# Robust Feature Extraction from Noisy ECG for Atrial Fibrillation Detection

Octavian Lucian Hasna, Rodica Potolea

Computer Science Department, Technical University of Cluj-Napoca, Romania

## Abstract

*In this article, we propose a novel methodology for detecting 1D periodical time series (single lead Electrocardiograph, ECG) by classification from QRS features extracted from the signal. Our approach starts by dividing the input signal in windows of a number of periods. The boundaries of a period are detected using the state of the art QRS detector. For each period, the input signal is filtered and the S, T, P and Q points are localized. Next we define two sets of classification features, one measuring the distances between the identified points and one considering the differences between the ECG values corresponding to these points. These features are further combined with the features from the remaining periods from the same window. This is the input data for the classification stage, where we use a decision tree. The process is repeated for each window extracted from an ECG and the final result is decided by a voting mechanism.*

## 1. Introduction

Atrial fibrillation (AF) is one of the major causes of stroke, heart failure, sudden death and cardiovascular morbidity in the world. In 2010 more than 33 millions humans were diagnosed with this condition and it is expected that by 2030 this number will increase by 50% [1]. A stroke increases the risk of dementia and there is a consistent evidence that AF is associated with accelerated cognitive decline and higher risk of dementia for adults aged 45-85 years [2].

We propose a new approach to detect AF from a single short lead of ECG by comparing multiple sample rates and different filtering techniques. The designed approach samples the input ECG, filters out the noise and, in the end, extracts new features. The features are dependent on the strategy used for detecting the positions of S, T, P and Q points (Figure 2). The challenges for detecting the positions of the points are related to both the presence of measurement errors and to the noise inside the ECG data.

## 2. Related work

Many algorithms that have been proposed for detecting AF rely on the absence of P-waves or depend on RR segment variability [3]. The main drawback of the methods that use the P-wave (f-wave substitution) is the generation of false positives as the position of P might be confounded with noise. The other category of algorithms are more robust in computing the RR segment variability, but the variability alone can come from different heart arrhythmias.

In [4] the authors claim that RR variability for short ECG can be caused by the presence of ectopic beats, thus increasing false AF detection rate. They proposed to use TQ intervals and to decompose these intervals in wavelets.

For longer ECG the algorithm proposed in [5] computes the differences between consecutive RR segments and then transforms the values in symbols. Consecutive symbols are grouped in words. After that the algorithm computes the entropy between words so as to decide the randomness of the words. If the value is above a threshold that means the ECG recording contains AF.

## 3. Proposed Algorithm

A ECG recording is a time series and can be defined as a sequence of points ($p$) at successive moments in time ($t$). Our time series is one-dimensional and we assume that all the time series are sampled at the same frequency. Using this assumption we can define a time series as:

$$TS\colon \mathbb{N} \rightarrow \mathbb{R}, TS(t) = p_t \tag{1}$$

Our architecture (Figure 1) contains four main modules that interact so as to classify the input time series.

The first module, the Time Series Collector (TSC), samples the continuous time series with the frequency $f_{sampling}$. After the sampling, the module collects consecutive windows of $n$ values. Formally the module can be described as:

$$TSC(D_{raw}) = \{TS_1, TS_2, ...\} \tag{2}$$

where $TS_i$ is the $i^{th}$ time series window and $n = |TS_{w_i}|$.

The second module, the Time Series Pre-Processor (TSP), detects and removes the short-term variations using

ISSN: 2325-887X DOI:10.22489/CinC.2017.164-301

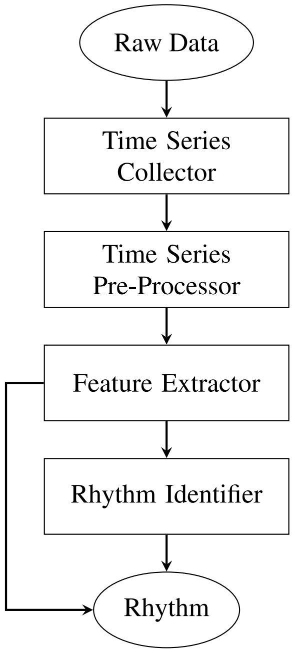

Figure 1. Conceptual Architecture.

