A Generic and Robust System for Automated Patient-Specific Classification of Electrocardiogram Signals

Turker Ince, Serkan Kiranyaz and Moncef Gabbouj

***Abstract*-- This paper presents a generic and patient-specific classification system designed for robust and accurate detection of electrocardiogram (ECG) heartbeat patterns. The proposed feature extraction process utilizes morphological wavelet transform features, which are projected onto a lower-dimensional feature space using principal component analysis, and temporal features from the ECG data. For the pattern recognition unit, feed-forward and fully-connected artificial neural networks (ANNs), which are optimally designed for each patient by the proposed multi-dimensional particle swarm optimization (MD PSO) technique, are employed. By using relatively small common and patient-specific training data, the proposed classification system can adapt to significant inter-patient variations in ECG patterns by training the optimal network structure and thus achieves higher accuracy over larger data sets. The classification experiments over a benchmark database demonstrate that the proposed system achieves such average accuracies and sensitivities better than most of the current state-of-the-art algorithms for detection of ventricular ectopic beats (VEB) and supraventricular ectopic beats (SVEB). Over the entire database, the average accuracy - sensitivity performances of the proposed system for VEB and SVEB detections are 98.3% - 84.6%, and 97.4% - 63.5%, respectively. Finally, due to its parameter invariant nature, the proposed system is highly generic and thus applicable to any ECG dataset.**

***Index Terms*—Biomedical signal classification; Particle swarm optimization; Multi-dimensional search; Evolutionary Neural networks**

I. Introduction

EACH individual heartbeat in the cardiac cycle of the recorded electrocardiogram (ECG) waveform

…

shows the time evolution of the heart's electrical activity, which is made of distinct electrical depolarization-repolarization patterns of the heart. Any disorder of heart rate or rhythm, or change in the morphological pattern is an indication of an arrhythmia, which could be detected by analysis of the recorded ECG waveform. Real-time automated ECG analysis in clinical settings is of great assistance to clinicians in detecting cardiac arrhythmias, which often arise as a consequence of a cardiac disease and may be life-threatening and require immediate therapy. However, automated classification of ECG beats is a challenging problem as the morphological and temporal characteristics of ECG signals show significant variations for different patients and under different temporal and physical conditions [1]. Many algorithms for automatic detection and classification of ECG heartbeat patterns have been presented in the literature including signal processing techniques such as frequency analysis [2], wavelet transform [3], [4], and filter banks [5], statistical [6] and heuristic approaches [7], hidden Markov models [8], support vector machines [9], artificial neural networks (ANNs) [10], and mixture-of-experts method [11]. In general, ECG classifier systems based on past approaches have not performed well in practice because of their important common drawback of having an inconsistent performance when classifying a new patient's ECG waveform. This makes them unreliable to be widely used clinically, and causes severe degradation in their accuracy and efficiency for larger databases, [12], [13]. Moreover, the *Association for the Advancement of Medical Instrumentation* (AAMI) provides standards and recommended practices for reporting performance results of automated arrhythmia detection algorithms [14]. However, despite quite many ECG classification methods proposed in the literature, only few [11], [15], [16] have in fact used the AAMI standards as well as the complete data from the benchmark MIT-BIH arrhythmia database.

The performance of ECG pattern classification strongly depends on the characterization power of the features extracted from the ECG data and on the design of the classifier (classification model or network structure and parameters). Due to its time-frequency localization properties, the wavelet transform is an efficient tool for analyzing non-stationary ECG signals [17]. The wavelet transform can be used to decompose an ECG signal according to scale, allowing separation of the relevant ECG waveform morphology descriptors from the noise, interference, baseline drift, and amplitude variation of the original signal. Several researchers have previously used the wavelet transform coefficients at the appropriate scales as morphological feature vectors rather than the original signal time series and achieved good classification performance [3], [4]. Accordingly in the current work the proposed feature extraction technique employs the translation-invariant dyadic wavelet transform (TI-DWT) in order to extract effectively the morphological information from ECG data. Furthermore, the dimension of the input morphological feature vector is reduced by projecting it onto a lower-dimensional feature space using principal component analysis (PCA) in order to significantly reduce redundancies in such a high dimensional data space. The lower-dimensional morphological feature vector is then combined with two critical temporal features related to inter-beat time interval to improve accuracy and robustness of classification as suggested by the results of previous studies [15].

Artificial neural networks (ANNs) are powerful tools for pattern recognition as they have the capability to learn complex, nonlinear surfaces among different classes, and such ability can therefore be the key for ECG beat recognition and classification [18]. Although many promising ANN-based techniques have been applied to ECG signal classification, [18]-[21], the global classifiers based on a static (fixed) ANN have not performed well in practice. On the other hand, algorithms based on patient-adaptive architecture have demonstrated significant performance improvement over conventional global classifiers [11], [13], [16]. Among all, one particular approach, a personalized ECG heartbeat pattern classifier based on evolvable block-based neural networks (BbNN) using *Hermite* transform coefficients [16], achieved such a performance that is significantly higher than the others. Although this recent work clearly demonstrates the advantage of using evolutionary ANNs, which can be automatically designed according to the problem (patient’s ECG data), serious drawbacks and limitations can also be observed. For instance, there are around 10-15 parameters/thresholds that needed to be set empirically with respect to the dataset used and this obviously brings about the issue of robustness when it is used for a different database. Another drawback can occur due to the specific ANN structure proposed, i.e. the BbNN, which requires equal sizes for input and output layers. Even more critical is the Back Propagation (BP) method, used for training, and Genetic Algorithm (GA), for evolving the network structure, both have certain deficiencies [22]. In particular, the BP most likely gets trapped into a local minimum, making it entirely dependent on the initial (weight) settings.

In order to address such deficiencies and drawbacks, in this paper we propose a multi-dimensional particle swarm optimization (MD PSO) technique, which automatically designs the optimal ANNs (both network structure and connection weights with respect to the training mean square error) specifically for each patient and according to the patient's ECG data. On the contrary to the specific BbNN structure used in [16] with the aforementioned problems, MD PSO is used to evolve traditional ANNs and so the focus is particularly drawn on automatic design of the MLPs. This evolutionary operator makes the proposed system *generic*, that is no assumption is made about the number of (hidden) layers and in fact none of the network properties (e.g. feed-forward or not, differentiable activation function or not, etc.) is an inherent constraint. As long as the potential network configurations are transformed into a hash (dimension) table with a proper hash function where indices represent the solution space dimensions of the particles, MD PSO can then seek both positional and dimensional optima in an interleaved PSO process. The optimum dimension found corresponds to a distinct ANN architecture where the network parameters (connections, weights and biases) can be resolved from the positional optimum reached on that dimension. Our earlier work on detection of premature ventricular contractions demonstrated performance improvement of the proposed approach over conventional techniques [23]. Moreover, we aim to achieve a high level of *robustness* with respect to the variations of the dataset, since the proposed system is designed with a minimum set of parameters, and in such a way that their significant variations should not show a major impact on the overall performance. Above all, using standard ANNs such as traditional MLPs, instead of specific architectures (e.g. BbNN in [16]) further contributes to the generic nature of the proposed system and in short, all these objectives are meant to make it applicable to any ECG dataset without any modifications (such as tuning the parameters or changing the feature vectors, ANN types, etc.).

The rest of the paper is organized as follows. Section II outlines the ECG dataset used in this study and provides a detailed description of the feature extraction methodology for the proposed patient-specific heartbeat classification system. MD PSO and its application over the automatic ANN design are presented in Section III. In Section IV, the optimality, performance and robustness of the proposed classifier are evaluated over the MIT/BIH arrhythmia database using standard performance metrics and the results are compared with previously published work. Finally, Section V concludes the paper.

