Computers in Biology and Medicine 113 (2019) 103386

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Detection of atrial fibrillation using discrete-state Markov models and Random Forests

Vignesh Kalidas*, Lakshman S. Tamil

Quality of Life Technology Laboratory, Department of Electrical and Computer Engineering, The University of Texas at Dallas, 800 West Campbell Road, Richardson, TX, 75080, United States

ARTICLE INFO

Keywords:
Atrial fibrillation
Electrocardiograms
RR-interval analysis
Discrete-state markov models
Random forests
Stationary wavelet transforms
Autocorrelation

ABSTRACT

In this paper, we present a fully automated technique for robust detection of Atrial Fibrillation (AF) episodes in single-lead electrocardiogram (ECG) signals using discrete-state Markov models and Random Forests. *Methods*: The ECG signal is first preprocessed using Stationary Wavelet Transforms (SWT) for noise suppression, signal quality assessment and subsequent R-peak detection. Discrete-state Markov probabilities modelling transitions between successive RR intervals along with other statistical quantities derived from the RR-interval series constitute the feature set to perform AF classification using Random Forests. Further enhancement in AF detection is achieved by using a post-processing false positive suppression algorithm based on autocorrelation analysis of the RR-interval series. *Datasets*: The AF classifier was trained using the Physionet/Computing in Cardiology 2017 AF Challenge dataset and the Atrial Fibrillation Termination Database (AFTDB). The test datasets consist of the MIT-BIH Atrial Fibrillation Database (AFDB) and the MIT-BIH Arrhythmia Database (MITDB). *Results*: Our algorithms achieved sensitivity, specificity and F-score values of 97.4%, 98.6% and 97.7% respectively on the AFDB dataset and 96.3%, 97.0% and 85.6% respectively on the MITDB dataset. It was also observed that inclusion of the false positive suppression step resulted in a 1.1% increase in specificity and a 4.0% increase in F-score for the MITDB dataset without any decrease in sensitivity. *Conclusion*: The proposed method of AF detection, combining Markov models and Random Forests, achieves high accuracy across multiple databases and demonstrates comparable or superior performance to several other state-of-the-art algorithms.

1. Introduction

The American Heart Association defines atrial fibrillation (AF) as a "supraventricular tachyarrhythmia that is characterized by uncoordinated atrial activation with consequent deterioration of mechanical function" [1,2]. Multiple theories have been put forth to explain the trigger mechanisms for atrial fibrillation. The common thread to all these theories is the existence of multiple focal points of atrial excitation, instead of just the Sino-Atrial node, which is usually termed the heart's natural pacemaker. These multiple focal points are a result of impulse re-entry, which is viewed as a disorder of impulse propagation, rather than impulse formation [3]. This leads to disorganized atrial activity resulting in stochastic changes in heart rates. In the presence of AF, the heart rate fluctuates in a highly irregular manner and the atrial rates can vary anywhere between 240 and 300 beats per minute. AF episodes can be sustained or self-terminating and are usually classified into various categories depending on episode duration [1]. Occurrence of two or more episodes of AF is termed Recurrent AF whereas episodic occurrences that terminate spontaneously within seven days are termed as Paroxysmal AF. When the arrhythmia lasts for more than seven days, it is categorized as Persistent AF and episodes lasting more than a year are labeled as Permanent AF. Although there does exist a significant number of instances where AF is not associated with any other detectable cardiac dysfunction, for about two-thirds of patients, AF is usually associated with other cardiac abnormalities [4]. AF is also known to contribute to a five-fold increase in strokes and can turn life-threatening if not properly diagnosed and treated. This is especially the case in the presence of asymptomatic AF, also known as silent AF. AF episodes often lead to other disorders apart from stroke such as heart failure, hemodynamic impairment and thromboembolic events [1]. AF also impacts cognitive functioning, leading to decreased quality of life and increased healthcare costs [4]. AF is one of the most commonly occurring arrhythmias, affecting about 1–2% of the general population and its occurrence varies with age and gender. The incidence proportion of AF is between 0.12% and 0.16% among individuals younger than 49 years, approximately 4% in individuals in the age group of

* Corresponding author.
E-mail addresses: vignesh.kalidas@utdallas.edu (V. Kalidas), laxman@utdallas.edu (L.S. Tamil).

https://doi.org/10.1016/j.compbiomed.2019.103386
Received 13 March 2019; Received in revised form 5 August 2019; Accepted 7 August 2019
Available online 08 August 2019
0010-4825/

60–70 years and about 10%–17% among those older than 80 years [2,4]. Also, the prevalence of AF is slightly higher in males compared to females (a ratio of 1.2:1) [2], and paroxysmal AF is more prevalent in younger adults. In 2010, 20.9 million men and 12.6 million women were estimated to suffer from AF, with increased prevalence in developed countries. It is expected that 25% of middle–aged adults in the U.S. and Europe will develop AF. By the year 2030, 14–17 million individuals in U.S. are estimated to be affected by AF. The rate of prevalence of AF in U.S. is expected to increase with 120,000 to 215,000 patients being additionally diagnosed every year. This increase in AF prevalence can be attributed to multiple factors including aging population, improved diagnosis techniques and advanced patient monitoring systems that help detect silent and paroxysmal AF episodes better, along with significant strides made in understanding the science behind atrial fibrillation [4].

Automated methods for classification of AF episodes generally rely on the information extracted from electrocardiogram (ECG) signals. The absence of periodically occurring P-waves or presence of fibrillatory f-waves in ECG (seen as undulations of the isoelectric baseline) coupled with irregular heart rate fluctuations are primary indicators of AF. Despite the significant progress made in understanding the factors contributing to occurrence of atrial fibrillation episodes, development of automated techniques to detect AF episodes remains far from achieving satisfactory results due to several contributing factors. First, there are several other arrhythmias that typically mimic AF in terms of their manifestation on the ECG as well as possessing similarities in terms of spectral content, extent of heart rate variability, etc. Secondly, the presence of external noise, especially owing to electrode/patient movements, hinders the performance of AF classifiers severely, giving rise to increased misclassification rates; this gains further significance in the context of today's wearable sensors. Currently, there is a growing interest in the use of wearable sensors for ECG recording and subsequent arrhythmia monitoring [37,40]. The modern-day wearable sensors offer cost-effective and hassle-free solutions for vital monitoring purposes, but are extremely susceptible to noise interference and are highly sensitive to motion artifacts. These afore-mentioned factors necessitate the need to develop an AF classifier that is highly robust to noise while being capable of accurately identifying AF rhythms, especially in the presence of other similar arrhythmias. These challenges serve as the primary motivation for our work presented here.

In this paper, we propose an RR-interval analysis technique to detect AF episodes from single-lead ECG signals. The discrete-state transitions in the RR-interval time series are modeled as an eight-state Markov process. The use of Markov models offers the advantage that sequential pattern changes in heart rates can be effectively captured, thus aiding in better distinction between AF and other arrhythmias with prominent heart rate variations. These Markov probabilities, along with other statistical parameters that help quantify randomness in RR-interval transitions, are input to a Random Forests based AF classifier for initial AF classification. Further post-processing steps are added to reduce potential false positives and increase the overall accuracy in detecting AF episodes.

2. Previous work

Several algorithms have been proposed in literature for detection of atrial fibrillation episodes. Automated AF detection techniques usually involve analyzing ECG signals to estimate the degree of irregularity in the RR-interval sequence, which is a characteristic feature of AF (other than in the presence of paced heart rates). In that regard, the authors of [6] propose use of linear and non-linear heart rate dynamics to distinguish AF rhythms from normal sinus rhythms and sinus rhythms with frequent ectopic beats. In another RR-interval based approach [7], the sequence of RR intervals during AF is modeled as a three-state Markov process, namely short, regular and long. The RR-intervals were assigned to one of these states based on the relation of the current RR-interval with the exponentially averaged mean RR-interval. Histogram analysis on distributions of RR intervals and first difference of RR-intervals are the basis of the approach adopted in [8], where reference distributions were created for AF and non-AF classes and at test time, the RR-interval distributions were matched with these reference distributions using Kolmogorov-Smirnov test for goodness-of-fit [32]. Other ventricular rate-based methods include studying Lorenz plots [9,10] of RR intervals, where successive RR-intervals are scatter-plotted against each other. The scatter-plots are more spread out in the presence of AF whereas they stay in one or more tight clusters in its absence. AF classification methods based purely on evaluation of RR-interval entropy have also been found to perform reasonably well in the past [38–40]. These entropy-based approaches rely on the template-matching (vector similarity) property of RR-segments in an ECG epoch, where lack of template similarity under a given tolerance indicates presence of AF rhythms [38–40]. The advantage of using RR-interval based techniques is that the determination of R-peak locations and subsequent computation of heart rate metrics are robust to most noise sources except electrode motion artifacts. On the other end of the spectrum, heart-rate-independent approaches do not include any beat detection procedures but they utilize information from atrial activity and such techniques include analyzing P-wave absence [11], average F-wave activity [12], wavelet features [14,22] and implementing echo state neural networks for QRST cancellation [13]. Use of P-wave and f-wave information can help prevent false detections in the presence of arrhythmias with continuously varying heart rates. Both heart-rate based methods and heart-rate-independent techniques have their limitations. The ECG lead orientation often results in mitigated amplitudes of regular and normal P-waves. The distinction between f-waves and isoelectric baseline noise is rendered difficult by frequent overlap of their spectral bandwidths, resulting in poor atrial activity analysis. On the other hand, examination of irregularities in the ventricular heart rate is complicated in the presence of other arrhythmias with significant heart rate variations such as frequently occurring premature ventricular contractions (PVCs) and non-arrhythmic factors such as electrode motion artifacts that often mimic QRS complex morphologies. These factors can render QRS detection algorithms inaccurate at times.