a moving average filter (MA):

$$TSP(TS) = TS - MA(TS) \tag{3}$$

For our use case we used the well known simple moving average (SMA) filter by computing the average sum of the points that are at a distance lower than a given radius $r$ from the initial point. The new values represent the transformed time series, as illustrated in the following equation:

$$SMA(t) = \frac{1}{2r+1} \sum_{i=t-r}^{t+r} TS(t) \tag{4}$$

The Feature Extractor (FE) module starts by finding the positions of R points (Figure 2) using the algorithm described in [6]. Instead of using a high band filter, we use a *sombrero hat filter* as suggested in [7]. The module extracts $m$ RR segments (RR = a segment between two consecutive R points). If the module doesn't find $m+1$ R points then it will predict the rhythm as being noise. Otherwise for each RR segment, it searches the positions of the Q and S points using the following formulas:

$$Q_{pos} = \underset{\frac{3*|RR|}{4} \leq i < |RR|}{\arg\min} RR(i) \tag{5}$$

$$S_{pos} = \underset{1 < i \leq \frac{|RR|}{4}}{\arg\min} RR(i) \tag{6}$$

The positions of T and P points is searched inside the segment SQ using the following formulas:

$$T_{pos} = S_{pos} + \underset{1 < i \leq \frac{|SQ|}{2}}{\arg\min} SQ(i) \tag{7}$$

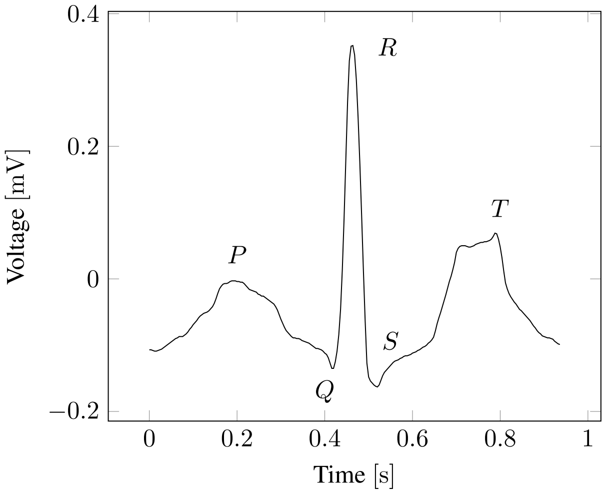

Figure 2. The positions of the P, Q, R, S and T points.

$$P_{pos} = S_{pos} + \underset{\frac{2*|SQ|}{3} \leq i < |SQ|}{\arg\min} SQ(i) \tag{8}$$

Using these positions, the module computes the Ox axis distances between the points $P, Q, R, S, T$ and the Oy axis distances between the pairs $(P, Q)$, $(R, S)$ and $(T, S)$. These distances are computed for each window ($m$ consecutive RR segments) from the input time series. For each group of distances within a window, it computes the standard deviation, minimum, maximum and average values. These values represent the feature set $FS$ for each window from the input time series.

$$FE(TS) = \{\min|RR|, \max|RR|, \delta_{|RR|}, ...\} = FS \tag{9}$$

The last module, Rhythm Identifier (RI) detects the rhythm from the extracted windows. This module has two layers of classifiers. In the first layer, decision trees analyse the FS of each window. In the second layer, a voting mechanism is applied to determine the overall rhythm for the entire input time series.

$$RI(FS_1, FS_2, ...) = rhythm \tag{10}$$

where $FS_i$ represent the feature set for the $i^{th}$ window.

## 4. Results and Evaluation

The dataset provided in [7] contains 8528 ECG recordings classified in 4 classes: normal rhythm, AF rhythm, other rhythm and noisy recordings. They also provide a method (Equation 11) for scoring the classifier. The dataset is heavily unbalanced, as normal case occurs frequently, but the other cases rarely occur.

We use for tuning the classifier a subset of 10% of the dataset that has the same distribution of classes as the full dataset. All the experiments were done in Matlab using the decision tree classifier. For each experiment we apply 10 cross-fold validation and we report the proposed score ($s$) and the F1 measure for AF ($F1_{AF}$).

$$s = \frac{1}{3} * (F1_{normal} + F1_{AF} + F1_{other}) \qquad (11)$$

where

$$F1 = \frac{2 * precision * recall}{precision + recall} \qquad (12)$$

Our first objective is to identify the needed amount of raw data which allows to find a correct solution of the problem. For this we need a trade-off between the used quantity of raw data which gives the minimum acceptable performance and the time to process which achieves the objective of early detecting the rhythm fault. The decision on the sample frequency and on the number of RR segments is taken based on the time performance ratio. We ran several experiments on different number of segments, $m \in \{5, 6, ..., 30\}$ and different sampling frequencies $f_{sampling} \in \{75, 150, 300\}$. We started with $m = 5$ as the minimum number of RR segments used in [8]. For these experiments we use all the features generated by FE. The results are shown in the (Table 1). As we increase $f_{sampling}$ we obtain better performance for less segments. Also we see that $F1_{normal}$ increases while $F1_{AF}$ starts to decrease. A possible explanation is that an AF recording is not entirety AF and it contains sub-sequences of normal recording and as we increase the length of the input window, the normal sub-sequences overcome in length the AF sub-sequences and so AF recordings will be harder to detect. We decide to use for the next experiment $m = 9$ and $f_{sampling} = 300$.