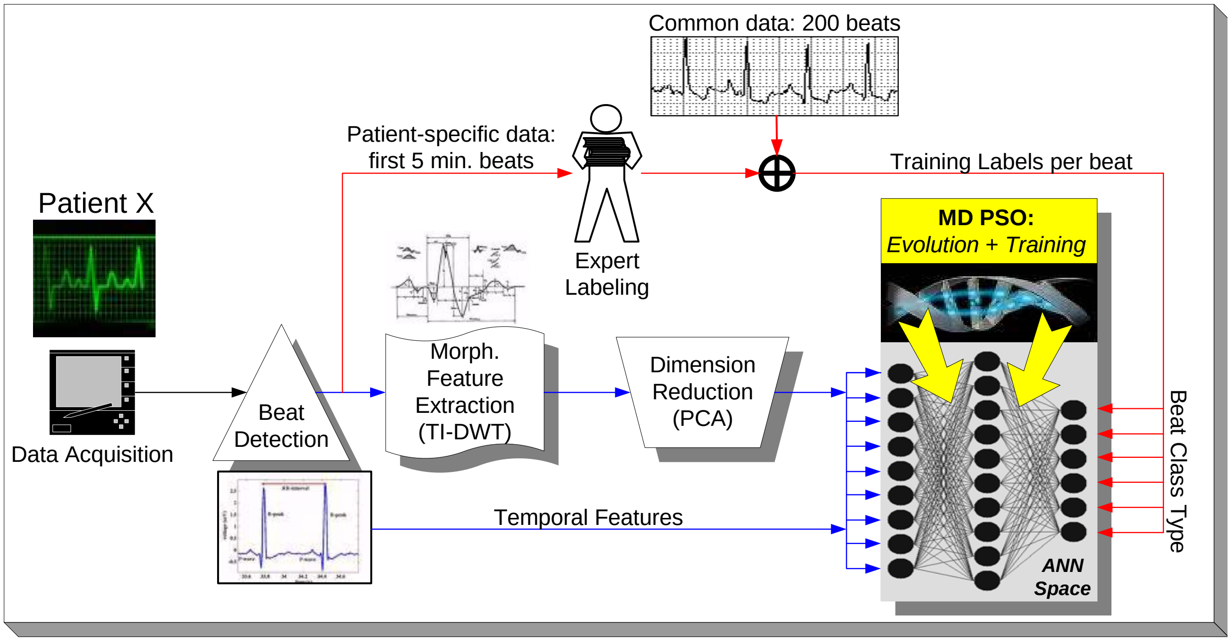


Figure 1: Patient-specific training process of the proposed ECG classification system.

II. ECG Data Processing

A. ECG Data

In this study, the MIT/BIH arrhythmia database [24] is used for training and performance evaluation of the proposed patient-specific ECG classifier. The database contains 48 records, each containing two-channel ECG signals for 30-min duration selected from 24-hour recordings of 47 individuals. Continuous ECG signals are band-pass 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 the current work, so as to comply with the AAMI ECAR-1987 recommended practice [14], we used 44 records from the MIT/BIH arrhythmia database, excluding 4 records which contain paced heartbeats. The first 20 records (numbered in the range of 100 to 124), which include representative samples of routine clinical recordings, are used to select representative beats to be included in the common training data. The remaining 24 records (numbered in the range of 200 to 234,) contain ventricular, junctional, and supraventricular arrhythmias. A total of 83648 beats from all 44 records are used as test patterns for performance evaluation. AAMI recommends that each ECG beat be classified into the following five heartbeat types: N (beats originating in the sinus mode), S (supraventricular ectopic beats), V (ventricular ectopic beats), F (fusion beats), and Q (unclassifiable beats). For all records, we used the modified-lead II signals and utilized the labels to locate beats in ECG data. The beat detection process is beyond the scope of this paper, as many highly accurate ($> 99\%$) beat detection algorithms have been reported in literature [25], [17].

B. Feature Extraction Methodology

As suggested by the results from previous works [10], [11], [15], both morphological and temporal features are extracted and combined into a single feature vector for each heartbeat to improve accuracy and robustness of the proposed classifier. The wavelet transform is used to extract morphological information from the ECG data. The time-domain ECG signatures were first normalized by subtracting the mean voltage before transforming into time-scale domain using the dyadic wavelet transform (DWT). According to wavelet transform theory, the multiresolution representation of the ECG signal is achieved by convolving the signal with scaled and translated versions of a mother wavelet. For practical applications, such as processing of sampled and quantized raw ECG signals, the discrete wavelet transform can be computed by scaling the wavelet at the dyadic sequence $\left(2^j\right)_{j\in Z}$ and translating it on a dyadic grid whose interval is proportional to 2^{-j}. The discrete WT is not only complete but also non-redundant unlike the continuous WT. Moreover, the wavelet transform of a discrete signal can be efficiently calculated using the decomposition by a two-channel multirate filter bank (the pyramid decomposition). However, due to the rate-change operators in the filter bank, the discrete WT is not time-invariant but actually very sensitive to the alignment of the signal in time [26].

To address the time-varying problem of wavelet transforms, Mallat proposed a new algorithm for wavelet representation of a signal, which is invariant to time shifts [27]. According to this algorithm, which is called a translation-invariant dyadic wavelet transform (TI-DWT), only the scale parameter is sampled along the dyadic sequence $\left(2^j\right)_{j\in Z}$ and the wavelet transform is calculated for each point in time. TI-DWTs pioneered by Mallat have been successfully applied to pattern recognition [27]. The fast TI-DWT algorithm, whose computational complexity is $O(N\log N)$, can be implemented using a recursive filter tree architecture [27]. In this study, we selected a quadratic spline wavelet with compact support and one vanishing moment, as defined in [27]. The same wavelet function has already been successfully applied to

QRS detection in [17], achieving a 99.8% QRS detection rate for the MIT/BIH arrhythmia database. In the proposed ECG classification system, using a wavelet-based beat detector such as in [17] allows the same wavelet transform block to operate directly on the raw input ECG signal for beat detection and then morphological feature extraction, thus making the system more efficient and robust.

Figure 2 shows sample beat waveforms, including Normal (N), PVC (V), and APC (S) AAMI heartbeat classes, selected from record 201, modified-lead II from the MIT/BIH arrhythmia database and their corresponding translation-invariant DWT decompositions computed for the first five scales. While wavelet-based morphological features provide effective discrimination capability between normal and some abnormal heartbeats (i.e. PVC beats), two temporal features (i.e. the R-R time interval and R-R time interval ratio) contribute to the discriminating power of wavelet-based features, especially in discriminating morphologically similar heartbeat patterns (i.e. Normal and APC beats).