Over the past decade or so, with rapid advances in data collection, storage and GPU capabilities, multiple techniques using deep learning as well as ensemble learning methods have been proposed for AF classification. Most notably, the annual Physionet/Computing in Cardiology Challenge for 2017 [15] focused on identifying AF episodes in short term ECG recordings. In fact, a majority of the data for training our Random Forests based AF classifier is obtained from the challenge's training database (see Section 3.1). The top-ranked results of this challenge and their respective methods are described in [27–30]. Methods described in [27,28] leverage deep learning techniques to develop an AF classifier capable of performing reliably in the presence of other arrhythmias. The algorithm discussed in [29] describes an approach based on training a RandomForests classifier on heart rate dependent features for AF classification. Cascaded binary classifiers using the Adaboost learning algorithm [33] form the basis of the work described in [30]. Apart from the algorithms proposed for the challenge, several other contributions in the field of automated AF classification have also adopted deep learning-based approaches lately. Recently, the authors of [31] proposed a convolutional neural network (CNN) based deep learning approach to distinguish different arrhythmias. Their model was trained using privately collected ECG data and its performance was evaluated on the MITDB database. In another deep learning related approach [26], stationary wavelet transforms and CNN architectures were combined to train a classifier capable of identifying AF episodes in very short ECG segments (~5s). Although deep learning techniques offer an attractive alternative to hand-computed feature extraction, limited availability of labeled AF datasets acts as a major bottleneck for training robust deep learning models for AF classification.

In comparison, our approach towards AF classification is characterized primarily by implementing a combination of Markov models and Random Forests classifiers to perform noise assessment and RR-interval based AF classification. The inconsistency in the detection of P-waves and delineation of fibrillatory f-waves from noisy ECG recordings contributed to omitting atrial activity analysis in our approach. The use of a nine-state Markov matrix for noise analysis and an eight-state Markov matrix for AF classification, complemented by Random-Forest based feature-learning ensures that our algorithms can effectively distinguish AF from other arrhythmias as well as robustly identifying noise interference. A post-processing false positive subroutine is added to further enhance our algorithm's accuracy. As can be observed in Section 5.3, our algorithm achieves comparable or better performance than other state-of-the-art AF classification algorithms on the standard databases. Additionally, our algorithms demonstrate consistently accurate performance across multiple different databases, as explained in Sections 4 and 5. Our methods, along with results and discussion, are described in the following sections.

3. Methods and materials

Fig. 1 shows the flow diagram of our AF classification algorithm. Our algorithm divides the ECG signal into 60s long non-overlapping epochs to analyze the presence of AF. Prior to deriving features or performing AF detection, signal quality analysis is performed on the entire ECG signal to identify segments that are too noisy to render any useful beat detection. Such segments are ignored for AF analysis. The next step involves determining the beat locations along with computing QRS onset and offset indices. This beat information is used for discretizing the RR-interval sequence into eight different states and computing its corresponding discrete-state Markov matrix. The elements of this Markov matrix along with other features obtained from RR-interval analysis are fed as input to a Random Forests based AF classifier. Once an epoch is classified as exhibiting AF, a reconfirmation of this classification is made by our false positive suppression algorithm that uses autocorrelation values of the RR-interval state sequence to suppress potential false positives. These could usually arise in the presence of several different arrhythmias, each with a differing heart rate variation pattern, all occurring in one single 60s epoch. One such example is presented later in Fig. 5.

3.1. AF datasets

The Random Forests model used for AF classification was trained using features derived from datasets available in the Physionet database [17]. The first one is the Physionet/Computing in Cardiology Challenge 2017 training dataset [15] which consists of 758 short single-lead ECG recordings of varying duration (~30s–60s) exhibiting atrial fibrillation and 2415 ECG records of similar duration having arrhythmia other than AF. All the signals are sampled at 300 Hz. The challenge data was augmented with AF recordings from the AF Termination Database (AFTDB). This database has 80 1-min long single-lead ECG recordings, each sampled at 128Hz [16]. We divided each of these records into two 30s long segments, resulting in a total of 160 additional AF training examples. This augmented dataset was then used to derive RR-interval based features for AF classification. For testing our algorithm, we used ECG records from the MIT-BIH Atrial Fibrillation Database (AFDB) [7] and the MIT-BIH Arrhythmia Database (MITDB) [18]. The databases used for testing are completely different from the ones used for training. In fact, in our approach, both the AFDB and the MITDB databases are used as test databases whereas most state-of-the-art methods rely on the AFDB database for training/developing their AF classifier models. More information regarding the test data and evaluation of our algorithm is provided in Section 4.

3.2. Random Forests

Random Forests is a machine learning algorithm that consists of growing an ensemble of binary decision trees for performing regression/classification. Introduced in [34], Random Forests algorithm is described by the authors as a modification of bagging which creates a forest of de-correlated trees and averages their predictions. Decision tree models generally suffer from high-variance and thus averaging out predictions from multiple decision trees, each with the same variance,

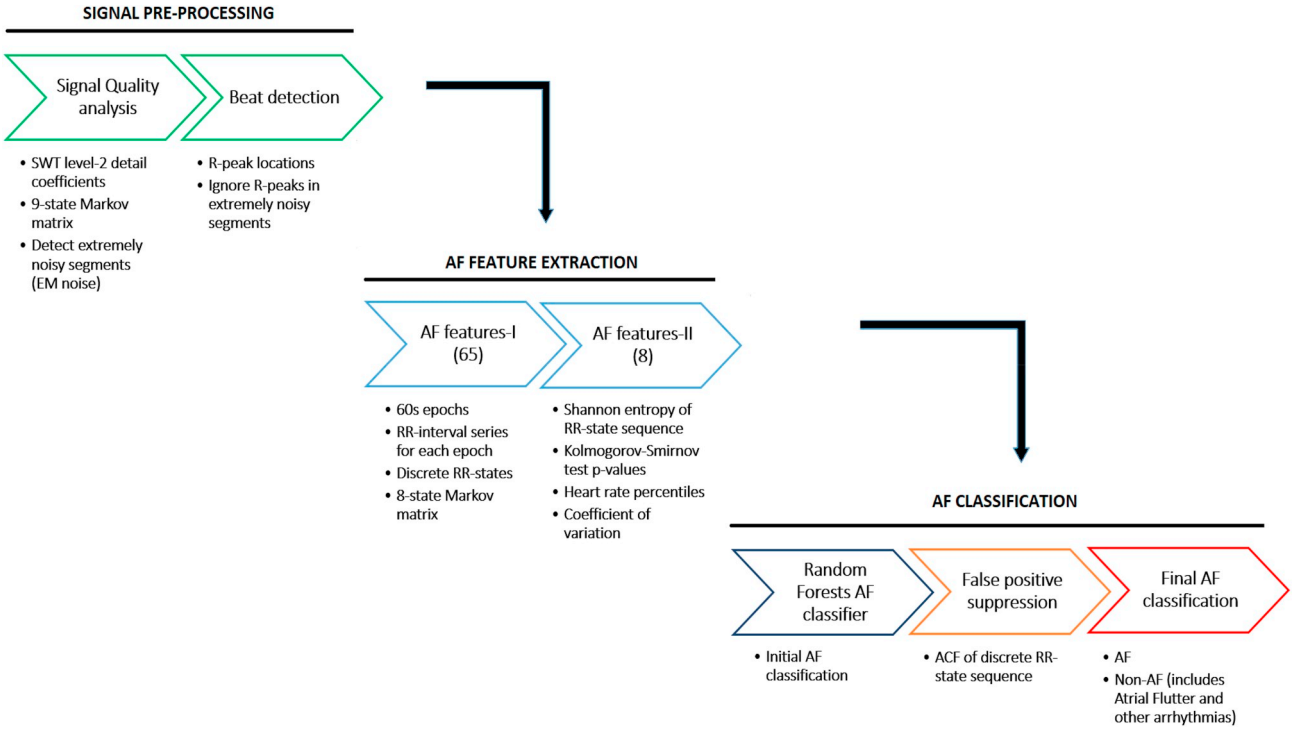

Fig. 1. Flow diagram for the AF classification algorithm.

leads to a decrease in overall variance [36]. The process of growing these decision trees is based on bagging (bootstrap aggregation), which is a technique for reducing the variance of an estimated prediction function. Here, bagging primarily involves creating several data subsets from the training data by random subsampling with replacement. Each of these data subsets is then used to train a decision tree model separately. Additionally, for each tree, only a random subset of predictors/features is used for creating the decision model. For each tree thus grown, a predictor is chosen to make a decision split at a particular node. This is usually done based on either the information gain/entropy or the Gini impurity that selects the best predictor for that particular node [41]. This process is repeated at each node in a tree and subsequently multiple trees are trained. Final decisions are made based on averaging (for regression) or majority voting (for classification) of individual tree predictions. Since data subsampling and predictor subsampling for each tree is carried out randomly on a forest of trees, this algorithm is termed Random Forests. The inherent randomness in the training data (data subsampling and predictor subsets) results in reduced overfitting [36] and places a limit on the generalization error [34]. Random Forest models are often viewed as a competitive alternative to boosting techniques such as Adaboost while being significantly more robust to noise and easier to train. More information about Random Forests can be found in [34,36,41].

In our algorithms, two separate Random Forests based classifiers are used – one for signal quality analysis and another one for AF classification. The steps involved in deriving the features for training each of these Random Forests classifiers are explained in the subsequent sections.

3.3. Signal quality analysis

Prior to AF analysis, signal quality analysis is performed to reduce erroneous AF classifications. The ECG signal can be corrupted by different sources of noise such as powerline interference, baseline wander, muscle noise, electrode motion artifacts, etc. Powerline interference, muscle noise and baseline wander generally occupy frequency bands that do not overlap with QRS complex frequencies and hence can be easily removed without compromising useful QRS information. On the other hand, electrode motion (EM) artifacts need more attention as they mimic QRS complexes in their morphology and usually occupy the same spectral bandwidth as QRS complexes. Hence exaggerated presence of EM artifacts could potentially hinder accurate beat detection. To identify segments exhibiting such significant EM noise, our algorithm segments the entire input signal into 2-s long segments and from each of these segments, a feature vector is created to be fed as input to a Random Forests noise classifier trained to detect EM noise segments. The steps involved in deriving these features are explained below:

- The input signal is normalized by performing mean subtraction and amplitude scaling as follows:

$$x' = x - \left(\frac{1}{L}\sum_{i=1}^{L} x(i)\right) \tag{1}$$

$$y = x'/(\max(|x'|)) \tag{2}$$

where, x is the input signal, L is the number of samples in the signal and y is the normalized output.

