



# Kalman-based Spectro-Temporal ECG Analysis using Deep Convolutional Networks for Atrial Fibrillation Detection

Zheng Zhao<sup>1</sup> · Simo Särkkä<sup>1</sup> · Ali Bahrami Rad<sup>1</sup>

Received: 11 December 2018 / Revised: 20 December 2019 / Accepted: 10 March 2020 / Published online: 30 April 2020  
© The Author(s) 2020

## Abstract

In this article, we propose a novel ECG classification framework for atrial fibrillation (AF) detection using spectro-temporal representation (i.e., time varying spectrum) and deep convolutional networks. In the first step we use a Bayesian spectro-temporal representation based on the estimation of time-varying coefficients of Fourier series using Kalman filter and smoother. Next, we derive an alternative model based on a stochastic oscillator differential equation to accelerate the estimation of the spectro-temporal representation in lengthy signals. Finally, after comparative evaluations of different convolutional architectures, we propose an efficient deep convolutional neural network to classify the 2D spectro-temporal ECG data. The ECG spectro-temporal data are classified into four different classes: AF, non-AF normal rhythm (Normal), non-AF abnormal rhythm (Other), and noisy segments (Noisy). The performance of the proposed methods is evaluated and scored with the PhysioNet/Computing in Cardiology (CinC) 2017 dataset. The experimental results show that the proposed method achieves the overall F1 score of 80.2%, which is in line with the state-of-the-art algorithms.

**Keywords** ECG analysis · Atrial fibrillation · Deep learning · Kalman filter · Spectrogram estimation

## 1 Introduction

Atrial fibrillation (AF) is the most common cardiac arrhythmia, and its prevalence is around 1–2% worldwide [7]. It is also estimated that by 2030 only in European Union 14–17 million patients suffer from AF [56]. AF is associated with an increased risk of having stroke (5-fold), blood clots, heart failure, coronary artery disease, or death (2-fold; death rates are doubled by AF) [7]. Therefore, developing automatic algorithms for early detection of AF is crucial.

During AF atrial muscle fibers have chaotic electrical activity which may emit impulses with 500 bpm rate to atrioventricular (AV) node, from which impulses pass randomly. This results to an irregular ventricular response which is one of the main characteristics of AF [49].

In addition, AF has the following characteristics on electrocardiogram (ECG): 1) “absolutely” irregular RR intervals; 2) the absence of P waves; and 3) variable atrial cycle length (when visible).

The analysis of ECG is the most common approach to AF detection, and during the past ten years, various algorithms have been developed for automatic AF detection [2–4, 6, 11, 15, 28, 52, 53]. Most of the existing algorithms follow a pipeline of preprocessing, feature extraction/selection, and classification. For example, automatic methods based on the standard classifiers such as support vector machine and random forest with carefully designed features have been successfully used for AF detection by achieving state-of-the-art performance [28, 53]. However, those methods require manual feature extraction, which often needs extensive human effort and domain knowledge/expertise. This issue can be bypassed by using the end-to-end deep learning (DL) techniques [25]. Deep neural networks can learn the inherent features directly from the input signal by providing a sufficient amount of training data [13]. A typical example is a deep convolutional neural network, where the feature extraction is done and learned automatically in the convolutional layers. The application of deep learning for

---

This work was supported by Business Finland

✉ Zheng Zhao  
zgbkdlm@gmail.com

<sup>1</sup> Department of Electrical Engineering and Automation, Aalto University, Rakentajanaukio 2c, Espoo, 02150, Finland

AF detection have just started (see, e.g., [1, 27, 31, 35, 37, 39, 43, 50]).

For ECG signals, one can directly adopt 1D convolutional or recurrent network models for the classification task. However, transforming signals into spectral domain (spectro-temporal features) is a promising alternative approach knowing that the current state-of-the-art deep convolutional neural networks (CNNs) structures are typically designed for 2D images. Deep CNNs such as AlexNet [24], Inception-v4 [45], and DenseNet [20] have proved their superiority in image classification.

