Comparison of ECG Beat Classification methods on a Mobile System

Fernando Arena Varella[1,2] favarella@inf.ufrgs.br

Guilherme L. de Lima[1] gllima@inf.ufrgs.br

Cirano Iochpe[1] ciochpe@inf.ufrgs.br

Valter Roesler[1] roesler@inf.ufgrs.br

1 Universidade Federal do Rio Grande do Sul

2 I9Access Tecnologia Ltda

Av. Bento Gonçalves, nº 9500, Setor 4 – Prédio 72 – Sala 211 - Agronomia –

Porto Alegre/RS - Brasil

ABSTRACT

The constant evolution of mobile technologies has lead to several systems with focus on the mobile electrocardiogram (ECG) telemonitoring of patients. They usually present a client-server architecture, where a web server centralizes the storage, management and processing of the patient's signals, and a mobile phone is responsible for acquiring and transmitting the data over the internet (WAP or 3G) to the server. However, there are countries, like Brazil, whose mobile network services do not present full coverage, even in large urban areas. This leads to a lack of security under certain situations like emergency calls or critic patient's monitoring when the patient is in an area without mobile network coverage. In these situations, the patient's vital signals must be analyzed as soon as possible, in order to allow fast preventive or reactive actions to be taken. An alternative way to deal with this problem is to support the analysis of vital signals in a mobile phone which can be either offline or communicating through a cell phone network. Though, since mobile phones, compared to PCs, have slower CPUs and less internal memory, it's important to select an ECG wave analysis method that is not costly with respect to computation time and memory usage. In this paper, we present comparison study made in order to select a reliable method for ECG beat classification running in a mobile phone. Three ECG beat classification methods were selected to be analyzed and implemented in a mobile phone. Tests were made in regions with limited cell phone network coverage in south Brazil. Test results were compared in order to choose a method whose implementation is efficient enough to run in a mobile phone and can achieve high accuracy on the classification of the ECGs, even if running in a phone with limited CPU as well as memory resources.

Keywords

Remote Home Care, ECG Beat Classification, Mobile Systems, Emergency ECG Monitoring

1. INTRODUCTION

Heart and cardiocirculatory diseases are the most common causes of death in the majority of western countries, including Brazil where more than 30% of the deaths are related to them [1]. An important characteristic of some heart disease treatments is the need for constant monitoring of the patient´s heart beat waves. This monitoring is typically performed by either an electrocardiographer or a holter. More important yet, in emergency situations, it is crucial the fast access to the patient's vital signals by the physician, allowing him or her to take preventive or reactive actions in order to save the patient's life.

The evolution of telecommunications helped the development of several systems with focus on the ECG telemonitoring of patients, allowing them to stay at home, instead of in a hospital in various situations. It helps to reduce the high cost with patients internment and transport and to shorten the long waits for hospital beds. Another usage for the ECG telemonitoring is in critical situations like strokes or heart stops, when the patient's signals must be analyzed as soon as possible. Most of the systems used in such situations rely on a mobile phone to transmit the data acquired by the electrocardiographer to a central server, where it can be analyzed by the physicians almost in real time. However, in countries like Brazil, where mobile network coverage is not ideal, even in large urban areas with dense people concentration, this strategy can be risky to the patients since network communication is not reliable in a number of regions. To deal with this problem, we suggest to perform the ECG analysis through an application running in the cell phone itself when it is impossible to transmit the ECG waves to the remote server. This could make the overall system fail-safe with respect to the lack of good network coverage.

Many computational methods have been proposed to automate the process of ECG waves analysis, helping physicians to diagnose anomalies on ECG exams. Although these methods have achieved very good performances, many of them rely on heavy computational algorithms and can be very costly in terms of CPU and memory usage. Considering this, we present, in this paper, a comparison of some of the most reliable ECG beat classification methods.

Our main objective is to point out one or more methods that can successfully classify ECG beats with high accuracy and fast response even when implemented and running in a mobile phone with limited CPU as well as memory resources.

The next sections are organized as follows. Section 2 introduces the ECG beat classification problem and the most commonly used techniques to solve it. Section 3 is reports on the implementation of some classification methods in mobile devices. In section 4 we comment on the results obtained in the evaluation of the implemented methods. Finally, in Section 5 we present our conclusions and point out to future work.