- The normalized signal is resampled at 80 Hz. Since useful ECG information is present in the [0.05 Hz–40 Hz] frequency band, resampling the signal at 80 Hz preserves this information while reducing the computation time needed for further analysis.
- Two-level Stationary Wavelet Transform (SWT) is applied to the resampled signal. SWT is a variant of discrete wavelet transform where there is dyadic compression in the frequency domain without any downsampling in the time domain. The wavelet coefficients therefore have the same length (duration) as that of the input at each scale which helps reduce resolution errors at higher scales (lower frequencies). The SWT step can be viewed as a means to compute the effective band-pass for the signal at each scale. In our work, the Daubechies3 (Db3) mother wavelet is used for computing the SWT coefficients. Sampling at 80 Hz allows us to reconstruct signal components up to 40 Hz in accordance with the Nyquist-Shannon sampling theorem. Since there is dyadic compression in the frequency domain, scale-1 coefficients correspond to signal components in the [20 Hz, 40 Hz] band and scale-2 coefficients correspond to signal components in the [10 Hz, 20 Hz] band. Since most of the QRS energy is concentrated in the 10–20 Hz range [25], our algorithm uses the scale-2 detail coefficients for EM noise assessment. More information regarding SWT and their applications can be found in [24].
- All the peak amplitudes from scale-2 coefficients are normalized in the range [-1, 1] and discretized into different states according to Table 1. This sequence of peak amplitude states is labeled pk_d. For a fairly clean ECG segment, most of these peaks must either lie in state 0 (non-QRS states – corresponding to amplitude range of $+0.25$ units) or transition back to state 0 from other (QRS) states within a couple of steps. In the presence of EM noise, the probability of these peaks rapidly transitioning back to state 0 is greatly minimized. Fig. 2 illustrates this behavior. The left plot shows the state-to-state transition probabilities for the SWT peaks belonging to extremely noisy records and the right plot displays the same for fairly clean ECG segments. It can be observed that for the latter, a majority of the transitions occur within state 0. This is evidenced by a prominent peak with a narrower base for the transition probability values near state 0. In contrast, for noisy records, the transitions are more spread out in the middle amongst multiple peak states. In the latter case, the approximate base of the central peak is much wider compared to noise-free records, corresponding to more randomness is the state-to-state transitions. This indicates higher entropy in the wavelet coefficients, and hence representative of a noisy signal environment.
- Using Table 1, we compute a 9 × 9 discrete-state Markov matrix **A** for the sequence *pk_d* where element $\mathbf{A_{ij}}$ equals the conditional probability of reaching state j given the current state is i, i.e.,

$$\forall\, i, \forall\, j \in \{-4, \ldots, 4\},$$

$$A_{ij} = P(next_{state} = j \mid current_{state} = i) \tag{3a}$$

$$\sum_{j=-4}^{4} A_{ij} = 1\,, \qquad -4 \le i \le 4 \tag{3b}$$

- The 81 elements of Markov matrix **A** along with the root mean square value of the normalized peak amplitudes form the final feature vector (82 features in total) for each 2s window.

Using the above steps, a [W x 82] matrix (where W is the number of non-overlapping 2s long ECG windows) is created and fed to a Random Forests based noise classifier. This classifier was trained with EM noise signals obtained from the Noise Stress Test Database [23] available in the Physionet database [17] using the same 82 features as above. The noise classifier assigns noise scores (prediction probabilities) in the range [0, 1] for each 2-s ECG window. ECG signal windows that are assigned noise scores greater than 0.9 are considered extremely noise and are omitted from further analysis by our algorithm. The pseudo-code for our signal quality analysis routine is presented below.

Pseudocode: Signal Quality Analysis

Input: *fs:* 80Hz, *ecg:* ECG segment resampled to 80Hz, *L:* resampled signal length
Output: Noise annotations *ns_ann*
/* **ns_ann**: 2-column matrix containing start and end timestamps (in samples) of extremely noisy epochs */
begin
  Compute 2-level Stationary Wavelet Transform on the input *ecg*
  Extract scale-2 detail coefficients *swt_det2*
  W := L/(2*fs) /* Number of 2s epochs in the input signal */
  ns_input := [ ] /* Feature matrix for signal quality analysis */
  ns_ann := [ ]
  for i=1 **to** W **do**
    st := ((i-1)*2*fs) + 1; /* epoch beginning */
    fi := i*2*fs; /* epoch ending */
    ns_timestamp[i,1:2] := [st/fs, fi/fs] /* Store start and end timestamps of i^{th} epoch */
    Extract *swt_det2* peaks corresponding to the interval [st … fi]
    Normalize above peaks in the range [-1 … 1]
    Discretize normalized peaks as per Table 1 and store them in *pk_d*
    Compute nine-state Markov matrix for elements of *pk_d* /* 81 features */
    Compute RMS value of normalized peaks /* 1 feature */
    Use these 82 features as the i^{th} row of *ns_input*
  end
  Input *ns_input* to a Random Forests based noise classifier
  Identify epochs where the classifier's output score for noise class exceeds 0.9 so as to suppress further analysis in them
End

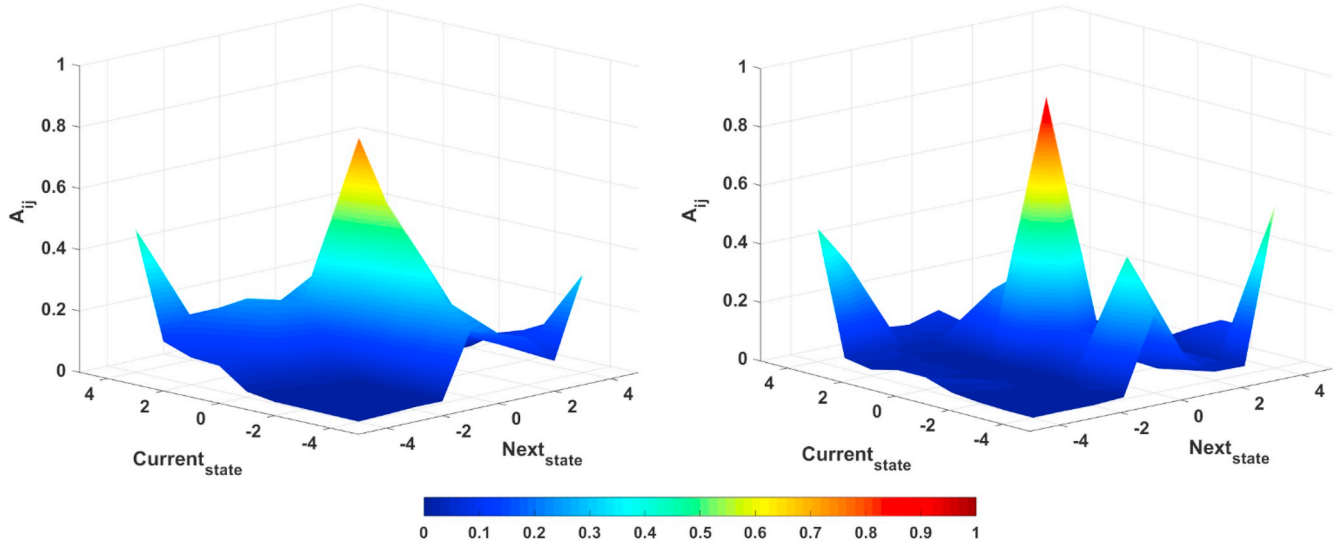

Fig. 2. (Left) Discrete-state transition probability values A_{ij} for noisy records in the training data. (Right) Discrete-state transition probability values A_{ij} for noise-free records in the training data.

Table 1
State assignment table for normalized SWT level-2 detail coefficient peaks.

NORMALIZED PEAK VALUE	STATE (*pk_d*)
[-1.0, −0.8)	−4
[-0.8, −0.6)	−3
[-0.6, −0.4)	−2
[-0.4, −0.25)	−1
[-0.25, 0.25)	0
[0.25, 0.4)	1
[0.4, 0.6)	2
[0.6, 0.8)	3
[0.8, 1.0]	4

3.4. Beat detection

After implementing signal quality analysis as above, our algorithm performs beat segmentation by determining R-peak, QRS onset and QRS offset locations using the methodology described in [5] with a few modifications as explained below.

- The moving window averaged (MWA) signal is shifted to the left by 0.075s (i.e. one-half of the moving window size of 0.15s - see Section III-C in [5]). This helped in better alignment of MWA peaks with the true R-peak locations in the ECG signal.
- For each peak location obtained from the MWA signal (see steps 2–7 under Section III-D in [5]), the index of the first sample on the left of the peak whose MWA amplitude falls below 0.1 units is labeled as its

corresponding QRS onset. Similarly, the index of the first sample on the right of the peak whose MWA amplitude falls below 0.1 units is labeled as QRS offset location for that particular beat.
- The sample index corresponding to the midpoint between the QRS onset and QRS offset for a particular beat is determined as the R-peak location for that beat. This manner of assigning R-peak locations ensured more stability while computing RR-interval based features for AF classification.

The R-peak indices, labeled *r_pk*, are used to create RR-interval analysis based features vectors to detect AF presence. The derivation of these features is described in the next subsection.

3.5. Features for initial AF classification

First, the input signal is segmented into 60s epochs. For each 60s epoch, the corresponding R-peaks stored in *r_pk*, are used to create feature vectors for training a Random Forests classifier for AF detection using the steps described below.

- The RR-interval time-series is computed as follows:

$$rr_{n-1} = r_n - r_{n-1}, \qquad 2 \le n \le N \tag{4}$$

where, N is the total number of R-peaks in the epoch, r_n is index of nth R-peak in the epoch and *rr* is the RR-interval series.