Within the previous studies, only a few have resorted to the use of time-varying spectrum for AF detection. The reasons might be the following. First, it is not easy to select hand-crafted features from 2D data using traditional classifiers. Second, the temporal features of spectrogram are usually hard to capture even in DL setting. Several studies [50, 55] have endeavoured DL for AF detection in spectral domain, but the use of traditional spectral estimation methods such as short-time Fourier transform (STFT) or continuous wavelet transform (CWT) may drop momentous information during the transformation, and produce less informative input data. Thus, to unravel these problems, it is beneficial to consider new spectro-temporal estimation methods that retain the temporal features better.

The contributions of this paper are: 1) We propose two extended models for spectro-temporal estimation using Kalman filter and smoother. We then combine them with deep convolutional networks for AF detection. 2) We test and compare the performance of proposed approaches for spectro-temporal estimation on simulated data and AF detection with other popular estimation methods and different classifiers. 3) For AF detection, we evaluate the proposals using PhysioNet/CinC 2017 dataset [8], which is considered to be a challenging dataset that resembles practical applications, and our results are in line with the state-of-the-art.

It is worth mentioning that, most of the deep learning algorithms for ECG analysis have been developed on the MIT-BIH arrhythmia ECG database [1, 27, 29, 39]. However, in this work, we use the PhysioNet/CinC 2017 dataset which is more appropriate for AF detection, and it is the most challenging and recent publicly available ECG dataset. This dataset was collected using AliveCor wireless hand-held devices which are used for personal recording and outpatient monitoring. This dataset is collected particularly for AF detection in a real-world scenario. The best performance (averaged F1 score) currently on this dataset is only 0.83 [9, 16, 19, 48, 53].

This paper is an extended version based of our previous conference paper “Spectro-temporal ECG Analysis for Atrial Fibrillation Detection” [54] presented at 2018 IEEE 28th International Workshop on Machine Learning for

Signal Processing. In addition to the original contributions in the conference article, in this article, we use a new stochastic oscillator model and show that the spectro-temporal estimation can also be implemented with a steady state (stationary) Kalman filter and smoother, which leads to a significant reduction in time consumption without losing estimation accuracy. We demonstrate this in both simulated data and AF data classification. In addition to the experiments in the conference paper, where we only showed a few comparisons among estimation methods and classifiers, we expand them to a wide range of both standard and modern (e.g., Random Forests, CNNs, and DenseNet) classifiers for a better and more solid illustration of the classification performance.

The paper is structured as follows: In Section 2, we propose spectro-temporal methods for ECG signal analysis. In Section 3, we apply the proposed estimation method to AF detection using an averaging procedure. In Section 4, we compare and discuss experimental results both in simulated data and ECG dataset, followed by conclusion in Section 5.

## 2 Spectro-Temporal Estimation Methods

Spectro-temporal signal analysis is an effective and powerful approach that is used in many fields ranging from biosignal analysis [34] and audio processing [33] to weather forecasting [10] and stock market prediction [21]. In ECG analysis, the temporal evolution of spectral information can be captured in spectro-temporal data representation, which can convey important information about the underlying biological process of the heart.

In this section, we develop new methods for spectro-temporal estimation. We first introduce a Fourier series model based upon the Bayesian spectrum estimation method of Qi et al. [32], and put Gaussian process priors on the Fourier coefficients. Then, by adopting the ideas presented in [42], we convert the Fourier series into a more flexible stochastic oscillator model and use a fast stationary Kalman filter/smoothing for its estimation. Finally, we demonstrate the estimation performance on simulated data.