2. ECG ANALYSIS BACKGROUND

The automatic ECG classification methods apply techniques from several computing areas, like computational statistics, pattern recognition, support vectors machines, etc. However, there is a basic set of common steps between these methods: Beat Detection, Feature Extraction and Beat Classification. The next sections will introduce the ECG classification process and review the most used techniques.

2.1 ECG Analysis Process

The Beat Detection step is usually made using common annotated ECG database, easing the detection process and unifying the data sets used amongst the majority of the authors. When applied to real ECGs, this stage must be made by a QRS detector algorithm, like the Pan Tompkins method [2].

The Feature Extraction follows the beat detection, and it is responsible for transforming the raw ECG signal, mapping the original values to meaningful features. This is a very important stage, and the most successful classification methods rely on a good feature extraction stage to achieve high performances.

Finally, the Beat Classification stage will be performed over the features extracted on the second stage. The authors employ several classification and pattern recognition techniques, allowing them to distinguish the type of each beat that is being classified.

Many authors have already employed these steps on the past, building classifiers that can achieve high accuracy rates on the classification task. We analyzed the results and implementation of many methods, aiming the selection of one of them for the usage on a mobile system. Thus, the method must be efficient in time and in memory usage, at the same time as it must achieve high accuracy to guarantee that it can improve the capacity of the system to save human lives.

Mehmet Engin [3] developed a method that unifies a Multilayer Perceptron (MLP) Artificial Neural Network (ANN) with the fuzzy c-means (FCM) clustering algorithm, forming a neuro-fuzzy network. His method uses the discrete wavelet transform (DWT) to extract the feature set. Similarly, Güler e Übeyli [4] proposed a method that uses a set of small MLPs combined with a larger MLP to perform the classification task and has its feature extraction step based on the DWT. In this case, the data is first classified by a set of MLPs, and then classified again by a final MLP. Yu and Chen [5] also used the DWT for the feature extraction, but the classification step is made by a Probability Neural Network. On their other work [6], they allied the DWT with the Higher Order Statistics (HOS) to build the feature set, and used a MLP as the classifier. Yu and Chou [7] introduced a switchable scheme for the classification task. They have 2 feature sets, both extracted using the DWT allied to the Independent Component Analysis (ICA). Depending on some characteristics of the feature sets, they switch between the minimum Euclidean Distance classifier, minimum Mahalanobis Distance classifier or a Bayesian classifier. Finally, Minhas and Arif [8] developed a method similar to the Chen and Yu's method, they rely on the HOS and the DWT to build the feature set and perform the classification with the K-Nearest Neighbor (KNN) Neural Network.

However there is a basic set of common steps between these methods: Beat Detection, Feature Extraction and Classification.

The Beat Detection is a signal processing consisting of QRS complex detection. The QRS complex detection methods already have excellent solutions and are not addressed in this work. What is true highlight is that the following step will process only segments of the QRS complexes. The feature extraction stage consists in transforming the signal into a suitable format to perform the classification. Finally we perform classification of extracted features in a certain class amongst the classes that the method is able to identify, such as normal beat, pacemaker beat, etc.

2.2 Materials and Methods

The three methods that this work analyzes uses discrete wavelet transform together with mathematical procedures to extract the features that will be classified.

2.2.1 Discrete Wavelet Transform

The wavelet functions are widely used in signal processing. While the Fourier Transforms work only in frequency domain, the wavelet analysis works also in the time domain, a fact which facilitates the analysis of local characteristics of signal, what is very important for classification of ECG beats. Both the Fourier analysis and wavelet analysis intend to approximate a signal by a linear combination of sines and cosines (Fourier analysis) or wavelets. The approximation of the signal in wavelet analysis is performed by a process called wavelet transform which has the continuous and discrete variants.

The continuous wavelet transform (Continuous Wavelet Transform - CWT) of a signal f is the inner product of the signal by a wavelet, in this case called mother wavelet ψ, as it appears in the following formula:

$$\left(W_{\psi} f\right)(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\, \psi\left(\frac{t-b}{a}\right) dt$$

Where:

$$a, b \in \mathbb{R}, a \neq 0$$

The mother wavelet ψ is given by the following formula:

$$\psi_{j,k}(t) = \psi(2^t - k) \qquad j, k \in \mathbb{Z}$$

The Discrete Wavelet Transform (DWT) of a signal expressed as a vector of samples X = (x_0, $x_{1,}$, …, x_T)^t can be expressed as:

$$d_{j,k}^{(\psi)} = \sum_{i=0}^{T-1} X_t \psi_{j,k}(t/T)$$

However, in practice, this turns into the use of a pyramidal algorithm, where a series of low pass and high-pass filters are applied to the signal. The Pyramid algorithm that computes the wavelet transform was proposed by Stephane G. Mallat [11], and is shown graphically in Figure 1.