- The successive difference of RR-intervals (ΔRR) in percentage values, '*rr_per*', is calculated as below:

$$\forall\ n \in \{2,\ldots,N-1\},$$

$$rr\_per_{n-1} = 100*(rr_n - rr_{n-1})/rr_{n-1} \tag{5}$$

- The *rr_per* series is discretized into 8 different states as per Table 2. This discretized series is labeled as *rr_d*. The *rr_per* boundaries for each state were determined empirically.

3.5.1. AF features I - Markov matrix features

The 8 × 8 discrete-state Markov matrix **B** for the *rr_d* state sequence is computed in a manner similar to (3a) and (3b), as follows:

$$\forall\ i, \forall\ j\ \in \{1,\ldots,8\},$$

$$B_{ij} = P(next_{state} = j \mid current_{state} = i) \tag{6a}$$

$$\sum_{j=1}^{8} B_{ij} = 1, \qquad 1 \le i \le 8 \tag{6b}$$

In the presence of AF, the Markov matrix **B** is more densely populated than in the presence of other arrhythmia with similar magnitude of heart rate variation, such as ventricular bigeminy/trigeminy, etc. This can be attributed to the fact that the variations between successive RR-interval values are random in AF episodes and do not follow any specific pattern. In the absence of irregular and random heart rate variations, the matrix **B** is sparsely populated. Fig. 3 shows an ECG segment completely dominated by AF rhythm. The corresponding Markov matrix (bottom plot) has values that are spread about the center of the matrix in a largely random fashion, indicating lack of discernible patterns in the heart rate variations. On the other hand, Fig. 4 represents a 60s long ECG epoch containing episodes of ventricular trigeminy, ventricular bigeminy and normal sinus rhythm coupled with a few ectopic beats. Although there is significant amount of heart rate variation here as well, evidenced by the *rr_per* series (in the middle subplot), the corresponding Markov matrix in the bottom plot looks vastly different. Here, most of the transition probability values are concentrated near the left-bottom and top-right corners of the matrix indicating the presence of specific sequential patterns in the RR-interval series. The elements of this 8 × 8 discrete-state Markov matrix form the first 64 features for AF classification.

The other feature obtained from the Markov matrix **B** is the total number of zero-valued elements present in it. This quantity reflects a measure of sparsity of the Markov matrix and in the presence of AF, this value is expected to be remain minimal. In the past, a 3 × 3 Markov matrix has been used in [7] for performing AF classification, however our computation and subsequent use of Markov matrix probabilities involves different parameters.

3.5.2. AF features II - other RR-interval features

Apart from the features derived from the Markov matrix as described above, we also compute eight other RR-interval based features to help distinguish AF from other similar rhythms effectively. They are as follows:

- Entropy of the *rr_d* state sequence, which indicates the extent of randomness in the RR-interval transitions. This entropy measure is computed using natural logarithm as follows:

$$e_{ss} = (-1)^* \sum_{k=1}^{8} [P(s_k)^* \log_e P(s_k)] \tag{7}$$

where, e_{ss} is entropy of the *rr_d* state sequence and $P(s_k)$ is the probability of occurrence of kth state in the rr_d state sequence.

- The value of hypothesis H and the p-value of Kolmogorov-Smirnov test [32] for normality, applied to the *rr_per* series. The null hypothesis (H = 0) is that the *rr_per* series comes from a normal distribution which is the case for AF. The test is conducted at significance level of 5% and the p-values when the AF is present is generally much greater than 0.05 [8]. The value of the alternate hypothesis (non-AF) is H = 1.
- Coefficient of variation of the RR-interval series, computed as per the equation below:

$$coeff_{var} = \frac{\sigma_{RR}}{\mu_{RR}} * 100 \tag{8}$$

where, σ_{RR} is the standard deviation of RR-interval series without the smallest and largest intervals. Similarly, μ_{RR} is the mean of the RR series without the smallest and largest intervals [8].

- The last four features are heart-rate quartiles derived from the RR-interval series. They include the (i) second quartile, (ii) the third quartile, (iii) difference between second and first quartiles, and (iv) the difference between third and second quartiles.
- All the above features together form a 73-element feature vector for each epoch. Subsequently, an [M x 73] matrix is created where M is the total number of 60s epochs in the ECG signal. This feature matrix is fed as input to a Random Forests based classifier that classifies each row of the feature matrix (i.e. each epoch) as either exhibiting AF presence or not. This initial classification is followed by a false positive suppression stage as described below.

3.6. False positives suppression

For each epoch classified as exhibiting AF by the above Random Forests-based AF classification model, our algorithm tests for false positives (FP) to reduce instances of other arrhythmias being misclassified as AF. This majorly comprises ventricular arrhythmias such as bigeminy/trigeminy, along with sinus rhythms with frequent ectopic beats, all of which can, at times, exhibit heart rate variability very similar to AF. Hence it is possible for the AF classification model to misclassify them as atrial fibrillation. To reduce such false positives, our algorithm first computes the autocorrelation function (ACF) values of the *rr_d* state sequence for each epoch that is classified as having AF. If the ACF for any such epoch decreases below −75% (negatively correlated) at lag 1 or exceeds +50% at lag 2, then that epoch is reclassified as non-AF (see Fig. 5). The reason behind adopting this approach is that if an

Table 2
State assignment table for ΔRR transitions.

ΔRR VALUES (rr_per, %)	STATE (rr_d)
(-∞, −50)	1
[-50, −30)	2
[-30, −15)	3
[-15, +15)	4
[+15, +45)	5
[+45, +75)	6
[+75, +100)	7
[+100, +∞)	8

epoch is dominated by the earlier mentioned ventricular arrhythmias, then the *rr_d* sequence has state values that follow a specific pattern repeatedly, unlike in the presence of AF where the state transitions are highly random. This repeated behavior can be well captured by the ACF metric and hence can help correctly classify the corresponding epoch. Adding this additional false positive suppression step helped further reduce false positives as can be evidenced by the increase in F-scores attained on the test data (see Table 3). The statistical significance of using this FP suppression step is explained in Section 5.2. This concludes the methodology description section. A Pseudocode summarizing our AF detection algorithm is presented below.

Pseudocode: AF classification

Input: ECG segment *ecg*, sampling frequency *Fs*, signal length *L*
Output: AF annotations *AF_ann*
/* **AF_ann**: 3-column matrix whose 1st and 2nd columns contain start and end timestamps of each epoch (in seconds), 3rd column contains corresponding AF annotations - vector with 1s for AF and 0s for non-AF epochs */
begin
Perform signal quality analysis on *ecg* as described in Section 3.3
Perform beat detection as described in Section 3.4
M := L/(60*Fs) /* Number of 60s epochs in the input signal */
AF_input := [ ] /* AF feature matrix input*/
for i=1 **to** M **do**
st := ((i-1)*60*Fs) + 1; /* epoch beginning */
fi := i*60*Fs; /* epoch ending */
Extract R-peaks whose locations fall in the interval [st … fi]
Compute RR-intervals time series *rr* using Equation (4)
Compute *rr_per* using Equation (5) and discretize as per Table 2 and store them in *rr_d*
Compute Markov matrix features as described in Section 3.5.1 /* 65 features */
Compute other RR-interval features as described in Section 3.5.2 /* 8 features */
Use these 73 features as the i^{th} row of *AF_input*
AF_ann[i,1:2] := [st/Fs, fi/Fs] /* Store start and end timestamps of i^{th} epoch */
end
Input *AF_input* to a Random Forests based AF classifier
Append the Random Forests AF classifier output as the third column of *AF_ann*
Perform False Positive suppression as described in Section 3.6
Update *AF_ann* after removing potential false positives
end

4. Results

We tested the performance of our algorithm on the MIT-BIH Atrial Fibrillation Database (AFDB) and the MIT-BIH Arrhythmia Database (MITDB). The AFDB database [7] has 23 two-lead ECG records, each sampled at 250 Hz and having a duration of approximately 10 h (except record 06453 which has a duration of approximately 9 h). The MITDB database [18] contains 48 two-lead ECG records, each sampled at 360 Hz and 30 min long. These records are divided into two categories - the 100-series and the 200-series. The 100-series records do not contain any annotated episodes of AF rhythms (hence, the N/As in Table 3). On the other hand, the 200-series contains eight records that have substantial AF presence. We use signals from the first ECG lead in both AFDB and MITDB databases to evaluate the performance of our algorithm. Atrial Flutter and Junctional arrhythmias in the AFDB records are grouped as non-AF in our evaluation. Since we process 60s epochs in our algorithm for AF detection, it would be unfair to compare our results with beat-to-beat annotations. Hence, instead we converted the true beat-to-beat annotations into *60s-epoch* annotations. A 60s epoch was annotated as having AF only if at least 50% of the beats in that epoch were originally annotated as AF beats. Using this approach, 136 AF epochs and 1304 non-AF epochs were obtained for the MITDB database. Similarly, 5528 AF epochs and 8226 non-AF epochs were obtained for the AFDB database.