### 2.1 Kalman-based Fourier Series Model for Spectro-Temporal Estimation

Apart from traditional STFT and CWT methods, the spectro-temporal analysis can also be done by modeling the signal as a stochastic state-space model and resorting to the Bayesian procedure (i.e., Kalman filter and smoother) for its estimation [32, 40]. The key advantages of this kind of approaches over other spectro-temporal methods are that we can apply them to both evenly and unevenly sampled

signals [32] and they require no stationarity guarantees nor windowing. Furthermore, as we show here, they can also be combined with state-space methods for Gaussian processes [17, 41].

Recall that any periodic signal with fundamental frequency  $f_0$  can be expanded into a Fourier series

$$z(t) = a_0 + \sum_{j=1}^M [a_j \cos(2\pi j f_0 t) + b_j \sin(2\pi j f_0 t)], \quad (1)$$

where the exact representation is obtained with  $M \rightarrow \infty$ , but for sampled (and thus bandlimited) signals it is sufficient to consider finite series. This stationary model is the underlying model in the STFT approach. STFT applies a window to each signal segment and finds a least squares fit (via discrete Fourier transform) to the coefficients  $\{a_j, b_j : j = 1, \dots, M\}$ .

In our approach, we start by assuming that the coefficients depend on time, and we put Gaussian process priors on them:

$$\begin{aligned} a_j(t) &\sim \mathcal{GP}(0, k_j^a(t, t')), \\ b_j(t) &\sim \mathcal{GP}(0, k_j^b(t, t')). \end{aligned} \quad (2)$$

As shown in [17, 41], provided that the covariance functions are stationary, we can express the Gaussian processes as solutions to linear stochastic differential equations (SDEs). We choose the covariance functions to have the form

$$\begin{aligned} k_j^a(t, t') &= (s_j^a)^2 \exp(-\lambda_j^a |t - t'|), \\ k_j^b(t, t') &= (s_j^b)^2 \exp(-\lambda_j^b |t - t'|), \end{aligned} \quad (3)$$

where  $s_j^a, s_j^b > 0$  are scale parameters and  $\lambda_j^a, \lambda_j^b > 0$  are the inverses of the time constants (length scales) of the processes.

The state-space representations (which are scalar in this case) are then given as

$$\begin{aligned} da_j &= -\lambda_j^a a_j dt + dW_j^a, \\ db_j &= -\lambda_j^b b_j dt + dW_j^b, \end{aligned} \quad (4)$$

where  $W_j^a, W_j^b$  are Brownian motions with suitable diffusion coefficients  $q_j^a, q_j^b$ . We can also solve the equations at discrete time steps (see, e.g., [14]) as

$$\begin{aligned} a_j(t_k) &= \psi_{jk}^a a_j(t_{k-1}) + w_{jk}^a, & w_{jk}^a &\sim \mathcal{N}(0, \Sigma_{jk}^a), \\ b_j(t_k) &= \psi_{jk}^b b_j(t_{k-1}) + w_{jk}^b, & w_{jk}^b &\sim \mathcal{N}(0, \Sigma_{jk}^b), \end{aligned} \quad (5)$$

where

$$\begin{aligned} \psi_{jk}^a &= \exp(-\lambda_j^a (t_k - t_{k-1})), \\ \psi_{jk}^b &= \exp(-\lambda_j^b (t_k - t_{k-1})), \\ \Sigma_{jk}^a &= q_j^a (1 - \exp(-2\lambda_j^a (t_k - t_{k-1}))), \\ \Sigma_{jk}^b &= q_j^b (1 - \exp(-2\lambda_j^b (t_k - t_{k-1}))). \end{aligned} \quad (6)$$

Let us now assume that we obtain noisy measurements of the Fourier series (1) at times  $t_1, t_2, \dots$ . What we can now do is to define a state vector  $\mathbf{x} =$