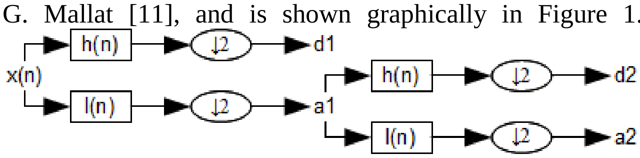

Figure 1. Two levels of the Mallat's DWT

The algorithm uses a low-pass filters *l* and *h* high-pass to calculate the approximation and detail coefficients, respectively. On the first level, the approximation coefficients are calculated by the convolution of the signal with the low-pass filter, and the detail coefficients by the convolution with the high-pass filter. After this process, the resulting coefficients are downsampled, since the convolution modifies the size of the resulting signals. For every new level of coefficients, the approximation coefficients from the previous level are taken as the original signal, and the same process is applied.

The filters depend on the type of wavelet-mother used on the transform. The most commonly used is the Haar wavelet, and its low-pass and high-pass filters are shown below:

h = [0.70710678118654757, 0.70710678118654757]

l = [0.70710678118654757, 0.70710678118654757]

Some authors use the *à trous* wavelet transform [12], whose main difference to the Mallat's algorithm is the lack of the downsampling after the application of the filters. This difference helps to preserve the temporal resolution and the time-invariant property of the signal on different scales [13].

2.2.2 Statistical Measurements

Several statistical measurements are used by the ECG beat classification methods, including the variance, standard deviation, auto correlation and relative amplitudes.

The variance is defined like the average of the square of the distance of each value from the mean, it shown the difference of a value in relation to the mean value. Can be expressed by the following formula:

$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^{N} [x(n) - \bar{x}]^2$$

From the variance we can calculate the standard deviation which is defined as the square root of the variance, or even the average distance of each value to the average, which shows how the sample is distributed to the extent that deviates from the mean.

Another important measure is the auto correlation, which is considered a measure of similarity between a sample and its shifted version. The auto correlation can be expressed as follows:

$$R_{xx}(l) = \sum_{n=i}^{N-|k|-1} x(n)x(n-1)$$

where x(n) is a sample of size N. This calculation tells how much a certain value influences the neighborhood values, i.e., how related are the values of a sample.

We still have the relative amplitude, which is nothing more than the ratio between the minimum and maximum value for a given sample, which shows morphological characteristics of the sample.

2.2.3 Principal Components Analysis

The Principal Component Analysis (PCA) reduces the dimension of matrices building a smaller matrix only with the principal information of the original one. For example, considering the training and testing sets that will be classified, it is possible to reduce a 10-dimensional feature set to a 5-dimension, extracting the higher five principal components. The process to find principal components is the process to find eigenvectors of a covariance matrix *R* formed by the feature set. The covariance matrix *R* is given by:

$$R = \frac{1}{N} \sum_{i=1}^{N} (F_i - \mu)(F_i - \mu)^t$$

The eigenvalues and eigenvectors are the obtained solving the eigenvalue problem. For this problem, ten eigenvalues and ten eigenvectors are obtained, the five eigenvectors corresponding to the highest five eigenvalues will be used to build a new feature set.

The new feature set is obtained by the multiplication of each sample by the five eigenvectors, y^k:

$$y^k = \phi^t (F_1^k - \mu)$$

2.2.4 Higher-order Statistics - Cumulants

A cumulant is a Higher Order Statistics measure that is commonly used in physics. They can be seen as a set of quantities that provide an alternative to the moments of the signal, providing useful information about the probabilities distribution of the signal [14].