The evaluation results of our algorithm are summarized in terms of sensitivity, specificity and F-score metrics in Table 3. A sensitivity of 97.4%, specificity of 98.6% and an F-Score of 97.7% was obtained on the AFDB database. Similarly, a sensitivity of 96.3%, a specificity of 97.0% and an F-Score of 85.6% was obtained on the MITDB database. It can be seen from Table 3 that use of a false positive suppression step results in enhanced AF classification performance that is more noticeable on the MITDB records. Specifically, for the MITDB-200 records, there is a 2.4% increase in specificity and a 4.1% increase in F-score with the use of the FP suppression step. Overall, the FP suppression stage results in a 1.1% increase in specificity and a 4.0% increase in F-score on all the MITDB records together without any decrease in sensitivity. On the other hand, relatively lower F-scores on the MITDB database compared to the AFDB database can be attributed to the high imbalance in the proportion of non-AF and AF annotations (nearly 10:1). The equations for computing the above three evaluation metrics are as follows:

$$Se = \frac{TP}{TP + FN} \tag{9a}$$

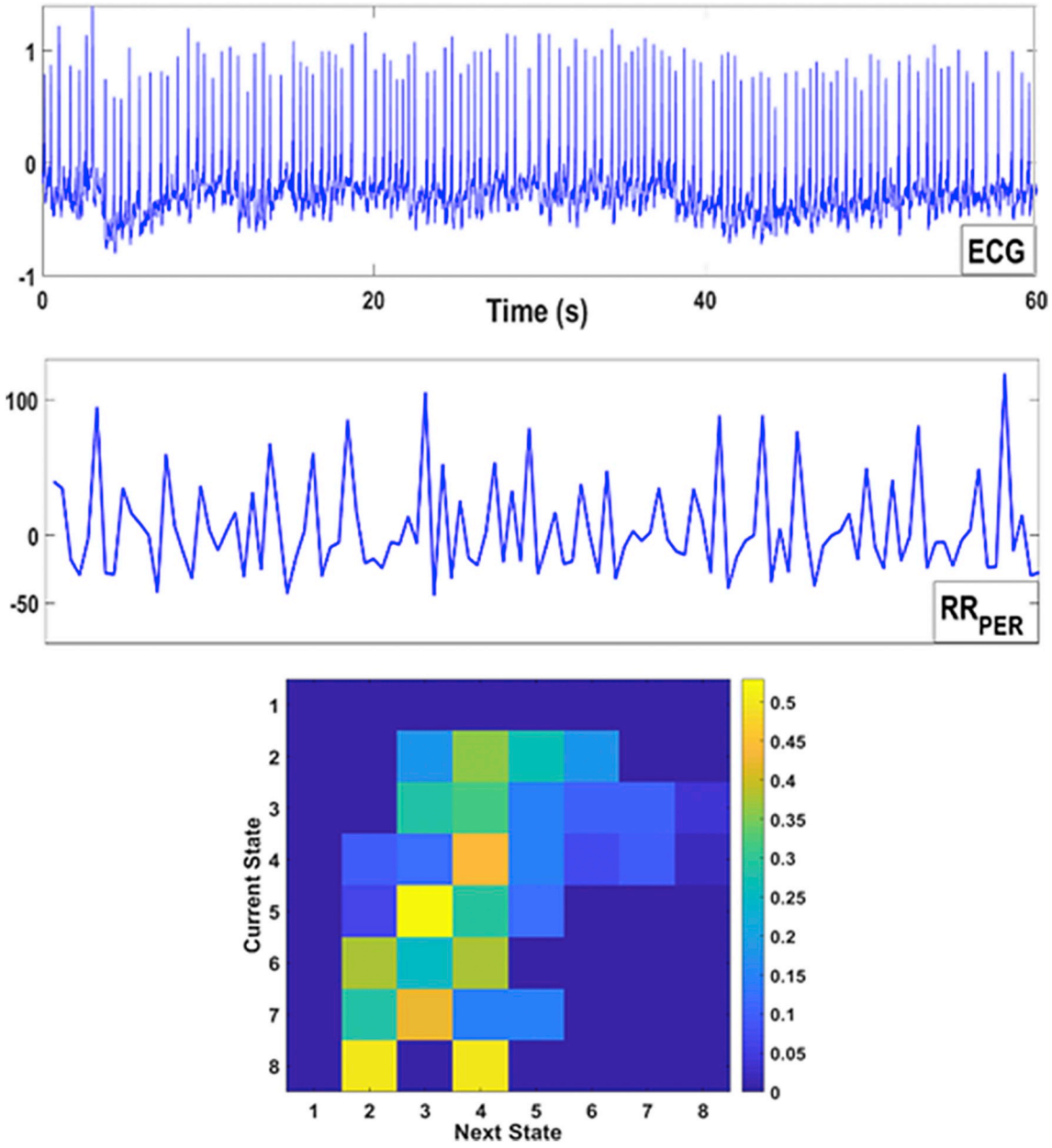

Fig. 3. True positive detection of an AF episode. (Top) 60s ECG epoch with complete AF presence. (Middle) The rr_per series. (Bottom) Corresponding discrete-state probability transition matrix. (Epoch belongs to record 04043 from the AFDB database).

$$Sp = \frac{TN}{TN + FP} \tag{9b}$$

$$Fsc = \frac{2*TP}{(2*TP) + FP + FN} \tag{9c}$$

where, *Se* refers to sensitivity, *Sp* refers to specificity and *Fsc* refers to F-Score for AF detection. TP refers to the number of true positives i.e. correctly detected AF segments, FN refers to number of false negatives i.e. AF epochs misclassified as non-AF, TN refers to the number of true negatives i.e. correctly detected non-AF segments and FP refers to the number of false positives, i.e. non-AF epochs misclassified as AF.

5. Discussion

Automated techniques for accurate AF classification represent a continually improving area of biomedical research. A stable AF classifier is expected to be robust to noise while remaining extremely precise in distinguishing AF rhythms from other arrhythmias as well as normal rhythms. The work described in this paper is a contribution in that direction. Features from a nine-state Markov matrix (derived from SWT analysis) were combined with a Random Forests learning algorithm resulting in efficient signal quality analysis and subsequent R-peak detection. Probabilities from an eight-state Markov matrix (modelling transitions in RR-intervals) along with other RR-interval features in 60s epochs were used to train a new Random Forests model for the purpose of AF detection. From Tables 3 and 4, it can be observed that our AF model achieves consistently high values of sensitivity, specificity and F-scores on different sets of databases with different sampling frequencies - Physionet AF Challenge 2017 (300 Hz) and AFTDB (128 Hz) databases for training; MITDB (360 Hz) and AFDB (250 Hz) databases for testing. This confirms the ability of our algorithm to generalize well across multiple databases irrespective of the underlying sampling rate as long as this information is provided as input to the

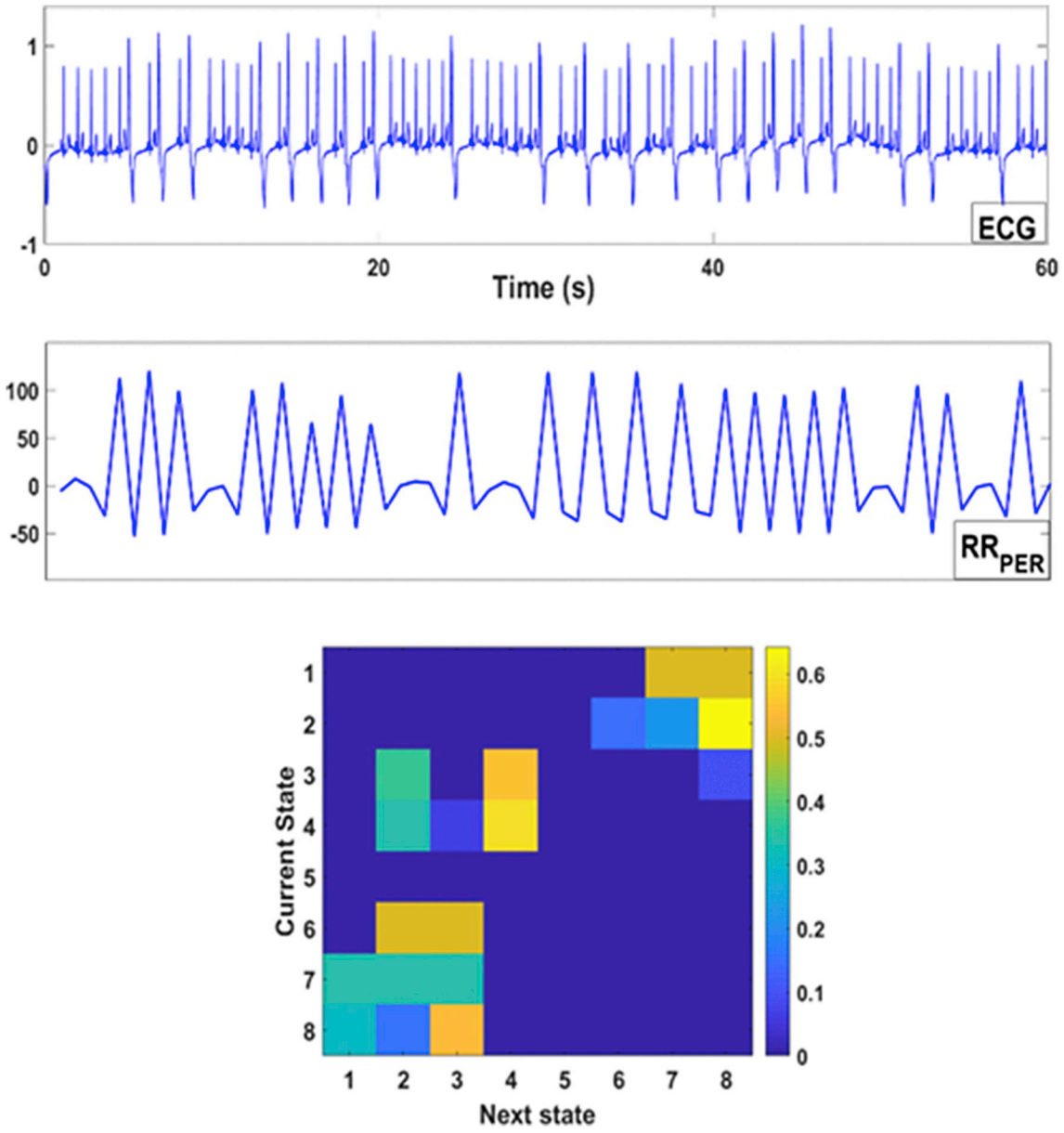

Fig. 4. True negative detection of a non-AF episode. (Top) 60s ECG epoch exhibiting a series of ventricular trigeminy, ventricular bigeminy and normal sinus rhythm coupled with a few ectopic beats. (Middle) The rr_per series values. (Bottom) Corresponding discrete-state probability transition matrix. (Epoch belongs to record 119 from the MITDB database).

algorithm. In this regard, we resample the input signal at 80 Hz for noise analysis (refer to Section 3.3). In the following subsections we provide justification for choosing a Random Forests model for AF classification, explain the significance of the FP suppression step and finally, we compare the performance of our proposed methodology with other state-of-the art AF detection techniques.