$[a_0, a_1, \dots, a_M, b_1, b_2, \dots, b_M]^T$  which stacks all the coefficients  $a_j$  and  $b_j$ . In this way, we can write  $\mathbf{H}_k = [1, \cos(2\pi f_0 t_k), \dots, \cos(2\pi M f_0 t_k), \sin(2\pi f_0 t_k), \dots, \sin(2\pi M f_0 t_k)]$ , which leads to

$$\begin{aligned} z(t_k) &= a_0 + \sum_{j=1}^M [a_j \cos(2\pi j f_0 t_k) + b_j \sin(2\pi j f_0 t_k)] \\ &= \mathbf{H}_k \mathbf{x}_k. \end{aligned} \quad (7)$$

We can also rewrite the dynamic model (5) as

$$\mathbf{x}_k = \Psi_k \mathbf{x}_{k-1} + \mathbf{q}_k, \quad (8)$$

where  $\Psi_k$  contains the terms  $\psi_{jk}^a$  and  $\psi_{jk}^b$  on the diagonal and  $\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_k)$  where  $\Sigma_k$  contains the terms  $\Sigma_{jk}^a$  and  $\Sigma_{jk}^b$  on the diagonal.

If we assume that we actually measure (7) with additive Gaussian measurement noise  $r_k \sim \mathcal{N}(0, R)$ , then we can express the measurement model as

$$y_k = \mathbf{H}_k \mathbf{x}_k + r_k. \quad (9)$$

Equations 8 and 9 define a linear state-space model where we can perform exact Bayesian estimation using Kalman filter and smoother [40]. In the original paper [32], the state vectors  $\mathbf{x}_1, \dots, \mathbf{x}_N$  are assumed to perform random walk, but here the key insight is to use a more general Gaussian process which introduces a finite time constant to the problem. Although here we have chosen to use quite simple Gaussian process model for this purpose, it would also be possible to use more general Gaussian process priors for the coefficients such as state-space representations of Matérn or squared exponential covariance functions [17, 41].

The Kalman filter for this problem then consists of the following forward recursion (for  $k = 1, \dots, N$ ):

$$\begin{aligned} \mathbf{m}_k^- &= \Psi_k \mathbf{m}_{k-1}, \\ \mathbf{P}_k^- &= \Psi_k \mathbf{P}_{k-1} \Psi_k^T + \Sigma_k, \\ S_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R, \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T / S_k, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k (y_k - \mathbf{H}_k \mathbf{m}_k^-), \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k S_k \mathbf{K}_k^T, \end{aligned} \quad (10)$$

and the RTS smoother the following backward recursion (for  $k = N - 1, \dots, 1$ ):

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_k \Psi_{k+1}^T [\mathbf{P}_{k+1}^-]^{-1}, \\ \mathbf{m}_k^s &= \mathbf{m}_k + \mathbf{G}_k [\mathbf{m}_{k+1}^s - \mathbf{m}_{k+1}^-], \\ \mathbf{P}_k^s &= \mathbf{P}_k + \mathbf{G}_k [\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-] \mathbf{G}_k^T. \end{aligned} \quad (11)$$

The final posterior distributions are then given as:

$$p(\mathbf{x}_k | y_{1:N}) = \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1, \dots, N. \quad (12)$$

The magnitude of the sinusoidal with frequency  $f_j = j f_0$  at time step  $k$  can then be computed by extracting the

elements corresponding to  $\hat{a}_j(t_k)$  and  $\hat{b}_j(t_k)$  from the mean vector  $\mathbf{m}_k^s$ :

$$[\mathbf{S}]_{j,k} = \sqrt{\hat{a}_j^2(t_k) + \hat{b}_j^2(t_k)}. \quad (13)$$

From now on, matrix  $\mathbf{S}$  is called spectro-temporal data matrix.

## 2.2 Oscillator Model for Spectro-Temporal Estimation

In practice, the computational cost of Kalman filter and smoother can be extensive when the length of the signal is very long. However, instead of the Fourier series state space model in previous section, one can also derive an alternative representation using stochastic oscillator differential equations. In this way, the dynamic and measurement models become linear time-invariant (LTI) so that we can leverage a stationary Kalman filter to reduce the time consumption. This kind of stochastic oscillator models were also considered in [42] and the link to period Gaussian process models was investigated in [44].