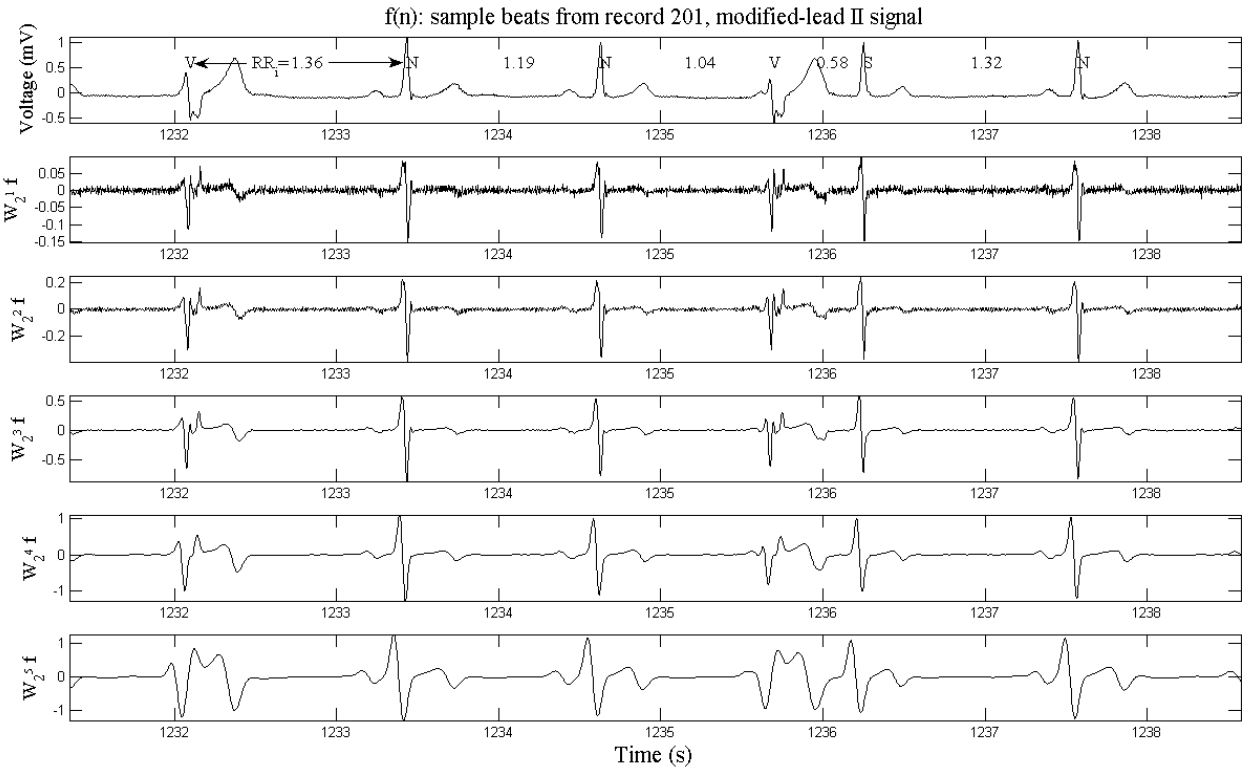
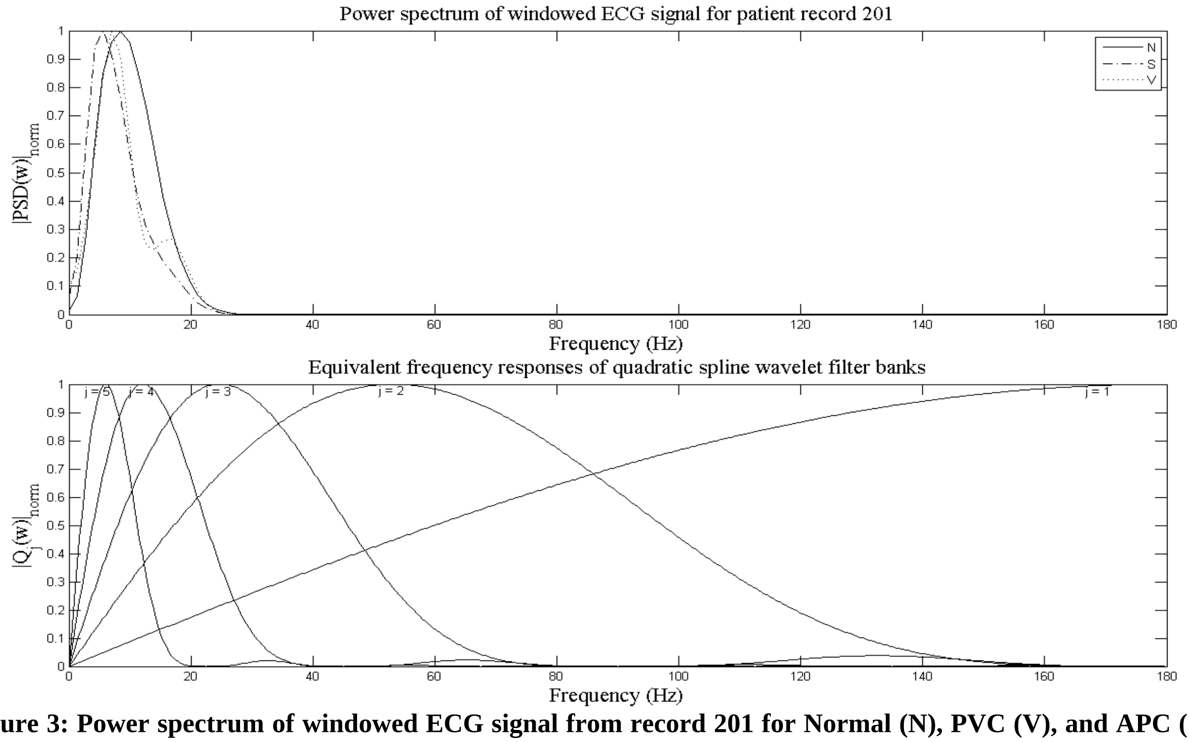


Figure 2: Sample beat waveforms, including Normal (N), PVC (V), and APC (S) AAMI heartbeat classes, selected from record 201 modified-lead II from the MIT/BIH arrhythmia database and corresponding TI DWT decompositions for the first five scales.

In Figure 3 (top), the estimated power spectrum of windowed ECG signal (a 500 ms long *Hanning* window is applied before FFT to suppress high-frequency components due to discontinuities in the end-points) from record 201 for N, V, and S beats is plotted, while equivalent frequency responses of FIR filters, $Q_j(w)$, for the first five scales at the native 360 Hz sampling frequency of the MIT/BIH data are illustrated at the bottom part of the figure. After analyzing the DWT decompositions of different ECG waveforms in the database, and according to the power spectra of ECG signal (the QRS complex, the P- and T-waves), noise, and artifact in [28], we selected $W_{2^4}f$ (at scale 2^4) signal as morphological features of each heartbeat waveform. Based on the –3 dB bandwidth of the equivalent $Q_4(w)$ filter (3.9 – 22.5 Hz) in Figure 3 (bottom), $W_{2^4}f$ signal is expected to contain most of QRS complex energy and the least amount of high-frequency noise and low-frequency baseline wander. The fourth scale decomposition together with RR-interval timing information was previously shown to be the best performing feature set for DWT-based PVC beat classification in [4]. Therefore, a 180-sample morphological feature vector is extracted per heartbeat from DWT of ECG signal at scale 2^4 by selecting a 500 ms window centered at the R-peak (found by using the beat annotation file). Each feature vector is then normalized to have a zero-mean and a unit variance to eliminate the effect of dc offset and amplitude biases.


Power spectrum of windowed ECG signal for patient record 201

Equivalent frequency responses of quadratic spline wavelet filter banks

Figure 3: Power spectrum of windowed ECG signal from record 201 for Normal (N), PVC (V), and APC (S) AAMI heartbeat classes, and equivalent frequency responses of FIR digital filters for a quadratic spline wavelet at 360 Hz sampling rate.