5.1. Selection of an optimal learning algorithm

Prior to choosing the Random Forests algorithm for training our AF classifier, we analyzed the performance of several other machine learning algorithms in order to choose the optimal AF classification model. The results are shown in Table 4. The same training data (Table 1) was used across each of the algorithms listed in the table and 10-fold cross validation was carried out for each algorithm. From Table 4, it can be seen that the Random Forests algorithm outperforms other methods in terms of all three metrics, i.e. sensitivity, specificity and F-score. Both Adaboost and bagging were implemented with decision trees. Bagging and Random Forests were implemented with 100 decision trees. Support vector machine (SVM) classifiers were trained with a Gaussian kernel whose gamma parameter was equal to the inverse of the number of features. From Table 4, it can be seen that although the Naïve Bayes algorithm gives the best possible sensitivity, it has a relatively poor F-score, indicating a considerably low specificity for the method. More information regarding these algorithms can be found in [35,36].

5.2. Statistical significance of false-positive suppression

We observed that including a post-classification false positive suppression step improves the overall classification accuracy of our algorithm. This is most noticeable in terms of specificity and F-score for the records in MITDB database in Table 3. It has to be noted that only a few records in the training data possessed ECG signals that had sustained bigeminy/trigeminy and other arrhythmias which would result in significantly large ACF values. Thus, only a small portion of non-AF

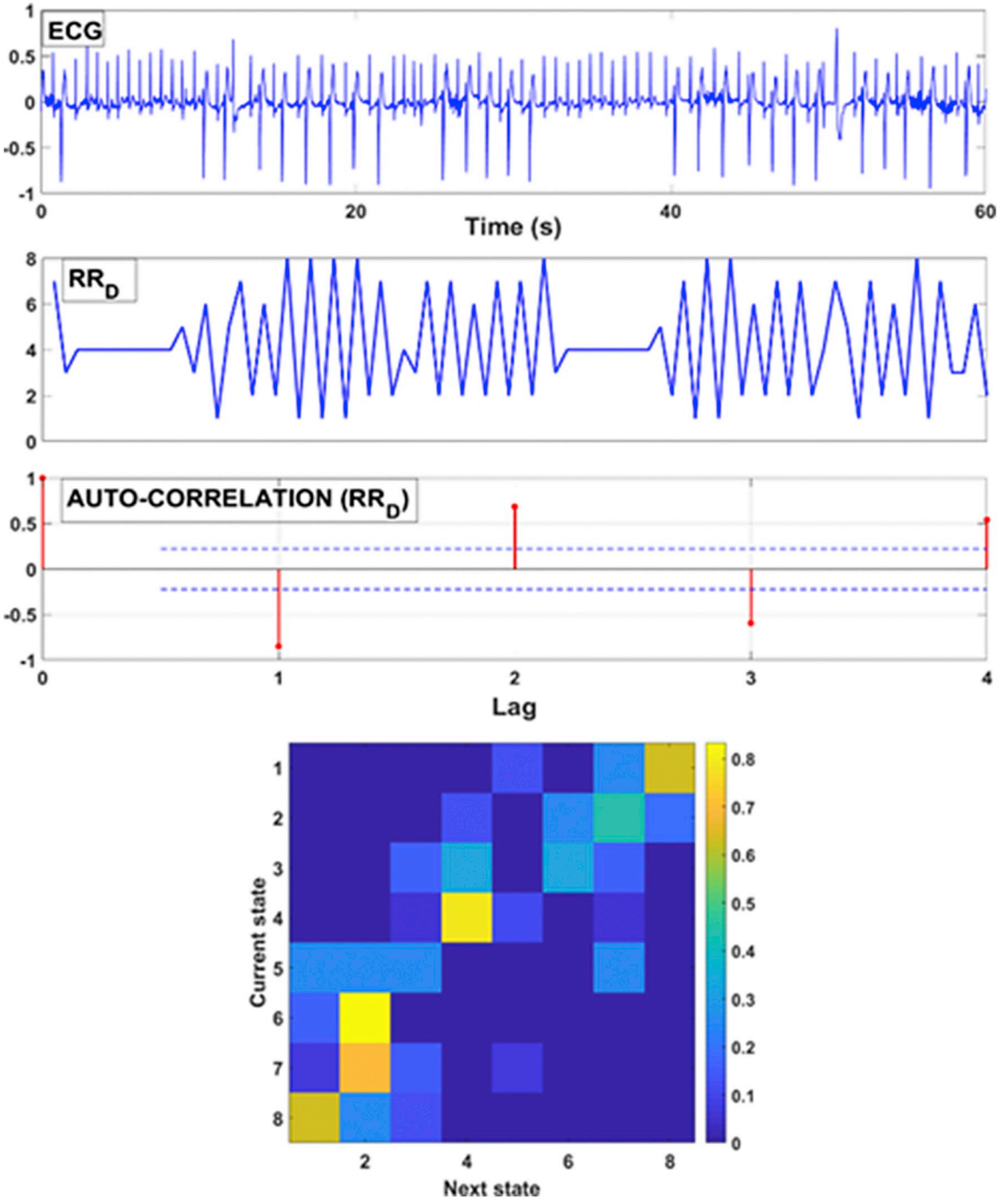

Fig. 5. False positive suppression. (Top) The first plot shows a 60s ECG epoch with ventricular bigeminy and normal sinus rhythm coupled with a few ectopic beats. The second plot shows the rr_d discrete-state sequence. The third plot shows the ACF values. An ACF value of -0.85 at lag 2 reclassifies the ECG epoch as non-AF, thus suppressing the initial false positive classification. (Bottom) Corresponding discrete-state probability transition matrix. It can be seen that the matrix has non-zero values at the center as well along the left-bottom and top-right corners. Hence the AF classifier initially misclassifies this epoch as AF, but the FP suppression routine corrects this. (Epoch belongs to record 200 from the MITDB database).

Table 3
Algorithm performance on AFDB and MITDB databases.

DATABASE	WITHOUT FP SUPPRESSION			WITH FP SUPPRESSION		
	SE	SP	FSC	SE	SP	FSC
AFDB	97.7%	98.5%	97.7%	97.4%	98.6%	97.7%
MITDB (100 series)	N/A	99.6%	N/A	N/A	99.6%	N/A
MITDB (200 series)	96.3%	91.7%	82.4%	96.3%	94.1%	86.5%
MITDB (Overall)	96.3%	95.9%	81.6%	96.3%	97.0%	85.6%

Table 4
Performance of different learning algorithms on the training data.

METHOD	SE (%)	SP (%)	F-SCORE (%)
Random forests	85.00	93.85	86.29
Adaboost	82.21	91.49	82.68
Bagging	83.21	93.21	84.70
Support Vector Machines	76.15	92.30	79.65
Logistic Regression	83.66	93.05	84.82
Naïve bayes	90.04	80.00	78.65

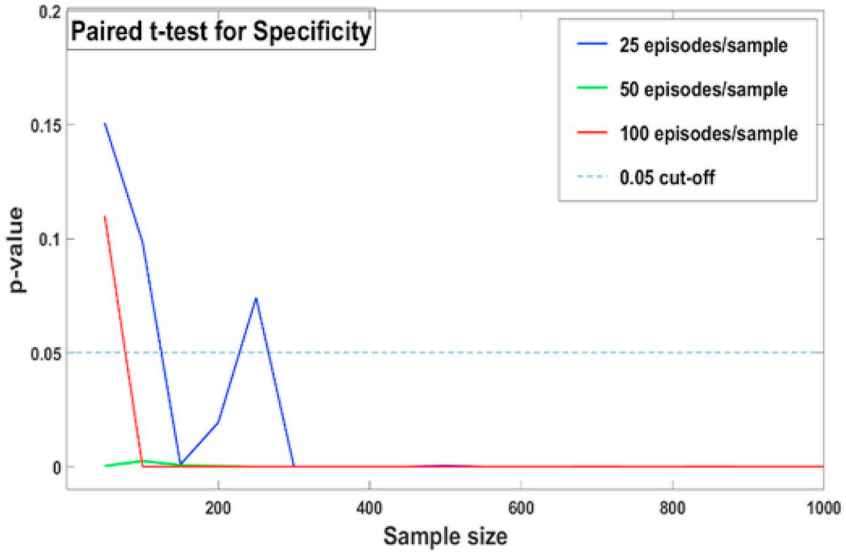

Fig. 6. P-values of the right-tailed paired *t*-test at $\alpha = 0.05$. This test assesses the significance of including a false positive suppression step in improving specificity for records in the MITDB database.

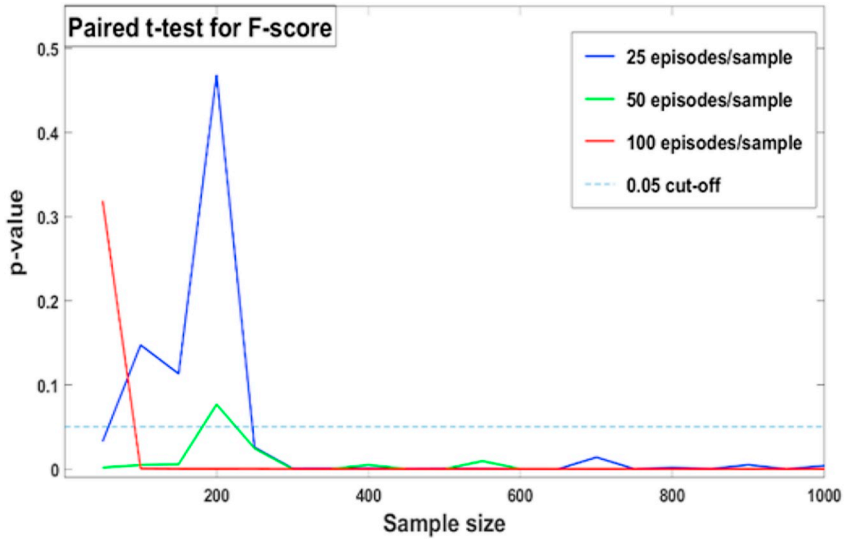

Fig. 7. P-values of the right-tailed paired *t*-test at $\alpha = 0.05$. This test assesses the significance of including a false positive suppression step in improving F-Score for records in the MITDB database.

training examples had significantly useful ACF values. Since Random Forests based learning generally involves subsampling in both the data domain (bagging with replacement) and feature domain (only a subset of features is used for decision splits at each node, usually chosen based on information gain, Gini impurity, etc.), we realized that the Random Forests algorithm could not effectively learn the importance of these sparsely occurring ACF values in the training data. So, we did not include them as part of the features to the Random Forests classifier, and instead added them as a post-classification step. Fig. 5 shows an example of such a scenario where the addition of a false positive suppression step helped in classifying a non-AF segment correctly.