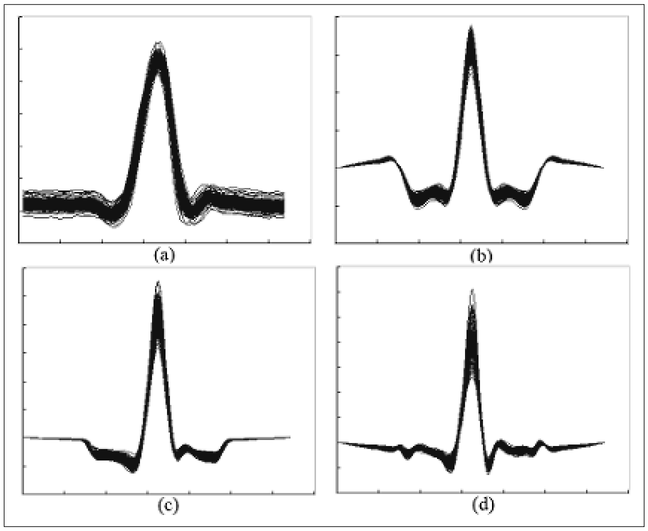

Figure 2. 200 samples of a NORMAL beat: (a) original sample, (b) 2nd order cumulants, (c) 3rd order cumulants and (d) 4th order cumulants

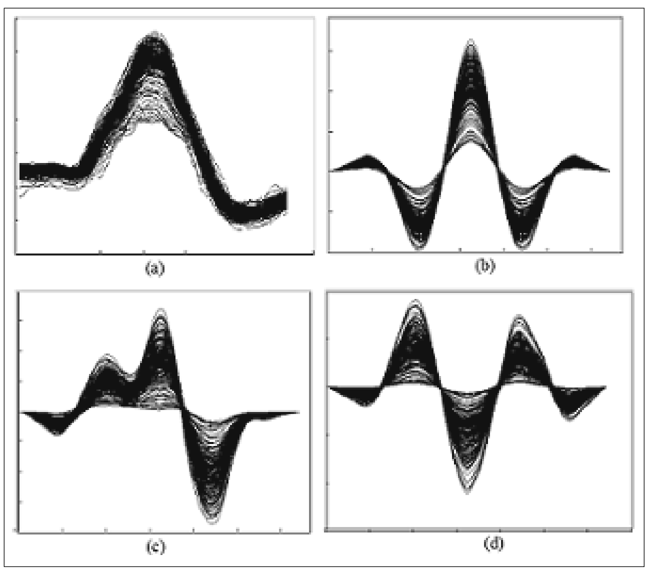

Figure 3. 200 samples of a Left Bundle Branch Block beat: (a) original sample, (b) 2nd order cumulants, (c) 3rd order cumulants and (d) 4th order cumulants

As it can be seen on Figure 2 and Figure 3, the employment of the cumulants on the task of classifying ECG beats can help to remove the variability between beats from the same type, and to enhance the differences between beats of the different types.

2.2.5 Probabilistic Neural Network

The Probabilistic Neural Network (PNN) is a neural network used primarily as a classifier, but it can also be seen as an alternative way to implement a statistical algorithm called discriminant [5]. It is instance-based, since it depends on the presence of the training instances on the classification stage. The network's topology is composed of four layers: Input Layer, Pattern Layer, Summation Layer and Output Layer. That topology can be seen in Figure 2.

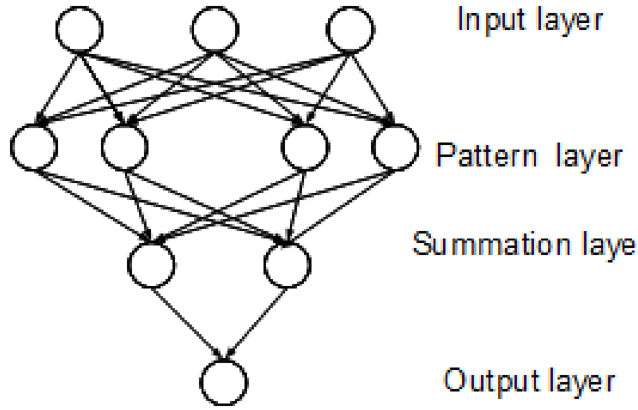

Figure 4. Probabilistic Neural Network Topology

The input layer distributes the samples to neurons of pattern layer, and all entries are propagated to all the neurons of layer pattern. Each sample of the training set corresponds to a unit-layer pattern, thus, the hidden layer will have the same number of neurons as the size of the training set. Each hidden layer neuron has a multi-dimensional Gaussian kernel, capable of estimating the probability density function (PDF) of the input relatively to that pattern. The summation layer has the same number of neurons as the number of classes on the training set, and each neuron in this layer responsible for accumulating the PDFs of all neurons from the pattern layer that have the same class.

The classification phase of this neural network is simple, yet costly. Each input from the training set will have its PDF estimated for every neuron from the Pattern Layer. The next layer, the Summation Layer will compute the probability of the sample of being of each class accumulating the values from the pattern layer and grouping them by class. Finally, the Output Layer will select the neuron with a higher value from the Summation Layer to be the class of the sample.