C. Preprocessing by Principal Component Analysis

The wavelet-based morphological features in the training set are post-processed using principal component analysis (PCA) to reduce dimensionality (and redundancy) of input feature vectors. PCA, also known as the Karhunen-Loéve transform (KLT), is a well-known statistical method that has been used for data analysis, data compression, redundancy and dimensionality reduction, and feature extraction. PCA is the optimal linear transformation, which finds a projection of the input pattern vectors onto a lower-dimensional feature space that retains the maximum amount of energy among all possible linear transformations of the pattern space. To describe the basic procedure of PCA, let *F* be a feature matrix of size $K\text{x}N$, whose rows are wavelet features of size $1\text{x}N$ each belonging to one of K heartbeats in the training data. First, the covariance matrix C_F of this feature matrix is computed as,

$$C_F = E\left\{(F - m)(F - m)^t\right\}, \tag{1}$$

where *m* is the mean pattern vector. From the eigen-decomposition of C_F, which is a $K\text{x}K$ symmetric and positive-definite matrix, the principal components taken as the eigenvectors corresponding to the largest eigenvalues are selected, and the morphological feature vectors are then projected onto these principal components (KL basis functions). In this work, 9 principal components which contain about 95% of overall energy in the original feature matrix, are selected to form a resultant compact morphological feature vector for each heartbeat signal. In this case, the PCA reduced the dimensionality of morphological features by a factor of 20. Figure 4 shows a scatter plot of Normal, PVC, and APC beats from record 201 in terms of the first and third principal components and inter-beat time interval. It is worth noting that dimensionality reduction of the input information improves efficiency of the learning for a NN classifier due to a smaller number of input nodes [29].

The data used for training the individual patient classifier consists of two parts: global (common to each patient) and local (patient-specific) training patterns. While patient-specific data contains the first 5 minutes segment of each patient’s ECG record and is used as part of the training data to perform patient adaptation, the global data set contains a relatively small number of representative beats from each class in the training files and helps the classifier learn other arrhythmia patterns that are not included in the patient-specific data. This practice conforms to the AAMI recommended procedure allowing the usage of at most 5 minutes section from the beginning of each patient’s recording for training [14].

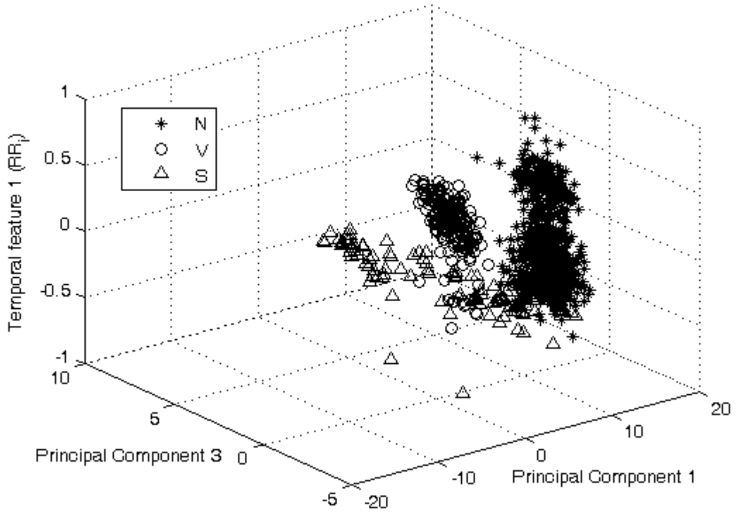


Figure 4: Scatter plot of Normal (N), PVC (V), and APC (S) beats from record 201 in terms of the first and third principal components and RR$_i$ time interval.

III. MD PSO Technique for Automatic ANN Design

As mentioned earlier, evolutionary ANNs are used for the classification of ECG data from each individual patient in the database. In this section, the MD PSO technique, which is developed for evolving ANNs, will be introduced first and we shall present its application for evolving the feed-forward ANNs next.

A. MD PSO Algorithm

The behavior of a single organism in a swarm is often insignificant but their collective and social behavior is of paramount importance. The particle swarm optimization (PSO) was introduced by Kennedy and Eberhart in 1995 as a population based stochastic search and optimization process [30]. It is originated from the computer simulation of the individuals (particles or living organisms) in a bird flock or fish school, which basically shows 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 (according to a given fitness score) of a function or a system. Each particle keeps track of its position in the search space and its best solution (i.e. lowest fitness score) 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 particle 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 particle in the next iteration. In principle, PSO follows the same path of the other evolutionary algorithms (EAs) such as Genetic Algorithm (GA), Genetic Programming (GP), Evolutionary Strategies (ES) and Evolutionary Programming (EP). The common point of all is that EAs are in population based nature and thus they can avoid being trapped in a local optimum. Thus they can find the optimum solutions; however, this is never guaranteed.

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 a range of dimensions. Accordingly, each particle keeps track of its last position, velocity and personal best position (*pbest*) in a particular dimension so that when it re-visits that 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. The details and the pseudo-code of the MD PSO algorithm are presented in the Appendix.

B. *MD PSO for Evolving ANNs*

As a stochastic search process in a multi-dimensional search space, MD PSO seeks for (near-)optimal (with respect to the training error) 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 an optimum dimension (*dbest*) and the optimum solution on that dimension, $x\hat{y}^{dbest}$. As mentioned earlier, the former corresponds to the optimal architecture and the latter encapsulates the (optimum) network parameters (connections, weights and biases).

In this section, we apply MD PSO technique for evolving fully-connected, feed-forward ANNs or the so-called MLPs. As mentioned earlier, the reasoning behind this choice is that MLP is the most widely used in this field and so our aim is to show that a superior performance as in [16] can also be achieved without using such specific ANN structures. MD PSO can evolve any MLP type and thus the architecture space can be defined over a wide range of configurations, i.e. say from a single-layer perceptron (SLP) to complex MLPs with many hidden layers. 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*, $\{N^l_{\min}, N^l_{\max}\}$.Without loss of generality, assume that the size of both input and output layers is determined by the problem and hence fixed. As a result, the architecture space can now be defined only by two range arrays, $R_{\min} = \{N_I, N^1_{\min}, ..., N^{L_{\max}-1}_{\min}, N_O\}$ and $R_{\max} = \{N_I, N^1_{\max}, ..., N^{L_{\max}-1}_{\max}, N_O\}$, one for the minimum and the other for the 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*th hidden layer for all those MLPs, which can have an *l*th hidden layer. The size of input and output layers, $\{N_I, N_O\}$, is fixed and remains the same for all configurations in the architecture space within which any *l*-layer MLP can be defined provided that $L_{\min} \le l \le L_{\max}$. $L_{\min} \ge 1$ and $L_{\max}$ can be set to any meaningful value for the problem at hand. 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 a minimum number of neurons given by $R_{\min}$, to the most complex network with $L_{\max}-1$ hidden layers, each of which has a maximum number of neurons given by $R_{\max}$.