For conclusiveness, we conducted statistical significance tests to confirm our hypothesis that the additional FP suppression step indeed improved our algorithm's efficiency. In this regard, individual paired t-tests were performed on the specificity and F-score values for the MITDB database to verify if these metrics improved consistently with FP suppression.

Random samples containing a subset of AF and non-AF episodes were drawn from the MITDB database and their corresponding specificities and F-score values were computed before and after false positive suppression. The t-tests always assume that the data come from a normal distribution and this assumption was verified using the Kolmogorov-Smirnov goodness-of-fit test for normality at 5% significance level ($\alpha = 0.05$). The resulting p-values were found to exceed 0.05, thus accepting the null hypothesis that the data indeed followed a normal distribution. Following normality assumption validation, separate right-tailed paired t-tests were performed at 5% significance level as follows:

For specificity,

$$H_0: Sp_{with\_FP} \leq Sp_{without\_FP}$$

$$H_A: Sp_{with\_FP} > Sp_{without\_FP}$$

For F-score,

$$H_0: Fsc_{with\_FP} \leq Fsc_{without\_FP}$$

$$H_A: Fsc_{with\_FP} > Fsc_{without\_FP}$$

where, H_0 and H_A are the null and alternate hypotheses respectively for

Table 5
Comparison with other AF methods.

METHOD	REMARKS	DATABASE	SE (%)	SP (%)
Our algorithm	HRD[a]	AFDB	97.4	98.6
		MITDB	96.3	97.0
Moody and Mark [7]	HRD	AFDB	93.5	N/A
		MITDB	N/A	N/A
Tateno and Glass [8]	HRD	AFDB	94.4	97.2
		MITDB	88.2	87.6
Dash *et al* [19]	HRD	AFDB	94.4	95.1
		MITDB	90.2	91.2
Liu *et al* [39]	HRD (ENT[b])	AFDB	98.5	89.9
		MITDB	98.7	79.7
Zhao *et al* [40]	HRD (ENT)	AFDB	96.5	92.6
		MITDB	N/A	N/A
Asgari *et al* [22]	HRI[c]	AFDB	97.0	97.1
		MITDB	N/A	N/A
Ladavich *et al* [11]	HRI	AFDB	98.1	91.7
		MITDB	N/A	N/A
Babaeizadeh *et al* [20]	HRD + AA[d]	AFDB	93.0	98.0
		MITDB	N/A	N/A
Couceiro *et al* [21]	HRD + AA	AFDB	93.8	96.1
		MITDB	N/A	N/A
Xia *et al* [26]	DL[e]	AFDB	98.3	98.2
		MITDB	N/A	N/A

[a] HRD – Heart Rate Dependent.
[b] ENT – Entropy only.
[c] HRI – Heart Rate Independent.
[d] AA – Atrial Activity.
[e] DL – Deep Learning.

each test, SP_{with_FP} is mean specificity with false positive suppression, $SP_{without\_FP}$ is mean specificity without the false positive suppression step, Fsc_{with_FP} is the mean F-score with false positive suppression and $Fsc_{without\_FP}$ is the mean F-score without false positive suppression.

Fig. 6 shows the p-values for different sample sizes and sample lengths (the latter refers to the number of episodes per sample). It can be seen that the p-value decreases below 0.05 with increase in the sample size. It should also be noted that when the number of episodes per sample is at least 100, then the p-value is consistently less than 0.05. Similar inferences can be made from Fig. 7 for the F-score values. These observations confirm the usefulness of including a false positive suppression step in our algorithm.

5.3. Comparison with other AF methods

We compared the performance of our algorithm with other state-of-the-art methods whose results have been reported on the AFDB and/or MITDB databases. These statistics are provided in Table 5. It can be seen that our algorithm performs comparably or better than these algorithms. The first three methods listed after our algorithm in Table 5 are heart-rate dependent and do not utilize atrial activity information. In fact, the work presented in [7] is one of the earliest publications in the field of automated AF analysis. Their algorithm considers a three-state (short, regular and long) Markov model for analyzing the RR-interval sequence and the transition probabilities are compared with that of reference AF transition probabilities for rhythm identification. The authors explain that the use of a Markov model in isolation results in unacceptably high false positive rates and hence they include a filtering and interpolation stage along with an ectopic beat removal step to improve their accuracy. Although reducing data storage, the definition of only three states for RR-intervals could sometimes lead to missing crucial heart rate variation information which can consequently render it difficult to distinguish AF from other similar rhythms. In the method proposed in [8], the distribution of ΔRR intervals for the ECG signal is tested against reference distributions of ΔRR intervals corresponding to AF and non-AF rhythms respectively. Kolmogorov-Smirnov goodness-of-fit tests were used for AF detection. This approach needs storage of reference distributions, thus requiring additional memory. Another potential drawback is that reference distributions as such do not capture the sequential patterns in ΔRR intervals that typically distinguish other rhythms from AF. This could result in increased false positive rates. The authors of [19] use statistical techniques to capture the beat-to-beat variability to aid detect AF. These include Root Mean Square of Successive RR Differences (RMSSD), turning points ratio (TPR) and Shannon entropy metrics. They reason that using these statistical tools ensures that their algorithm is less reliant on the diversity of the training data. They also include an ectopic beat filtering step to remove premature beats. The corresponding results shown in Table 5 are after ectopic beat removal. Although not shown in Table 5, they report an increase of around 22% in specificity on the MITDB database (200 series) and a decrease of around 6% in sensitivity after ectopic beat removal [19]. In contrast, the false positive suppression step in our algorithm results in less than 0.3% degradation in sensitivity for AFDB database (~14 AF epochs). In fact, on closer inspection, it was observed that 12 of these AF epochs belonged to record 04936, where a majority of these AF frames actually exhibited significant bigeminy patterns and hence the ACF values at Lag 1 were fairly high, resulting in our FP suppression stage classifying them as non-AF. On the MITDB database, our FP suppression step did not result in any decrease in sensitivity at all. The approaches of the next two methods are purely entropy based, where AF classification is based on entropy measures corresponding to template matching of RR-segments [39,40]. The results of [39] indicate a higher sensitivity than our approach, but this comes at a significantly lower specificity for both the databases under consideration in Table 5, especially the MITDB database. The authors carried out analysis for three window types, namely, 12-beat, 30-beat and 60-beat windows. In Table 5, we have shown the results for only the 60-beat windows which provided the best accuracy amongst their three window choices. For the MITDB database, their results are provided separately for the MITDB-100 series and MITDB-200 series in their paper. Therefore, we re-computed the specificity for the overall MITDB database using the statistics provided in Tables 1 and 2 in [39]. The sensitivity on the MITDB database is used directly from Table 1 in [39] as this reflects their overall sensitivity. The algorithms in [40] are also developed using similar entropy-based techniques. Their results are computed for 30-beat windows and it can be seen that their accuracy on the AFDB database is lower than our algorithm. The results on the MITDB database were not available for their work.

The subsequent two methods i.e. that of [11,22] are rate-independent and hence do not perform any beat detection. They report similar performance metrics as ours on the AFDB database. The slightly higher sensitivity (~1%) reported by [11] can be attributed to the use of a smaller window size of 7 beats for AF detection, compared to 60s epochs used by our algorithm. But the use of this smaller window size results in decreased specificity (91.7%) and PPV (79.2%), as reported by the authors. The results of [22] are based on 2-fold cross-validation on the AFDB database and evaluations on the MITDB database have not been reported. The next two methods listed in Table 5 i.e. that of [20,21] implement both beat detection as well as atrial activity analysis in their methods. As it can be seen, our algorithm has a much higher sensitivity than theirs on the AFDB database. Additionally, the authors in [7] do not report a specificity for their method. Instead they report a positive predictive value (PPV) of 85.9% on the AFDB database. Similarly, PPV metrics of 96.1%, 79.2% and 98% on the AFDB database are reported by [8,11,20] respectively. In comparison, our algorithm achieved a PPV of 97.9% (not shown in Table 3) for the AFDB database.

A deep learning-based approach listed at the end of Table 5 [26], consists of Deep Convolutional Neural Network based AF classifiers. They use 5s epochs for AF detection that can help detect very short AF episodes. They propose two kinds of inputs for their deep learning models, (i) using Short-Term Fourier Transforms and (ii) using SWT, respectively. Both the techniques provide similar accuracies. Their results are based on ten-fold cross-validation performed on the AFDB

database. A sensitivity of 98.3% and a specificity of 98.2% is reported on the AFDB database, which is slightly better than our sensitivity and slightly lower than our specificity. It would be interesting to see their results on the MITDB database as well which would help us in assessing their algorithm's ability to generalize across multiple databases. In fact, most algorithms in Table 5 report metrics evaluated on the AFDB database (usually cross-validated) and not on other databases such as the MITDB database, thus offering limited information regarding generalization capabilities of their respective techniques. Other recent contributions in the field of automated AF detection also use deep learning [27,28,31] and ensemble approaches [30] to develop AF classifiers. In the deep-learning based approach of [31], the authors trained a CNN-based arrhythmia detector using privately collected ECG data that achieved an AF detection F-score of 66.7% on the MITDB database. It is worth noting that their classifier was trained to classify fourteen different arrhythmias and not just atrial fibrillation. The AF classification algorithms published in [27–30] were developed and validated on the Physionet/Computing in Cardiology Challenge 2017 training and test datasets. Their algorithms were top ranked in the challenge, achieving F-scores in excess of 83% on the test dataset. They do not report accuracy statistics on the AFDB and MITDB databases, and hence they have not been included in Table 5 for comparison.