A single quasi-period stochastic oscillator can be described with the following stochastic differential equation model [44]:

$$d\mathbf{x}^j = \begin{bmatrix} -\lambda_j & -2\pi f_j \\ 2\pi f_j & -\lambda_j \end{bmatrix} \mathbf{x}^j dt + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} d\mathbf{W}_j, \quad (14)$$

$$= \mathbf{F}_j \mathbf{x}^j dt + \mathbf{L} d\mathbf{W}_j.$$

where  $\mathbf{x}^j = [a_j \ b_j]^\top$  and the Brownian motion  $\mathbf{W}_j = [W_j^a \ W_j^b]^\top$  has a suitably chosen diffusion matrix  $\zeta^j = q_j \mathbf{I}$  [44]. By solving the SDE in discrete time steps, we have

$$\mathbf{x}_k^j = \mathbf{A}^j \mathbf{x}_{k-1}^j + \mathbf{q}^j, \quad \mathbf{q}^j \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^j), \quad (15)$$

where  $\mathbf{A}^j$  and  $\mathbf{Q}^j$  are given by:

$$\mathbf{A}^j = \exp(\mathbf{F}_j \Delta t),$$

$$\mathbf{Q}^j = \int_0^{\Delta t} \exp(\mathbf{F}_j (\Delta t - s)) \mathbf{L} \zeta^j \mathbf{L}^\top \times \exp(\mathbf{F}_j (\Delta t - s))^\top ds, \quad (16)$$

where  $\Delta t = t_k - t_{k-1}$ .

A general quasi-periodic signal can be modeled using a superposition of stochastic oscillators of the above form [44]. If we construct  $\mathbf{x}_k = [(\mathbf{x}_k^0)^\top (\mathbf{x}_k^1)^\top \dots (\mathbf{x}_k^M)^\top]^\top$ , then the resulting time-invariant model can be written as:

$$\mathbf{x}_k = \mathbf{A} \mathbf{x}_{k-1} + \mathbf{q}_k, \quad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

$$y_k = \mathbf{H} \mathbf{x}_k + r_k, \quad r_k \sim \mathcal{N}(0, R), \quad (17)$$

where  $\mathbf{A}$ ,  $\mathbf{Q}$  and  $\mathbf{H}$  are defined as:

$$\mathbf{A} = \begin{bmatrix} 1 & & & \\ & \mathbf{A}^1 & & \\ & & \ddots & \\ & & & \mathbf{A}^M \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} q_b \Delta t & & & \\ & \mathbf{Q}^1 & & \\ & & \ddots & \\ & & & \mathbf{Q}^M \end{bmatrix}, \quad (18)$$

$$\mathbf{H} = [\mathbf{H}^1 \dots \mathbf{H}^M] = [1 \ 1 \ 0 \ 1 \ 0 \dots \ 1 \ 0]. \quad (19)$$

In this model, the first component of the state is a slowly drifting Brownian motion with diffusion coefficient  $q_b$  modeling the possible non-zero mean of the signal.

The estimation problem can be solved with a Kalman filter and smoother. However, because the model is LTI, the Kalman filter is known to converge to a steady-state Kalman filter [22]. The steady-state Kalman filter can be obtained by solving the following discrete algebraic Riccati equation (DARE) for the limit covariance  $\mathbf{P}_k^- \rightarrow \mathbf{P}_\infty^-$ :

$$\mathbf{P}_\infty^- = \mathbf{A} \mathbf{P}_\infty^- \mathbf{A}^\top + \mathbf{Q} - \mathbf{A} \mathbf{P}_\infty^- \mathbf{H}^\top (\mathbf{H} \mathbf{P}_\infty^- \mathbf{H}^\top + R)^{-1} \mathbf{H} \mathbf{P}_\infty^- \mathbf{A}^\top. \quad (20)$$

A positive-semi-definite solution to the equation is known to exist provided that the pair  $[\mathbf{A}, \mathbf{H}]$  is detectable [22].

