In this problem, BP exhibits a better performance on the majority of the configurations, and the 100 MD PSO runs show that there are in fact two optimum dimensions: 21 and 34 (corresponding to 3-layer MLPs in  $7 \times 5 \times 2 \times 3$  and  $7 \times 2 \times 4 \times 3$  forms), which can achieve minimum mean square errors (MSEs) over the training data, i.e.  $mMSE(21) < 0.8 \cdot 10^{-1}$  and  $mMSE(34) < 10^{-1}$ . The majority of MD PSO runs, which is represented in the *dbest* histogram on Figure 1, evolved to (near-) optimum networks except the one run for *dbest*=16. This is the minority case where MD PSO trapped to local minima.

The performance of the patient-specific classifiers were verified by testing over 40 records (patients) containing total of 99128 beats from the MIT-BIH arrhythmia database. In this experiment, the following types of beats are considered: normal, PVC, LBBB, RBBB, aberrated atrial premature beat, atrial premature contraction, and supraventricular premature beat [11]. Table I summarizes ECG classification performance of the proposed technique and compares results with the other recent major technique by Inan *et al.* using the same standard metrics: accuracy (A), sensitivity (Se), and positive predictivity (Pp). Each experiment was performed 5 times (i.e. 5 MD PSO runs

per patient using the same setup for the patient record 214). Overall, the proposed classifier achieved high average detection accuracy of 97.0% with average standard deviation of 0.5% during the tests, and average accuracy of 98.6% with average standard deviation of 0.3% during the training phase.

**TABLE I**  
Performance Comparison of PVC Detection (in percent)

Method	A	Normal		PVC		Other	
		Se	Pp	Se	Pp	Se	Pp
Inan <i>et al.</i>	95.2	98.1	97.0	85.2	92.4	87.4	94.5
<b>Proposed</b>	<b>97.0</b>	<b>99.4</b>	<b>98.9</b>	<b>93.4</b>	<b>93.3</b>	<b>87.5</b>	<b>97.8</b>

## References

- [1] R. Hoekema, G. J. H. Uijen, and A. v. Oosterom, “Geometrical aspects of the interindividual variability of multilead ECG recordings,” *IEEE Trans. Biomed. Eng.*, vol. 48, no. 5, pp. 551–559, May 2001.
- [2] P. de Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ECG morphology and heartbeat interval features,” *IEEE Trans. Biomed. Eng.*, vol. 51, no. 7, pp. 1196–1206, Jul. 2004.
- [3] S. Osowski, L. T. Hoai, and T. Markiewicz, “Support vector machinebased expert system for reliable heartbeat recognition,” *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 582–589, Apr. 2004.
- [4] Y. Hu, S. Palreddy, and W. J. Tompkins, “A patient-adaptable ECG beat classifier using a mixture of experts approach,” *IEEE Trans. Biomed. Eng.*, vol. 44, no. 9, pp. 891–900, Sep. 1997.
- [5] R. V. Andreado, B. Dorizzi, and J. Boudy, “ECG signal analysis through hidden Markov models,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 8, pp. 1541–1549, Aug. 2006.
- [6] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sörnmo, “Clustering ECG complexes using Hermite functions and self-organizing maps,” *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 838–848, Jul. 2000.
- [7] X. Alfonso and T. Q. Nguyen, “ECG beat detection using filter banks,” *IEEE Trans. Biomed. Eng.*, vol. 46, no. 2, pp. 192–202, Feb. 1999.
- [8] Y. Hu, S. Palreddy, and W. J. Tompkins, “A patient-adaptable ECG beat classifier using a mixture of experts approach,” *IEEE Trans. Biomed. Eng.*, vol. 44, no. 9, pp. 891–900, Sep. 1997.
- [9] H. G. Hosseini, K. J. Reynolds, and D. Powers, “A multi-stage neural network classifier for ECG events,” in *Proc. 23rd Int. Conf. IEEE EMBS*, 2001, vol. 2, pp. 1672–1675.
- [10] Inan, *et al.*, “Robust neural-network based classification of PVCs using wavelet transform and timing interval features,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2507–2515, Dec. 2006.
- [11] R. Mark and G. Moody, “MIT-BIH Arrhythmia Database Directory,” [Online]. Available: <http://ecg.mit.edu/dbinfo.html>
- [12] S. G. Mallat and S. Zhong, “Characterization of signals from multiscale edges,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 710–732, July 1992.
- [13] C. Li, C. X. Zheng, and C. F. Tai, “Detection of ECG characteristic points using wavelet transforms,” *IEEE Trans. Biomed. Eng.*, vol. 42, no. 1, pp. 21–28, Jan. 1995.
- [14] L. Y. Shyu, Y. H. Wu, and W. C. Hu, “Using wavelet transform and fuzzy neural network for VPC detection from the holter ECG,” *IEEE Trans. Biomed. Eng.*, vol. 51, no. 7, pp. 1269–1273, Jul. 2004.
- [15] J. Kennedy, R. Eberhart, “Particle swarm optimization,” in *Proc. of IEEE Int. Conf. On Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.