6. Conclusion

In conclusion, this paper describes a machine-learning based approach employing a combination of discrete-state Markov models and Random forests to detect AF episodes from single-lead ECG signals. The proposed features based on a nine-state Markov model for signal quality analysis and an eight-state Markov model for AF classification result in comparable or better AF detection performance than other state-of-the art AF detection techniques. An additional false positive suppression stage results in a 1.1% increase in specificity and 4.0% increase in F-Score on the MITDB database without any significant decrease in sensitivity. We use auto-correlation analysis of RR-interval series to suppress possible false positives, unlike most other approaches which usually implement atrial activity analysis. This makes our false suppression step more robust to signal noise and less dependent on ECG lead orientation and other factors that can potentially affect accurate analysis of P-waves/f-waves and which as a result could lead to suppression of true AF episodes. Some of the limitations of our approach include skipping detection of very short AF episodes owing to the use of 60s epochs. Also, since we do not perform any atrial activity analysis, our algorithm's accuracy inherently depends on the quality of beat detection although our beat detection technique is accurate [5] and satisfactory for the purpose of AF classification. Many improvements can be incorporated in the future that can further increase the accuracy of our algorithm. Discrete-state Markov matrices of higher orders can be used to distinguish RR-interval state transitions corresponding to non-AF and AF episodes more efficiently. Also, improved signal acquisition setups can result in accurate P-wave detection that could result in efficient atrial activity analysis. This can further help reduce false alarms and increase overall accuracy.

Conflicts of interest

None Declared.

References

[1] C. January, L. Wann, J. Alpert, H. Calkins, J. Cigarroa, J. Cleveland, J. Conti, P. Ellinor, M. Ezekowitz, M. Field, K. Murray, R. Sacco, W. Stevenson, P. Tchou, C. Tracy, C. Yancy, Aha/acc/hrs guideline for the management of patients with atrial fibrillation: a report of the American college of Cardiology/American heart association task force on practice guidelines and the heart rhythm society, Circulation 130 (23) (2014) e199–e267 2014.

[2] P. Kirchhof, S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, M. Castella, H. Diener, H. Heidbuchel, J. Hendriks, G. Hindricks, A. Manolis, J. Oldgren, B. Popescu, U. Schotten, B. Van Putte, P. Vardas, S. Agewall, J. Camm, G. Baron Esquivias, W. Budts, S. Carerj, F. Casselman, A. Coca, R. De Caterina, S. Deftereos, D. Dobrev, J. Ferro, G. Filippatos, D. Fitzsimons, B. Gorenek, M. Guenoun, S. Hohnloser, P. Kolh, G. Lip, A. Manolis, J. McMurray, P. Ponikowski, R. Rosenhek, F. Ruschitzka, I. Savelieva, S. Sharma, P. Suwalski, J. Tamargo, C. Taylor, I. Van Gelder, A. Voors, S. Windecker, J. Zamorano, K. Zeppenfeld, ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS, Eur. Heart J. 37 (38) (2016) 2893–2962 2016.

[3] G. Veenhuyzen, C. Simpson, H. Abdollah, Atrial fibrillation, Can. Med. Assoc. J. 171 (7) (2004) 755–760.

[4] M. Zoni-Berisso, F. Lercari, T. Carazza, S. Domenicucci, Epidemiology of atrial fibrillation: European perspective, Clin. Epidemiol. 6 (2014) 213–220.

[5] V. Kalidas, L. Tamil, Real-time QRS detector using stationary wavelet transform for automated ECG analysis, IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE), 2017, 2017, pp. 457–461.

[6] M. Carrara, L. Carozzi, J. Moss, M. de Pasquale, S. Cerutti, M. Ferrario, D.E. Lake, J.R. Moorman, Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy, Physiol. Meas. 36 (2015) 1873–1888.

[7] G.B. Moody, R.G. Mark, A new method for detecting atrial fibrillation using R-R intervals, Comput. Cardiol. (1983) 227–230.

[8] K. Tateno, L. Glass, "Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of RR and ΔRR intervals, Med. Biol. Eng. Comput. 39 (2001) 664–671.

[9] S. Sarkar, D. Ritscher, R. Mehra, A detector for a chronic implantable atrial tachyarrhythmia monitor, IEEE Trans. Biomed. Eng. 55 (3) (2008) 1219–1224.

[10] J. Park, S. Lee, M. Jeon, Atrial fibrillation detection by heart rate variability in poincare plot, Biomed. Eng. Online 8 (2009).

[11] S. Ladavich, B. Ghoraani, Rate-independent detection of atrial fibrillation by statistical modeling of atrial activity, Biomed. Signal Process. Control 18 (2015) 274–281.

[12] X. Du, N. Rao, M. Qian, D. Liu, J. Li, W. Feng, L. Yin, X. Chen, A novel method for real-time atrial fibrillation detection in electrocardiograms using multiple parameters, Ann. Noninvasive Electrocardiol. 19 (2014) 217–225.

[13] V. Marozas, L. Sornmo Petrenas, A. Lukosevicius, An echo state neural network for QRST cancellation during atrial fibrillation, IEEE Trans. Biomed. Eng. 59 (10) (2012) 2950–2957.

[14] J. Rodenas, M. Garcia, R. Alcaraz, J.J. Rieta, Wavelet entropy automatically detects episodes of atrial fibrillation from single-lead electrocardiograms, Entropy 17 (2015) 6179–6199.

[15] G.D. Clifford, C. Liu, B. Moody, L.H. Lehman, I. Silva, Q. Li, A.E. Johnson, R.G. Mark, AF classification from a single short lead ECG recording: the Physionet/Computing in Cardiology Challenge 2017, Computing in Cardiology, 2017.

[16] G.B. Moody, Spontaneous termination of atrial fibrillation: a challenge from Physionet and Computers in Cardiology 2004, Computers in Cardiology, 2004, pp. 101–104.

[17] L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C. Peng, H.E. Stanley, Physiobank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals, Circulation 101 (2000) e215–e220.

[18] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Engineering in Medicine and Biology Magazine, vol. 20, 2001, pp. 45–50 no. 3.

[19] S. Dash, K.H. Chon, S. Lu, E.A. Raeder, Automatic real time detection of atrial fibrillation, Ann. Biomed. Eng. 37 (2009) 1701–1709.

[20] S. Babaeizadeh, R.E. Gregg, E.D. Helfenbein, J.M. Lindauer, S.H. Zhou, Improvements in atrial fibrillation detection for real-time monitoring, J. Electrocardiol. 42 (2009) 522–526.

[21] R. Couceiro, P. Carvalho, J. Henriques, M. Antunes, M. Harris, J. Habetha, Detection of atrial fibrillation using model-based ECG analysis, 2008 19th International Conference on Pattern Recognition, 2008, pp. 1–5.

[22] S. Asgari, A. Mehrnia, M. Moussavi, Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine, Comput. Biol. Med. 60 (2015) 132–142.

[23] G.B. Moody, W.E. Muldrow, R.G. Mark, A noise stress test for arrhythmia detectors, Comput. Cardiol. 11 (1984) 381–384.

[24] G.P. Nason, B.W. Silverman, The Stationary Wavelet Transform and Some Statistical Applications, Springer New York, New York, 1995, pp. 281–299.

[25] M. Elgendi, M. Jonkman, F.D. Boer, Frequency bands effects on QRS detection, Proc. international joint conference on biomedical engineering systems and technologies, biosignals, 2010 2010, pp. 428–431.

[26] Y. Xia, N. Wulan, K. Wang, H. Zhang, Detecting atrial fibrillation using deep convolutional neural networks, Comput. Biol. Med. 93 (2018) 82–94.

[27] S. Hong, M. Wu, Y. Zhou, Q. Wang, J. Shang, H. Li, J. Xie, ENCASE: an ensemble classifier for ECG classification using expert features and deep neural networks, Computing in Cardiology, 2017.

[28] T. Teijeiro, C.A. Garcia, D. Castro, P. Felix, Arrhythmia classification from the abductive interpretation of short single-lead ECG records, Computing in Cardiology, 2017.

[29] M. Zabihi, A.B. Rad, A.K. Katsaggelos, S. Kiranyaz, S. Narkilahti, M. Gabbouj, Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier, Computing in Cardiology, 2017.

[30] S. Datta, C. Puri, A. Mukherjee, R. Banerjee, A.D. Choudhury, R. Singh, A. Ukil, S. Bandyopadhyay, A. Pal, S. Khandelwal, Identifying normal, AF and other abnormal ECG rhythms using a cascaded binary classifier, Computing in Cardiology, 2017.

[31] P. Rajpurkar, A.Y. Hannun, M. Haghpanahi, C. Bourn, A.Y. Ng, Cardiologist-Level Arrhythmia Detection with Convolutional Neural Net-Works, (July 2017) arXiv:1707.01836 [cs.CV].
[32] M.A. Stephens, EDF statistics for goodness of fit and some comparisons, J. Am. Stat. Assoc. 69 (347) (1974) 730–737.
[33] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[34] L. Breiman, Random forests, Mach. Learn. 45 (1) (Oct. 2001) 5–32.
[35] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.
[36] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2001.
[37] S. Cheng, L.S. Tamil, B. Levine, A mobile health system to identify the onset of paroxysmal atrial fibrillation, 2015 International Conference on Healthcare Informatics, Dallas, TX, 2015, pp. 189–192.
[38] D.E. Lake, J.R. Moorman, Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted ventricular devices, Am. J. Physiol. Heart Circ. Physiol. 300 (1) (2011) H319–H325.
[39] C. Liu, J. Oster, E. Reinertsen, Q. Li, L. Zhao, S. Nemati, G. Clifford, A comparison of entropy approaches for AF discrimination, Physiol. Meas. 39 (2018) 074002.
[40] L. Zhao, Z. Liu, S. Wei, Q. Shen, F. Zhou, J. Li, A new entropy-based atrial fibrillation detection method for scanning wearable ECG recordings, Entropy 20 (12) (2018) 904.
[41] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer, 2014.