To estimate the probability density function of a given sample is performed using the kernel density estimation approach, also called Parzen window, which is basically a technique of data interpolation. Given an instance x, the Parzen window PDF estimates b fx(x) for each sample data set. As the following expression:

$$\hat{f}_x(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \quad (1)$$

where K is the kernel function, h is a smoothing parameter and n is the number of samples in the data set.

The kernel function most used is a Gaussian that can be represented by:

$$H(v) = (2\pi)^{-1/2}e^{-1/2}v^2$$

Thus, the expression (1) is denoted by:

$$f(x) = \frac{1}{(2\pi)^{p/2}\sigma^p}\frac{1}{m}\sum_{i-1}^{m} exp[\frac{-(X - X_i)^t(X - X_i)}{2\sigma^2}]$$

Finally, the PNN has fast training, the training set is used directly in the kernel, and converges to the optimal Bayesian decision surface.

2.2.6 *Nearest Neighbor*

The K Nearest Neighbors algorithm calculates the euclidean distance of a testing sample with all the training samples. The input sample is classified with the same class of the mode class of the k nearest neighbor according to the euclidean distance [8]. The pseudo code is described in the figure 3.

```
Input  : sample, training set, k
Output: class

for i ← 1 to #training_Set do
 | aux[i] ← euclidianDistance((sample, training_Set[i]));
end
aux ← sort((aux));
class ← mode((aux));
```

Figure 5. KNN pseudo algorithm

The Euclidean distance is defined as the distance between two points, is based on the Pythagorean formula, can be expressed like the square root of the square of the difference between the points:

$$d(x, y) = \sqrt{(x - y)^2}$$

The sort procedure only order the distances vector, from the smaller to the bigger distance. The mode procedure chooses, between the k smaller distances, the mode class point.

2.2.7 *Feed forward Back propagation Neural Network (MLP)*

The Multi-Layer Perceptron (MLP) is a neural network that is usually used to solve classification problems. A common type of MLP is the Feed forward Back propagation Neural Network (FFBNN), whose name is given because the input is forward propagated and the errors are back propagated to correct the neurons weights.

To explain the behavior of the FFBNN, consider the topology described in figure.

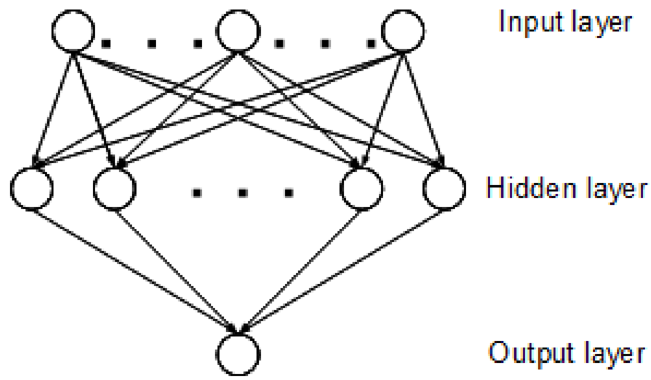

Figure 6. Multi-Layer Perceptron Topology

The input layer propagates the input values to the hidden layer, multiplying the value of each neuron with the respective weight. The final value on the hidden layer neurons are obtained applying the activation function on the summation of all weighted values. The value of the output layer neurons are calculated with the same procedure, however, this value is compared with the desired value from the training set, and the difference (the error) is backpropagated to correct the neuron weights for all layers.

One efficient method to backpropagate the error, minimizing the overall error of the network, was proposed by Broyden-Fletcher-Goldfarb-Shanno (BFGS) as a generic optimization algorithm. It is a quasi-Newton non-linear method for optimization that transforms the error problem on a minimization problem solving the hessian matrix approximation problem [15]

3. THE IMPLEMENTATION OF THE METHODS

Although in the previous sections we presented only a subset of the methods that have been proposed in the literature, we have chosen these methods relying on their accuracy and the sensitivity rates. Thus, the most efficient methods are covered by this study.