Thus we can obtain  $\mathbf{P}_\infty^-$  by solving DARE in Eq. 20, and the stationary Kalman filter for the forward mean propagation is:

$$\mathbf{m}_k = \mathbf{A} \mathbf{m}_{k-1} + \mathbf{K} (y_k - \mathbf{H} \mathbf{A} \mathbf{m}_{k-1}), \quad (21)$$

where the stationary gain is

$$\mathbf{K} = \mathbf{P}_\infty^- \mathbf{H}^\top (\mathbf{H} \mathbf{P}_\infty^- \mathbf{H}^\top + R)^{-1}. \quad (22)$$

The corresponding smoother then turns out to converge to its steady state as well, and the backward propagation for the resulting steady-state smoother is:

$$\mathbf{m}_k^s = \mathbf{m}_k + \mathbf{G} (\mathbf{m}_{k+1}^s - \mathbf{A} \mathbf{m}_k). \quad (23)$$

where the gain is computed as

$$\mathbf{G} = \mathbf{P}_\infty^- \mathbf{A}^\top [\mathbf{P}_\infty^-]^{-1},$$

$$\mathbf{P}_\infty^- = \mathbf{P}_\infty^- - \mathbf{P}_\infty^- \mathbf{H}^\top (\mathbf{H} \mathbf{P}_\infty^- \mathbf{H}^\top + R)^{-1} \mathbf{H} \mathbf{P}_\infty^-. \quad (24)$$

In this way, the calculation of the filter and covariances at every time step is not needed, which reduces the computational cost significantly. The disadvantage is that we need to solve the DARE in order to construct the stationary filter and smoother, which also adds to the computational cost.

After computing the estimates  $\mathbf{m}_k^s$  for each time step, we can extract the estimates of  $\hat{a}_j(t_k)$  and  $\hat{b}_j(t_k)$  and use Eq. 13 to compute the spectro-temporal data matrix.

## 2.3 Estimation Trials on Simulated Data

A quantitative evaluation of the proposed spectro-temporal methods for ECG classification is discussed in Sections 4

and 5.2. However, in this section we visually inspect the proposed spectro-temporal representations on the simulated data and compare them with other standard time-frequency approaches such as STFT, CWT, and BurgAR. To avoid confusion in terminology, from now on, we refer the proposals in Sections 2.1 and 2.2 as FourierKS and OscKS, respectively.

We simulated a noise-observed multi-sinusoidal signal  $y(t)$  as shown in Eq. 25 and Fig. 1 with time step  $\Delta t = 0.1$  and  $\varepsilon_k \sim N(0, 0.1^2)$ .

$$y(t_k) = \varepsilon_k + \begin{cases} \sin(2\pi \cdot 0.01 t_k) + \sin(2\pi \cdot 0.3 t_k), & 1 \leq t_k < 150 \\ \sin(2\pi \cdot 0.2 t_k) + \sin(2\pi \cdot 0.3 t_k), & 150 \leq t_k < 250 \\ \sin(2\pi \cdot 0.13 t_k) + \sin(2\pi \cdot 0.2 t_k), & 250 \leq t_k < 300 \\ \sin(2\pi \cdot 0.2 t_k) + \sin(2\pi \cdot 0.43 t_k), & 300 \leq t_k < 400 \\ \sin(2\pi \cdot 0.1 t_k) + \sin(2\pi \cdot 0.43 t_k), & 400 \leq t_k < 500 \end{cases} \quad (25)$$

In Fig. 2, we plot the time-varying spectrum results using FourierKS, OscKS, STFT, CWT, and BurgAR. The settings for estimation we use here are described in the figure captions.