Table 1. Comparison between the score ($s$) and F1 measure for AF ($F1_{AF}$) and theirs standard deviations for different numbers of RR intervals ($m$) and sampling frequencies ($f_{sampling}$).

| | $f_{sampling} = 150$ | | | | $f_{sampling} = 300$ | | | |
|---|---|---|---|---|---|---|---|---|
| m | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ |
| ... | | | | | | | | |
| 7 | 82.1 | 1.7 | 77.2 | 3.6 | 82.9 | 0.9 | 78.0 | 1.3 |
| 8 | 83.6 | 0.8 | 79.1 | 1.7 | 84.1 | 0.8 | 79.7 | 1.3 |
| 9 | 84.9 | 1.3 | 81.0 | 2.8 | 85.0 | 0.9 | 80.3 | 1.7 |
| 10 | 84.4 | 1.0 | 82.6 | 2.0 | 84.4 | 1.3 | 81.4 | 2.5 |
| 11 | 84.2 | 1.0 | 82.3 | 3.0 | 84.0 | 1.2 | 81.6 | 2.8 |
| ... | | | | | | | | |

Our next objective is to find the proper smoothing factor so as to remove most of the noise while keeping all the actual data to allow for correct rhythm identification. We have applied de-noising techniques at 3 layers: raw data, RR segment and SQ segment. In the first layer, we remove the major fluctuation from the input raw data. We use SMA with radius $r_{raw}$ for approximating this fluctuation. Then in the second layer we apply SMA with a smaller radius $r_{RR}$ so as to identify with a better precision the position of the points: Q and S. In the last layer we apply SMA with radius $r_{SQ}$ for the identification of P and T. We run the experiments on a set product of $r_{raw} \in \{15, 16, ..., 45\}$, $r_{RR} \in \{0, 2, ..., 10\}$ and $r_{SQ} \in \{0, 2, ..., 10\}$. The best result was obtained for $r_{raw} = 34$, $r_{RR} = 6$ and $r_{SQ} = 8$.

After that we compare the QRS complex detection algorithm from [6] using a band pass filter and the sombrero hat filter. The results are presented in Table 2. The sombrero hat filter gives an increase of 15% for the $score$ and the results are more stable, the standard deviation is with 69% smaller.

Table 2. Comparison between different QRS complex detection algorithm filters.

| Filters | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ |
|---|---|---|---|---|
| band 5-15 Hz | 72.7 | 1.3 | 64.9 | 2.6 |
| sombrero | 84.0 | 0.9 | 81.1 | 2.8 |

In the next experiments we compare 4 strategies of finding the position of the points: S, T, P and Q. We start by searching the minimum/maximum value in a given interval. Because the time series contains spikes (noise) and we can miss the real position of a point we develop a new strategy that takes into consideration the length of the ascending/descending sub-sequences. If both sub-sequences length are over $ms$ (which depends on RR length) then we found a candidate point. We ran a couple of tests and heuristically determined that the setting $ms = RR * 10\%$ yields the best results. Because of the smoothing when we return to the raw data the candidate points are not exactly in the local minimum/maximum. To solve this problem we decided to adjust the position of the point with at most $ms$ positions to the left or right. The results are presented in Table 3.

Table 3. Comparison between different search strategies.

| Search strategy | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ |
|---|---|---|---|---|
| sub-sequence constraint | 84.5 | 0.6 | 82.0 | 1.7 |
| min/max | 85.0 | 0.8 | 83.0 | 1.4 |
| min/max with adjusting | 84.1 | 1.2 | 81.6 | 2.9 |
| sub-sequence constraint with adjusting | 84.5 | 0.9 | 82.0 | 2.1 |

The feature vectors that we use for our classifier are described in Table 4. The features can be grouped in 5 categories: segment lengths ($RR, PP, PQ, PR, QS, QT, ST$), segment ratio ($QTp = QT/RR$), difference between two consecutive segments ($RR_{diff}$, $PP_{diff}$), ECG values ($P$, $R$ and $T$) and differences between ECG values ($P - Q$, $R - S$ and $T - S$). Because in a window we have multiple segments we used the following features: the minimum, the maximum, the average value and the standard deviation of the previous features. The feature set $FS_5$ was obtained by concatenating the other features sets. The results are shown in the Table 5. The best score is obtained for the feature set $FS_5$ which contains all the feature that we define.