Method	**# beats**	**Accuracy (%)**	**Lesser Sensitivity**
Engin	4	98,0	95,3
Güler e Übeyli	4	96,94	95,56
Yu and Chen	6	99,65	99,04
Yu and Chou	6	99,51	98,24
Chen and Yu	6	99,70	98,88
Minhas and Arif	6	99,49	98,92

Table 1. Accuracy and sensitivity rates for the ECG Beat classification methods [3][4][5][7][6][8]

Table 1 shows the accuracy of the methods, and the lowest sensitivity rate between the sensitivities for all beats. The latter statistical measurement is important because it assures that all kinds of ECG beats will be properly classified, avoiding the overestimation that the accuracy measure can cause. Considering these facts, we decided to choose three methods to compare and implement on a mobile device, they are: Chen and Yu method [6], Minhas and Arif method [8] and Yu and Chen method [5].

The selected methods were implemented in the Java Language because it can be easily ported to the mobile phones. With little adaptations, the methods were capable of executing on PCs and on a mobile phone with the Google's Android Operational System, allowing us to analyze the results on a real mobile phone. The mobile phone used for the tests was a Motorola Milestone A853, which has 256Mb RAM memory, an Arm Cortex A8 CPU, which has a frequency of 600MHz and runs the version 2.2 of the Android system.

The following subsections will review the selected methods.

3.1 Yu & Chen method

The method proposed by Yu and Chen [6] is based on a Probabilistic Neural Network (PNN) to classify up to six different ECG beat types – including Normal beat (N), left (L) and right (R) bundle branch block beats, premature ventricular contraction (PVC), atrial premature beat (APB) and paced beat (P). The features are extracted calculating statistical measurements from the DWT coefficients of the raw signal.

3.1.1 Feature Extraction

The DWT is applied on the raw ECG signals to perform the extraction of the features. From the first and second level wavelet (WT) detail coefficients (D1 and D2), and from the second level approximation coefficients (A2) several statistical measurements are calculated, including the variance of the coefficients, the variance of the auto correlation function of the coefficients, the ratio between the maximum and minimum values and the interval between the current beat and the last one.

The authors proposed two feature sets, namely: FS1 and FS2. FS1 includes the variance of the original signal, the variance of D1, D2 and A2, the variance of the autocorrelation of D1, D2 and A2 and the ration between the maximum and minimum values of D1, D2 and D3, for a total of 10 features. FS2 contains all the features of FS1 plus the instantaneous RR interval, since it is an important information for arrhythmias.

After the feature extraction, each feature is normalized, using the hyperbolic tangent sigmoid function to map the values of each feature to the range [-1, +1].

3.1.2 Classification

Yu and Chen use a PNN to perform the classification of the beats. All the training instances must be loaded on the network at the moment of the classification, thus, this method is very costly in terms of memory usage. The fact that the PNN is instance-based also implies that the method is costly in terms of time, because it needs to process the whole training set to classify each testing instance.

For each sample that needs to be calculated, the algorithm needs to compute its PDF relative to each instance from the training set. After this phase, which is made on the Pattern Layer, the Summation layer will sum the results of all neurons from the Pattern Layer, grouping by the class of the neuron and making a weighted average of the value. Finally, the Decision Layer will classify the instance with the class that had the higher average on the Summation Layer.

3.2 Minhas & Arif method

The method developed by Minhas and Arif [8] is similar to the Yu and Chen method, they use the same statistical measurements to build the feature set, but there are some differences on the DWT and on the classifier. The authors also apply a resampling on the records from 360Hz to 250Hz.

3.2.1 Feature Extraction

To perform the feature extraction the *à trous* DWT, using the Quadratic Spline Wavelet as the mother wavelet, is applied to the samples.

The first and second level detail coefficients, and the second level coefficients are selected as a base for the feature set. From them, the same statistical features as the FS2 from the Yu and Chen method are calculated.

The authors also employed the Principal Component Analysis to generate a second feature set, with a reduced dimensionality. However, this procedure also reduced the accuracy of the method, thus, we will focus only on the original feature set, which contains 11 features for each sample.

3.2.2 Classification

Minhas and Arif employed a k-Nearest Neighbor algorithm to classify the testing set. This classifier is extremely easy to implement, since it's depends only on a ordered list of distances between the sample that is being classified and all other samples from the training set. The Euclidean Distance was the selected distance metric, and the *k* parameter was set to 3.

3.3 Chen & Yu method

The method proposed by Chen and Yu uses Higher Order Statistics with the DWT to build a large feature set for each ECG beat, what is classified by a Multilayer Perceptron. The classifier covers the same beat types set as the Yu & Chen method.

3.3.1 Feature Extraction

To be successful on the difficult task of finding a reliable feature set the authors employed Higher Order Statistics allied to the DWT.