Although all methods can approximate the simulated data to a good extent, FourierKS and OscKS have higher frequency resolution with less noisy representation which can help us to extract more robust features from spectro-temporal representation. Moreover, the results from FourierKS and OscKS methods are almost the same although they have different state-space models.

The computational complexities of FourierKS and OscKS are  $\mathcal{O}(N)$ , where  $N$  is the number of samples. This is because the Kalman filter and smoother that we use here

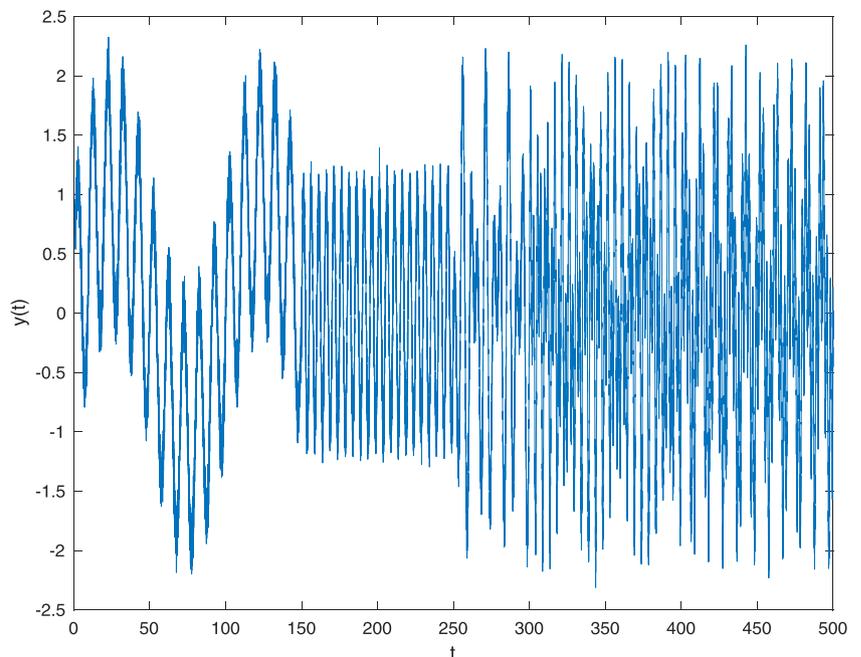
scale linearly with respect to  $N$  [40]. To numerically verify the computational efficiency of the stationary proposal in Section 2.2, we run each of the estimation methods 20 times and record the mean values of their CPU time. We test with  $\Delta t = 0.1$  and  $\Delta t = 0.01$  to control the length of the signal. The results in Table 1 clearly show that the time reduction from FourierKS (3.39 s, 9.18 s) to OscKS (0.18 s, 0.95 s) is significant. For OscKS method, the time for solving DARE is 0.09 s which accounts for almost half of the total time (0.18 s). To reduce the time usage further, one can resort to better DARE solvers or lower resolution in frequency axis. For a longer signal (i.e.  $\Delta t = 0.01$ ), OscKS (0.95 s) method becomes faster than CWT (1.32 s), which indicates a competent efficiency for long signals.