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 takes place. Let *F* be the activation function applied over the weighted inputs plus a bias, as follows:

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

where $y_k^{p,l}$ is the output of the *k*th neuron of the *l*th hidden/output layer when the pattern *p* is fed, w_{jk}^{l-1} is the weight from the *j*th neuron in layer (*l*-1) to the *k*th neuron in layer *l*, and θ_k^l is the bias value of the *k*th neuron of the *l*th hidden/output layer, respectively. The cost function under which optimality is sought is the training mean square error, *MSE*, formulated as,

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

where t_k^p is the target (desired) output and $y_k^{p,O}$ is the actual output from the *k*th neuron in the output layer, *l*=*O*, for pattern *p* in the training set *T* with size *P*, respectively. At time *t*, suppose that the particle *a* in the swarm, $\xi = \{x_1,..,x_a,..,x_S\}$, has the positional component formed as, $xx_a^{xd_a(t)}(t) = \left\{ \{w_{jk}^0\}, \{w_{jk}^1\}, \{\theta_k^1\}, \{w_{jk}^2\}, \{\theta_k^2\}, ..., \{w_{jk}^{O-1}\}, \{\theta_k^{O-1}\}, \{\theta_k^O\} \right\}$ 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, a particle *a* in the swarm represents all potential network parameters of the MLP architecture at the dimension (hash index) $xd_a(t)$. As mentioned earlier, the dimension range, $D_{\min} \le xd_a(t) \le D_{\max}$, where MD PSO particles can make inter-dimensional jumps, is determined by the architecture space defined. Apart from the regular limits such as (positional) velocity range, $\{V_{\min}, V_{\max}\}$, dimensional velocity range,

$\{VD_{\min}, VD_{\max}\}$, the data space can also be limited with some practical range, i.e. $X_{\min} < xx_a^{xd_a(t)}(t) < X_{\max}$. In short, only some meaningful boundaries should be defined in advance for a MD PSO process, as opposed to other GA-based methods, which use several parameters, thresholds and some other (alien) techniques (e.g. SA, BP, etc.) in a complex process. Setting *MSE* in Eq. (3) as the fitness function enables MD PSO to perform *evolutions* of both network parameters and architectures within its native process.

IV. Experimental Results

In this section, we shall first demonstrate the *optimality* of the networks (with respect to the training mean square error as in Eq. (3)), which are automatically evolved by the MD PSO method according to the training set of an individual patient record in the benchmark database. We shall then present the overall results obtained from the ECG classification experiments and perform comparative evaluations against several state-of-the-art techniques in this field. Finally the robustness of the proposed system against variations of major parameters will be evaluated.

A. MD PSO Optimality Evaluation

In order to determine which network architectures are optimal (whether it is global or local) for a particular problem, we apply exhaustive BP training over every network configuration in the architecture space defined. As mentioned earlier, BP is a gradient descent algorithm and thus for a single run, it is susceptible to get trapped to the nearest local minimum. However, performing it a large number of times (e.g. K=500) with randomized initial parameters eventually increases the chance of converging to (a close vicinity of) the global minimum of the fitness function. 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 under the assumed criterion.

Due to the reasoning given earlier, the architecture space is defined over MLPs (possibly including one SLP) with the following activation function: *hyperbolic tangent* ($\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$). The input and output layer sizes are determined by the problem. We use a learning parameter for BP as $\lambda = 0.001$ and iteration number is 10000. We kept the default PSO parameters for MD PSO with a swarm size, *S* = 100, and velocity ranges are empirically set as $V_{\max} = -V_{\min} = X_{\max}/2$, and $VD_{\max} = -VD_{\min} = D_{\max}/2$. The dimension range is determined by the architecture space defined and the position range is set as $X_{\max} = -X_{\min} = 2$. Unless stated otherwise, these parameters are used in all experiments presented in this section.

In order to show the optimality of the network configurations evolved by MD PSO with respect to the MSE criterion, we first use the "limited" architecture space with 41 ANNs ($R^1 : R^1_{\min} = \{N_I, 1, 1, N_O\}$ and $R^1_{\max} = \{N_I, 8, 4, N_O\}$) containing the simplest 1, 2 or 3 layers MLPs with $L^1_{\min} = 1$, $L^1_{\max} = 3$, and $N_I = 11$, $N_O = 5$. We then select one of the most challenging records among MIT/BIH arrhythmia database, belonging to the patient record 222. For this record we perform 100 MD PSO runs with 100 particles, each of which terminates at the end of 1000 epochs (iterations). At the end of each run, the best fitness score (minimum MSE) achieved, $f(x\hat{y}^{dbest})$, by the particle with the index *gbest(dbest)* at the optimum dimension *dbest* is stored. The histogram of *dbest*, which is a hash index indicating a particular network configuration in R^1, eventually provides the crucial information about the (near-) optimal configuration(s).

Figure 5 shows *dbest* histogram and the error statistics plot from the exhaustive BP training the data of patient record 222. From the minimum (mean-square) error (*mMSE*) plot of the exhaustive BP training on top, it is clear that only four distinct sets of network configurations can achieve training *mMSEs* below 0.1. The corresponding indices (dimensions) to these four optimal networks are *dbest* = 9, 25, 33 and 41, where MD PSO managed to evolve either to them or to a neighbor (near-optimal) configuration. MD PSO can in fact achieve the best (lowest) training *MSEs* for two sets of configurations: *dbest* = 33 and 41 (including their 3 close neighbors). These are 3-layer MLPs; *dbest*= 33 is for *11x8x3x5* and *dbest*= 41 is for *11x8x4x5*. All MD PSO runs evolved either to *dbest*=25 (corresponding to configuration *11x8x2x5*) or to its neighbors, performed slightly worse than the best configurations. MD PSO runs, which evolved to the simplest MLPs with single hidden layer (i.e. *dbest*= 8 and 9 are for the MLPs *11x7x5* and *11x8x5*) achieved the worst *mMSE, about 15% higher than for dbest* = 33 and 41. The reason of MD PSO evolutions to those slightly worse configurations (for *dbest*=25 and particularly for *dbest*=9) is that MD PSO or PSO in general performs better in low dimensions. Furthermore, premature convergence is still a problem in PSO when the search space is in high dimensions [31]. Therefore, MD PSO naturally favors a low-dimension solution when it exhibits a competitive performance compared to a higher dimension counterpart. Such a natural tendency eventually yields the evolution process to compact network configurations in the architecture space rather than the complex ones, as long as optimality prevails.

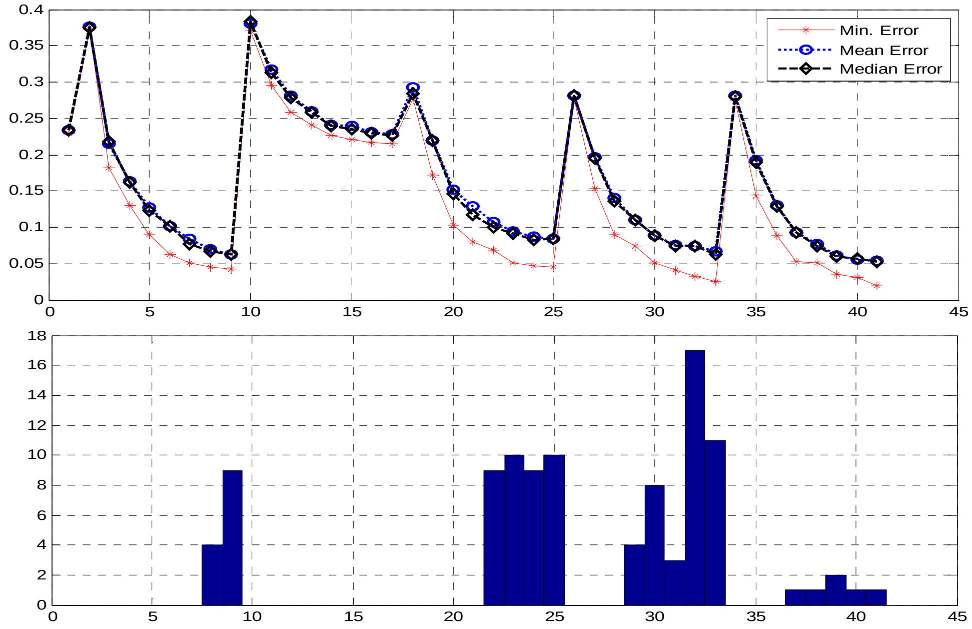


Figure 5: **Error (MSE) statistics from exhaustive BP training (top) and *dbest* histogram from 100 MD PSO evolutions (bottom) for patient record 222.**

B. *Classification Performance*

We performed classification experiments on 44 records of the MIT/BIH arrhythmia database, which includes a total of 100389 beats to be classified into five heartbeat types following the AAMI convention. For the classification experiments in this paper, the common part of the training data set contains a total of 245 representative beats, including 75 from each type N, -S, and -V beats, and all (13) type-F and (7) type-Q beats, *randomly* sampled from each class from the first 20 records (picked from the range 100 to 124) of the MIT/BIH database. The patient-specific training data includes the beats from the first 5 min of the corresponding patient’s ECG record. Patient-specific feed-forward MLP networks are trained with a total of 245 common training beats and a variable number of patient-specific beats depending on the patient's heart rate, so only less than 1% of the total beats are used for training each neural network. The remaining beats (25 min) of each record, in which 24 out of 44 records are completely new to the classifier, are used as test patterns for performance evaluation.