Three higher order statistics features are selected to be help to build the feature set, namely: second order cumulant, third order cumulant and fourth order cumulant. The authors use the *à trous* DWT, with the 'sym6' as the mother wavelet, to calculate five levels of coefficients. The 2^{nd}, 3^{rd} and 4^{th} order cumulants are extracted from the detail wavelet coefficients of level 3, 4 and 5. As a result, nine cumulants are calculated for each beat, and from them several features are calculated, including: Cumulant Variance and Normalized Summation for each cumulant, the Number of Zero-Crossings on the cumulants of the level 5 coefficients and the Symmetry of the 3^{rd} and 4^{th} order cumulants. Additionally, three RR-interval related features are defined, including: current RR interval, previous RR interval and the ratio between the current and the previous RR interval. In summary, the feature set is formed by 30 features.

3.3.2 Classification

The classification stage of this method is performed by a MLP, which is a neural network composed by, in this case, three layers of neurons. The first layer only propagates the inputs signal, while the hidden and output layers will combine the inputs with a set of neuron weights and apply an activation function to compute the output. The output is compared to the desired output and an error value is produced. This error is used to adjust the previous neurons weights in a process called Backpropagation training.

The authors empirically defined that the hidden layer is formed by sixty neurons, and the activation function of the network is the hyperbolic tangent sigmoid. The Backpropagation training uses the BFGS method to minimize the error.

4. RESULTS ANALYSIS

To evaluate the behavior of the implemented methods we used 23 records from the MIT/BIH arrhythmia database [9], which consists of a large database of annotated electrocardiograms containing several types of arrhythmias. We used the same dataset to test all the implemented methods in order to avoid overestimating of any of them.

Type	MIT/BIH Record	# Training	# Testing
N	103, 113, 115, 123, 220, 234	600	600
LBBB	109, 111, 207, 214	600	600
RBBB	118, 124, 212, 231	600	600
PVC	119	200	200
	221	150	150
	200, 233	400	400
APB	209	150	150
	222	100	100
	232	600	600
PB	107, 217	600	600
Total		11600	11600

Table 2. The records selected from the MIT/BIH database, grouped by the beat type.

Table 2 shows details of the records from the MIT/BIH database that were used to form the training and testing instance sets. Since neural networks need large training sets to achieve a good performance, we used 11,600 instances on each set.

Our implementation of the Chen and Yu method is not fully functional yet. We need to review the feature extraction phase in order to achieve better results. In our experiments, it achieved at most 89.52%, while the authors reported an accuracy of 99.70% in their work.

Statistical Measurement	Minhas and Arif	Yu and Chen
Accuracy %	99.0	98.67
Specificity %	99.81	99.5
Sensitivity % LBBB	98.29	97.58
Sensitivity % RBBB	99.04	97.79
Sensitivity % PVC	97.91	98.87
Sensitivity % APB	98.00	98.82
Sensitivity % PB	99.67	99.83

Table 3. Comparative results of the Minhas and Arif and Yu and Chen methods

Our experiments for the Minhas and Arif method and for the Yu and Chen methods were successful on the PC and on the mobile phone, achieving an accuracy of 99.0% and 98.67%, respectively.

Due to the heavy computations needed by the classification phase of the PNN and the KNN, the mobile phone needed too much time to classify each beat. For the Yu and Chen method, the PNN classifier needs to compute the PDF for each of the 11,600 neurons from the Pattern Layer, perform the summation and chose the desired class. The selected mobile phone (Motorola Milestone A853) took an average of 623ms to perform this task for each beat. The classifier used by the Minhas and Arif method is the KNN, a classifier that is similar to the PNN, however, instead of finding the PDF for each neuron, it needs to compute the *distance* of the beat to each other beat from the training set and select the class of the *closest* class to be the target class of the beat. The average time taken to classify each beat with this method was 573ms.

On an implementation that targets the monitoring of real patients, the CPU will also be busy acquiring and decoding the patient's signal from the electrocardiographer, detecting the QRS complexes to find the bounds of each beat and extracting the features from the beat. As a result, all these operations, together with the classification phase, can take more than 800ms to be completed for each beat. A traditional heart rate of 75bpm will generate 1 beat each 800ms, in other words, on the selected phone, these methods would not be able to monitor patients with heart rates higher than 75bpm, turning their use impracticable in a real system.