Table 4. Comparison between the structure of the feature sets.

| Features | $FS_1$ | $FS_2$ | $FS_3$ | $FS_4$ | $FS_5$ |
|---|---|---|---|---|---|
| RR, PP, PQ, PR, QS, ST | x | x | x | x | x |
| QTp | x | x |  |  | x |
| STp |  | x | x |  | x |
| $RR_{diff}$, $PP_{diff}$ | x | x | x | x | x |
| P, R, T |  | x |  | x | x |
| P-Q, R-S, T-S | x |  | x |  | x |

Table 5. Comparison between classification results using a given feature set.

| Feature sets | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ |
|---|---|---|---|---|
| $FS_1$ | 84.19 | 0.98 | 81.23 | 2.12 |
| $FS_2$ | 84.65 | 0.83 | 83.24 | 2.00 |
| $FS_3$ | 84.25 | 0.75 | 81.62 | 2.04 |
| $FS_4$ | 84.59 | 0.67 | 83.09 | 1.87 |
| $FS_5$ | 84.91 | 0.93 | 83.34 | 2.17 |

Finally, we tune the decision tree classifier, by comparing different values for the pruning level. The results are shown in the Table 6. We observed that by applying a small pruning level to the decision tree it makes the tree more general and it increases the $score$ and $F1$, but if we apply a greater level of pruning then the score decreases rapidly.

Table 6. Comparison between classification results using a given pruning level.

| Pruning level | s | $\delta_s$ | $F1_{AF}$ | $\delta_{F1}$ |
|---|---|---|---|---|
| 0 | 84.84 | 0.90 | 82.84 | 1.22 |
| 1 | 85.18 | 0.57 | 83.99 | 1.24 |
| 2 | 84.99 | 1.09 | 83.51 | 2.31 |
| 3 | 85.29 | 1.09 | 83.90 | 2.16 |
| 4 | 85.21 | 0.78 | 83.54 | 1.60 |
| 5 | 84.80 | 1.55 | 83.01 | 3.37 |
| ... |  |  |  |  |

Until now we tuned our model using only 10% of the initial dataset. We used the best configuration to test the model on the whole dataset. Also we use all the windows with $m = 9$ RR segments from the input ECG and apply a majority voting between the result generated from each window. We apply 10 fold-cross validation and obtain $s = 85.26\%$, $\delta_s = 1.06\%$, $F1_{AF} = 83.86\%$ and $\delta_{F1} = 2.06\%$.

## 5. Conclusions

This paper presents a methodology for analysing an ECG recording. We propose multiple strategies for extracting important points from ECG. Also our classifier permits multiple consecutive sub-sequences for improving the performance. The first level is composed of decision trees classifiers for each feature set extracted from a sub-sequence, the second layer contains a meta-classifier that performs a majority voting on the results of the first layer.

In this paper we highlight the following contributions: the design of a fast and robust feature extraction based on the important points, the fine tuning of decision tree classifiers in order to achieve best accuracy and, finally, a generic method for time series analysis in order to detect malfunctions. The proposed method was applied to the problem of ECG analysis and specifically to AF classification. As future work, we envision using deep learning techniques for AF rhythm classification.

## References

[1] Kirchhof P, Benussi S, Kotecha D, et. all. ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS. European Heart Journal 2016;.

[2] Singh-Manoux A, Fayosse A, Sabia S, Canonico M, Bobak M, Elbaz A, Kivimäki M, Dugravot A. Atrial fibrillation as a risk factor for cognitive decline and dementia. European Heart Journal 2017;.

[3] Dash S, Chon KH, Lu S, Raeder EA. Automatic Real Time Detection of Atrial Fibrillation. Annals of Biomedical Engineering 2009;.

[4] Garca M, Rdenas J, Alcaraz R, Rieta JJ. Application of the relative wavelet energy to heart rate independent detection of atrial fibrillation. Computer Methods and Programs in Biomedicine 2016;.

[5] Zhou X, Ding H, Ung B, Pickwell-MacPherson E, Zhang Y. Automatic online detection of atrial fibrillation based on symbolic dynamics and Shannon entropy. Biomedical engineering online 2014;.

[6] Pan J, Tompkins WJ. A real-time QRS detection algorithm. IEEE transactions on biomedical engineering 1985;.

[7] Clifford G, Liu C, Moody B, et. all. Af classification from a short single lead ecg recording: The physionet computing in cardiology challenge 2017. In Computing in Cardiology. 2017; .

[8] Rdenas J, Garca M, Alcaraz R, Rieta J. Wavelet Entropy Automatically Detects Episodes of Atrial Fibrillation from Single-Lead Electrocardiograms. Entropy 2015;.

Address for correspondence:

Octavian Lucian Hasna
Computer Science Department, Technical University of Cluj-Napoca, Romania, Bariiu St., 26-28, room M03
octavian.hasna@cs.utcluj.ro