Table I: Summary table of beat-by-beat classification results for all 44 records in the MIT/BIH arrhythmia database. Classification results for the testing dataset only (24 records from the range 200 to 234) are shown in parenthesis.

		Classification Result				
		N	**S**	**V**	**F**	**Q**
Ground Truth	**N**	73019 (40532)	991 (776)	513 (382)	98 (56)	29 (20)
	S	686 (672)	1568 (1441)	205 (197)	5 (5)	6 (5)
	V	462 (392)	333 (299)	4993 (4022)	79 (75)	32 (32)
	F	168 (164)	28 (26)	48 (46)	379 (378)	2 (2)
	Q	8 (6)	1 (0)	3 (1)	1 (1)	1 (0)

Table I summarizes beat-by-beat classification results of ECG heartbeat patterns for all test records. Classification performance is measured using the four standard metrics found in the literature [11]: classification accuracy (*Acc*), sensitivity (*Sen*), specificity (*Spe*), and positive predictivity (*Ppr*). While accuracy measures the overall system performance over all classes of beats, the other metrics are specific to each class and they measure the ability of the classification algorithm to distinguish certain events (i.e. VEBs or SVEBs) from nonevents (i.e. non-VEBs or non-SVEBs). The respective definitions of these four common metrics using true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*) are as follows: *Accuracy* is the ratio of the number of correctly classified patterns to the total number of patterns classified, $Acc = (TP+TN)/(TP+TN+FP+FN)$; *Sensitivity* is the rate of correctly classified events among all events, $Sen = TP/(TP+FN)$; *Specificity* is the rate of correctly classified nonevents among all nonevents, $Spe = TN/(TN+FP)$; and *Positive Predictivity* is the rate of correctly classified events in all detected events, $Ppr = TP/(TP+FP)$. Since there is a large variation in the number of beats from different classes in the training/testing data (i.e. 39465/50354 type-N, 1277/5716 type-V, and 190/2571 type-S beats), sensitivity, specificity, and positive predictivity are more relevant performance criteria for medical diagnosis applications.

Table II: VEB and SVEB classification performance of the proposed method and comparison with the three major algorithms from the literature.

Methods	**VEB**				**SVEB**			
	Acc	***Sen***	***Spe***	***Ppr***	***Acc***	***Sen***	***Spe***	***Ppr***
Hu et al. [11] [1]	94.8	78.9	96.8	75.8	N/A	N/A	N/A	N/A
Chazal et al. [15] [1]	96.4	77.5	98.9	90.6	92.4	76.4	93.2	38.7
Jiang and Kong [16] [1]	98.8	94.3	99.4	95.8	97.5	74.9	98.8	78.8
Proposed [1]	97.9	90.3	98.8	92.2	96.1	81.8	98.5	63.4
Jiang and Kong [16] [2]	98.1	86.6	99.3	93.3	96.6	50.6	98.8	67.9
Proposed [2]	97.6	83.4	98.1	87.4	96.1	62.1	98.5	56.7
Proposed [3]	98.3	84.6	98.7	87.4	97.4	63.5	99.0	53.7

[1] The comparison results are based on 11 common recordings for VEB detection and 14 common recordings for SVEB detection.

[2] The VEB and SVEB detection results are compared for 24 common testing records only.

[3] The VEB and SVEB detection results of the proposed system for all training and testing records.

The proposed system is compared with three existing algorithms, [11], [15] and [16], which comply with the AAMI standards and use *all* records from the MIT/BIH arrhythmia database. For comparing the performance results, the problem of VEB and SVEB detection is considered individually. The VEB and SVEB classification results of the proposed technique over all 44 records are summarized in Table II. The performance results for VEB detection in the first four rows of Table II are based on 11 test recordings (200, 202, 210, 213, 214, 219, 221, 228, 231, 233, and 234) that are common to all four methods. For SVEB detection, comparison results are based on 14 common recordings (with the addition of records 212, 222, and 232) between the proposed technique and the methods in [15] and [16]. Several interesting observations can be made from these results. First, for SVEB detection, sensitivity and positive predictivity rates are comparably lower than VEB detection, while a high specificity performance is achieved. The reason for the worse classifier performance in detecting SVEBs is that SVEB class is under-represented in the training data and hence more SVEB beats are misclassified as normal beats. Overall, the performance of the proposed technique in VEB and SVEB detection is significantly better than [11] and [15] for all measures and is comparable to the results obtained with evolvable BbNNs in [16]. Moreover, it is observed that the proposed classifier achieves comparable performance over the training and testing set of patient records. It is worth noting that the number of training beats used for each patient's classifier was less than 2% of all beats in the training data set and the resulting classifiers designed by the MD PSO process have improved generalization ability, i.e. the same low number of design parameters are used for all networks.

C. Robustness

In order to investigate the robustness of the proposed technique against the variations of the few PSO parameters used, such as the swarm size *S*, the iteration number *I*, and to evaluate the effect of the architecture space (and hence the characteristics of the ANNs used), we performed four classification experiments over the MIT/BIH arrhythmia database (I-IV) and their classification accuracy results per VEB and SVEB, are presented in Table III. Experiments I-III are performed over the same architecture space, with 1-, 2- and 3-layer MLP architectures defined by $R^1{}_{\min} = \{11, 8, 4, 5\}$, $R^1{}_{\max} = \{11, 16, 8, 5\}$. Between I and II, the swarm size, and between II and III, the iteration number are changed significantly, whereas in IV an entirely different architecture space containing 4-layer MLPs is used. From the table, it is quite evident that the effects of such major variations over the classification accuracy are insignificant. Therefore, any set of common PSO parameters within a reasonable range can be conveniently used within the proposed technique. Furthermore, for this ECG database, the choice of the architecture space does not affect the overall performance, yet any other ECG dataset containing more challenging ECG data might require the architecture spaces such as in IV in order to obtain a better generalization capability.

Table III: VEB and SVEB classification accuracy of the proposed method for different PSO parameters and architecture spaces.