In contrast to them, the Chen and Yu method could be able to classify each beat in a small time. Due the nature of the MLP classifier, that just needs to propagate the inputs through the network to find the target class, it achieved an average of 27ms to classify each beat. However, as we have already mentioned, our implementation of the method still needs to be reviewed in order to achieve the accuracy reported on the author's paper.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a comparison study of ECG beat classification methods destined to detect arrhythmias running in a mobile phone. This study was motivated by the need for a rapid response for the analysis of the exams in some critical situations. Thus, our main objective was to point one or more methods that can successfully classify ECG beats on a mobile phone, with high accuracy fast response.

Relying on the specialized literature, we selected three methods from the authors that reported the higher accuracies together with high sensitivities, namely: Chen and Yu method, Minhas and Arif method and Yu and Chen method.

As we reported in Section 4, our implementation of the Chen and Yu method is not fully completed yet. By now, it achieves at most an accuracy of 89.52%. On the other hand, our implementation of both the Yu and Chen method, and the Minhas and Arif method were successful for both the PC and the mobile phone, achieving accuracies of 98.67% and 99.00%. This performance could be observed in both implementations, for the PC and for the mobile phone. However, these two methods, when running in the mobile phone were too slow to classify the beats. Both of them need more than 500ms. This is indeed a problem, since normal heart beat rates generate more than one beat per second, and fast heart rates will generate more than one beat per 500ms. Thus, both algorithms would not be able to process all beats when running in a cell phone, at least in our implementation.

5.1 Future work

To overcome the performance limitations found in our study so far, we will focus on the implementation of the Chen and Yu method in order to be able to test it in a mobile phone. Since the MultiLayer Perceptron's classification phase is not instance-based, it is not costly in terms of CPU and memory usage. If we fail to fix our implementation of the Chen and Yu method, we'll try to develop a novel ECG beat classification method to run with acceptable performance in a cell phone implementation. The new method will have to combine a fast classification algorithm with a feature set that provides better results.

6. REFERENCES

[1] WHO - World Health Organization. 2004. **Table 3. Estimated deaths per 100,000 population by cause, and Member State. World Health Organization**. Available on: http://www.who.int/entity/healthinfo/statistics/bodgbddeathdalyestimates.xls.

[2] Pan, J.; Tompkins, W. J. 1985. **A Real-Time QRS Detection Algorithm**. IEEE Trans. Biomed. Eng., BME-32 (3):230-236.

[3] Engin, M. 2004. **ECG beat classification using neuro-fuzzy network**. Pattern Recognition Letters, v. 25.

[4] Güler, I.; Übeyli, E. D.2005. **ECG beat classifier designed by combined neural network model**. Pattern Recognition.

[5] Yu, S.-N.; Chen, Y.-H. 2007. **Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network.** Pattern Recognition Letters.

[6] Chen, Y.-H.; Yu, S.-N. 2007. **Subband Feature Based on Higher Order Statistics for ECG Beat Classification.** Proceedings of the 29th Annual International Conference of the IEEE EMBS. Lyon, France.

[7] Yu, S.-N.; Chou, K.-T. 2007. **A switchable scheme for ecg beat classification based on independent component analysis.** Expert Systems with Applications.

[8] Minhas, F.-U.-A. A.; Arif, M. 2008. **Robust electrocardiogram (ECG) beat classification using discrete wavelet transform**. Physiological Measurement.

[9] Mark, R., Moody, G. 1988. **MIT-BIH Arrhythmia Database Directory.** (Cambridge, MA: MIT Press).

[10] Figueredo, M. V. M.; Dias, J. S. 2004. **Mobile Telemedicine System for Home Care and Patient Monitoring**. Proceedings of the 26th Annual International Conference of the IEEE EMBS.

[11] Mallat, S. G. 1989. **A theory for multiresolution signal decomposition: The wavelet representation.** IEEE Transactions on Pattern Analysis and Machine Intelligence

[12] Cohen, A., Kovacevic, J. 1996. **Wavelets: the mathematical background.** Proc. IEEE 84 514–22.

[13] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha and P. Laguna, 2004. **A wavelet-based ECG delineator: evaluation on standard databases.** IEEE Trans. Biomed. Eng.

[14] Osowski, S.; Linh, T. H. 2001. **ECG Beat Recognition Using Fuzzy Hybrid Neural Network**. IEEE Transactions on Biomedical Engineering, v. 48.

[15] Flecher, R. 1980. **Practical methods of optimization.** Section 3.2, p38.