(%)	I	II	III	IV
VEB	98.3	98.2	98.3	98.0
SVEB	97.4	97.3	97.1	97.4

I: $R^1{}_{\min} = \{11, 8, 4, 5\}$, $R^1{}_{\max} = \{11, 16, 8, 5\}$, *S*=100, I=500

II: $R^1{}_{\min} = \{11, 8, 4, 5\}$, $R^1{}_{\max} = \{11, 16, 8, 5\}$, *S*=250, I=200

III: $R^1{}_{\min} = \{11, 8, 4, 5\}$, $R^1{}_{\max} = \{11, 16, 8, 5\}$, *S*=80, I=200

IV: $R^1{}_{\min} = \{11, 6, 6, 3, 5\}$, $R^1{}_{\max} = \{11, 12, 10, 5, 5\}$, *S*=400, I=500

V. CONCLUSIONS

In this work, we proposed an automated patient-specific ECG heartbeat classifier, which is based on an efficient formation of morphological and temporal features from the ECG data and evolutionary neural network processing of the input patterns individually for each patient. The translation-invariant DWT and the PCA are the principal signal processing tools employed in the proposed feature extraction scheme.

The wavelet-based morphology features are extracted from the ECG data and are further reduced to a lower-dimensional feature vector using PCA technique. Then, by combining compact morphological features with the two critical temporal features, the resultant feature vector to represent each ECG heartbeat is used as the input to MLP classifiers, which are automatically designed (both network structure and connection weights are optimized) using the proposed MD-PSO technique.

With the proper adaptation of the native MD PSO process, the proposed method can thus evolve to the optimum network within an architecture space and for a particular problem according to a given error function. It is furthermore generic as it is applicable to any type of ANNs in an architecture space with varying size and properties, as long as a proper hash function enumerates all possible configurations in the architecture space with respect to their complexity into proper hash indices representing the dimensions of the solution space over which a MD PSO process seeks for the optimal solution. MD PSO evolutionary process has a simple and unique structure for evolving ANNs without any significant parameter dependency or interference of any other method. It also has a native characteristic of having better and faster convergence to optimum solution in low dimensions, and consequently leads to *compact* networks.

The results of the classification experiments, which are performed over the benchmark MIT/BIH arrhythmia database show that the proposed classifier technique can achieve average accuracies and sensitivities better than most of the existing algorithms for classification of ECG heartbeat patterns according to the AAMI standards. An overall average accuracy of 98.3% and an average sensitivity of 84.6% for VEB detection, and an average accuracy of 97.4% and an average sensitivity of 63.5% for SVEB detection were achieved over all 44 patient records from the MIT/BIH database. The overall results promise a significant improvement over other major techniques in the literature with the exception of the BbNN-based personalized ECG classifier in [16], which gave comparable results over the test set of 24 records. However, it used many critical parameters, Back propagation and a specific ANN architecture, which may not suit feature vectors in higher dimensions. The proposed method is based only on well-known, standard techniques such as DWT and PCA, whilst using the most typical ANN structure, the MLPs. Experimental results approve that its performance is not affected significantly by variations of the few parameters used. Therefore, the resulting classifier successfully achieves the main design objectives, i.e. maintaining a robust and generic architecture with superior classification performance. As a result it can be conveniently applied to any ECG database “as is”, alleviating the need of human “expertise” and “knowledge” for designing a particular ANN.

APPENDIX

In a MD PSO process at time (iteration) *t*, each particle *a* in the swarm, $\xi = \{x_1,..,x_a,..,x_S\}$, is represented by the following characteristics:

$xx_{a,j}^{xd_a(t)}(t)$: *j*th component (dimension) of the position in dimension $xd_a(t)$

$vx_{a,j}^{xd_a(t)}(t)$: *j*th component (dimension) of the velocity in dimension $xd_a(t)$

$xy_{a,j}^{xd_a(t)}(t)$: *j*th component (dimension) of the personal best (*pbest*) position in dimension $xd_a(t)$

gbest(d) : Global best particle index in dimension *d*

$x\hat{y}_j^d(t)$: *j*th component (dimension) of the global best position of swarm, in dimension *d*

$xd_a(t)$ **:** Dimension component

$vd_a(t)$**:** Velocity component of dimension

$x\tilde{d}_a(t)$**:** Personal best dimension component

Let *f* denote the dimensional fitness function that is to be optimized within a certain dimension range, ($D_{\min} \leq N \leq D_{\max}$). Without loss of generality, assume that the objective is to find the minimum (position) of *f* at the optimum dimension within a multi-dimensional search space. Assume that the particle *a* visits (back) the same dimension after *T* iterations (i.e. $xd_a(t) = xd_a(t+T)$**)**, then the personal best position can be updated in iteration *t*+*T* as follows,

$$xy_{a,j}^{xd_a(t+T)}(t+T) = \left\{ \begin{matrix} xy_{a,j}^{xd_a(t)}(t) & if \;\; f(xx_a^{xd_a(t+T)}(t+T)) > f(xy_a^{xd_a(t)}(t)) \\ xx_{a,j}^{xd_a(t+T)}(t+T) & else \end{matrix} \right\} j = 1,2,...,N \tag{4}$$

Furthermore, the personal best dimension of particle *a* can be updated in iteration *t*+1 as follows,

$$x\tilde{d}_a(t+1) = \left\{ \begin{matrix} x\tilde{d}_a(t) & if \;\; f(xx_a^{xd_a(t+1)}(t+1)) > f(xy_a^{x\tilde{d}_a(t)}(t)) \\ xd_a(t+1) & else \end{matrix} \right\} \tag{5}$$

Recall that *gbest(d)* is the index of the global best particle at dimension *d* and let *S(d)* be the total number of particles in dimension *d*, then $x\hat{y}^{dbest}(t) = xy_{gbest(dbest)}^{dbest}(t) = \min(xy_1^{dbest}(t),..,xy_S^{dbest}(t))$. For a particular iteration *t*, and for a particle $a \in \{1, S\}$, first the positional components are updated in its current dimension, $xd_a(t)$ and then the dimensional update is performed to determine its next (*t*+1)st dimension, $xd_a(t+1)$**.** The positional update is performed for each dimension component, $j \in \{1, xd_a(t)\}$ as follows:

$$\begin{gathered} vx_{a,j}^{xd_a(t)}(t+1) = w(t)\, vx_{a,j}^{xd_a(t)}(t) + c_1 r_{1,j}(t)\left(xy_{a,j}^{xd_a(t)}(t) - xx_{a,j}^{xd_a(t)}(t)\right) + c_2 r_{2,j}(t)\left(x\hat{y}_j^{xd_a(t)}(t) - xx_{a,j}^{xd_a(t)}(t)\right) \\ xx_{a,j}^{xd_a(t)}(t+1) = xx_{a,j}^{xd_a(t)}(t) + vx_{a,j}^{xd_a(t)}(t+1) \end{gathered} \tag{6}$$

Note that the particle's new position, $xx_a^{xd_a(t)}(t+1)$, will still be in the same dimension, $xd_a(t)$; however, the particle may fly to another dimension afterwards with the following dimensional update equations:

$$\begin{gathered} vd_a(t+1) = \left\lfloor vd_a(t) + c_1 r_1 \; (t)\left(x\tilde{d}_a(t) - xd_a(t)\right) + c_2 r_2(t)\left(dbest - xd_a(t)\right) \right\rfloor \\ xd_a(t+1) = xd_a(t) + vd_a(t+1) \end{gathered} \tag{7}$$

where $\lfloor . \rfloor$ is the *floor* operator. Accordingly, the general pseudo-code of the MD PSO method can be expressed as follows:

```
    MD PSO ( termination criteria:{IterNo, $\varepsilon_C$, ...}, $V_{\max}$, $VD_{\max}$, $D_{\min}$, $D_{\max}$ )
1.  For $\forall a \in \{1, S\}$ do:
    1.1.  Randomize $xd_a(0), vd_a(0)$
    1.2.  Initialize $\tilde{xd}_a(0) = xd_a(0)$
    1.3.  For $\forall d \in \{D_{\min}, D_{\max}\}$ do:
        1.3.1.  Randomize $xx_a^d(0)$, $xv_a^d(0)$
        1.3.2.  Initialize $xy_a^d(0) = xx_a^d(0)$
    1.4.  End For.
2.  End For.
3.  For $\forall t \in \{1, IterNo\}$ do:
    3.1.  For $\forall a \in \{1, S\}$ do:
        3.1.1.  If ( $f(xx_a^{xd_a(t)}(t)) < \min\left( f(xy_a^{xd_a(t)}(t-1), \min_{p \in S-\{a\}}\left(f(xx_p^{xd_a(t)}(t))\right)\right)$ ) then do:
            3.1.1.1.  $xy_a^{xd_a(t)}(t) = xx_a^{xd_a(t)}(t)$
            3.1.1.2.  If ( $f(xx_a^{xd_a(t)}(t)) < f(xy_{gbest(xd_a(t))}^{xd_a(t)}(t-1))$ ) then $gbest(xd_a(t)) = a$
            3.1.1.3.  If ( $f(xx_a^{xd_a(t)}(t)) < f(xy_a^{\tilde{xd}_a(t-1)}(t-1))$ ) then $\tilde{xd}_a(t) = xd_a(t)$
            3.1.1.4.  If ( $f(xx_a^{xd_a(t)}(t)) < f(x\hat{y}^{dbest}(t-1))$ ) then $dbest = xd_a(t)$
        3.1.2.  End If.
    3.2.  End For.
    3.3.  If the termination criteria are met, then Stop.
    3.4.  For $\forall a \in \{1, S\}$ do:
        3.4.1.  For $\forall j \in \{1, xd_a(t)\}$ do:
            3.4.1.1.  Compute $vx_{a,j}^{xd_a(t)}(t)$ using (6)
            3.4.1.2.  if( $\left|vx_{a,j}^{xd_a(t)}(t)\right| > V_{\max}$ ) then clamp it to $\left|vx_{a,j}^{xd_a(t)}(t)\right| = V_{\max}$
            3.4.1.3.  Compute $xx_{a,j}^{xd_a(t)}(t)$ using (6)
        3.4.2.  End For.
        3.4.3.  Compute $vd_a(t)$ using (7)
        3.4.4.  if( $\left|vd_a(t+1)\right| > VD_{\max}$ ) then clamp it to $\left|vd_a(t+1)\right| = VD_{\max}$
        3.4.5.  Compute $xd_a(t+1)$ using (7)
        3.4.6.  if( $xd_a(t+1) < D_{\min}$ ) then $xd_a(t+1) = D_{\min}$ )
        3.4.7.  if( $xd_a(t+1) > D_{\max}$ ) then $xd_a(t+1) = D_{\max}$ )
    3.5.  End For.
4.  End For.
```

To avoid arbitrary high values (i.e. exploding) thereby causing premature convergence [30], along with the (positional) velocity limit $V_{\max}$, two more clamping are applied for dimensional PSO components, such as $\left|vd_{a,j}(t+1)\right| < VD_{\max}$ and the initial dimension range set by the user, $D_{\min} \le xd_a(t) \le D_{\max}$. Once the MD PSO process terminates, the optimum solution will be $x\hat{y}^{dbest}$ at the optimum dimension, *dbest*, achieved by the particle with the index *gbest(dbest)* and finally the best (fitness) score achieved will naturally be $f(x\hat{y}^{dbest})$.

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] K. Minami, H. Nakajima, and T. Toyoshima, “Real-Time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network,” IEEE Trans. Biomed. Eng., vol. 46, no. 2, pp. 179–185, Feb. 1999.

[3] 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.

[4] 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.

[5] 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.

[6] J. L. Willems and E. Lesaffre, “Comparison of multigroup logisitic and linear discriminant ECG and VCG classification,” J. Electrocardiol., vol. 20, pp. 83-92, 1987.

[7] J. L. Talmon, Pattern Recognition of the ECG. Berlin, Germany: Akademisch Proefscrift, 1983.

[8] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, “An approach to cardiac arrhythmia analysis using hidden Markov models,” IEEE Trans. Biomed. Eng., vol. 37, no. 9, pp. 826–836, Sep. 1990.

[9] S. Osowski, L. T. Hoai, and T. Markiewicz, “Support vector machine based expert system for reliable heartbeat recognition,” IEEE Trans.Biomed. Eng., vol. 51, no. 4, pp. 582–589, Apr. 2004.

[10] Y. H. Hu, W. J. Tompkins, J. L. Urrusti, and V. X. Afonso, “Applications of artificial neural networks for ECG signal detection and classification,” J. Electrocardiol., pp. 66–73, 1994.

[11] 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.

[12] S. C. Lee, “Using a translation-invariant neural network to diagnose heart arrhythmia,” in IEEE Proc. Conf. Neural Information Processing Systems, Nov. 1989.

[13] P. de Chazal and R. B. Reilly, “A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features,” IEEE Trans. Biomed. Eng., vol. 53, no. 12, pp. 2535–2543, Dec. 2006.

[14] “Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms,” Association for the Advancement of Medical Instrumentation, Arlington, VA, 1987.

[15] 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.

[16] W. Jiang and S. G. Kong, “Block-based neural networks for personalized ECG signal classification,” IEEE Trans. Neural Networks, vol. 18, no. 6, pp. 1750–1761, Nov. 2007.

[17] 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.

[18] R. Silipo, C. Marchesi, “Artificial neural networks for automatic ECG analysis,” IEEE Trans. Signal Processing, vol. 46, no. 5, pp. 1417–1425, May. 1998.

[19] S. Osowski and T. L. Linh, “ECG beat recognition using fuzzy hybrid neural network,” IEEE Trans. Biomed. Eng., vol. 48, pp. 1265–1271, Nov. 2001.

[20] 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.

[21] R. Silipo, et al., “ST-T segment change recognition using artificial neural networks and principal component analysis, ” Computers in Cardiology, pp. 213-216, 1995.

[22] X. Yao and Y. Liu, “A new evolutionary system for evolving artificial neural networks”, IEEE Trans. On Neural Networks, vol.8, no.3, pp. 694–713, May 1997.

[23] T. Ince, S. Kiranyaz, and M. Gabbouj, “Automated patient-specific classification of premature ventricular contractions,” in IEEE Proc. Int. Conf. EMBS, pp. 5474-5477, 2008.

[24] R. Mark and G. Moody, “MIT-BIH Arrhythmia Database Directory,” [Online]. Available: http://ecg.mit.edu/dbinfo.html

[25] J. Pan and W. J. Tompkins, “A real-time QRS detection algorithm,” IEEE Trans. Biomed. Eng., vol. 32, No. 3, pp. 230–236, Mar. 1985.

[26] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, CA: Academic Press, 1999.

[27] 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.

[28] N. V. Thakor, J. G. Webster, and W. J. Tompkins, “Estimation of QRS complex power spectra for design of a QRS filter,” IEEE Trans. Biomed. Eng., vol. BME-31, pp. 702-705, 1984.

[29] S. Pittner and S. V. Kamarthi, “Feature extraction from wavelet coefficients for pattern recognition tasks,” IEEE Trans. Pattern Anal. Machine Intell., vol. 21, pp. 83–88, Jan 1999.

[30] J. Kennedy, R Eberhart., “Particle swarm optimization”, in Proc. of IEEE Int. Conf. On Neural Networks, vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[31] F. Van den Bergh, An Analysis of Particle Swarm Optimizers, PhD thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.