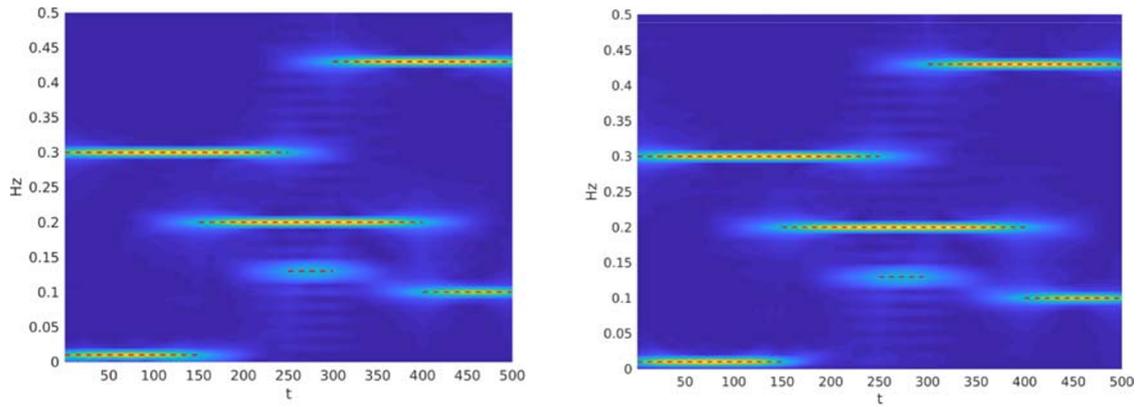
### 3 Materials and Methods for ECG Classification

#### 3.1 ECG Dataset

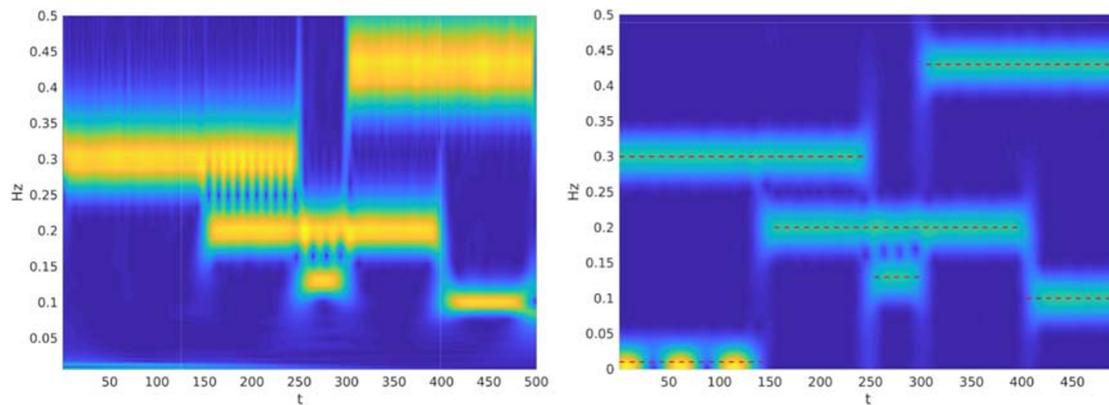
In the AF experiments, we used the ECG dataset provided by PhysioNet/CinC Challenge 2017 [8]. In total 8528 short single lead ECG recordings were collected using AliveCor hand-held devices. The recordings were uploaded automatically through an application on the user’s mobile phone. In addition, the data were sampled at 300 Hz and band-pass filtered by the AliveCor devices. The duration of ECG recordings were between 9 s to 61 s with 30 s median. The distribution of ECG recordings among different classes is as follows: Normal (5076 recordings), AF (758), Other (2415), and Noisy (279).

**Figure 1** Simulated sinusoidal data.

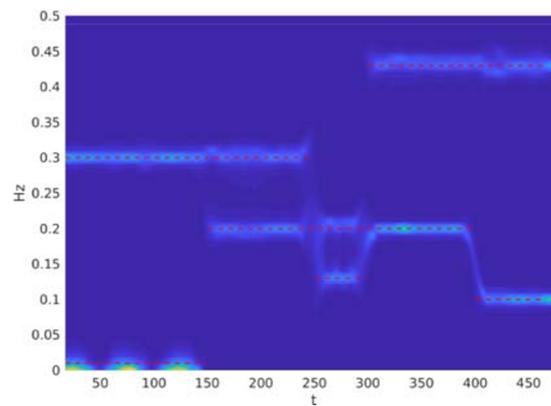




(a) Kalman  $\lambda = 0.01$ ,  $M_0 = 0$ ,  $M_{50} = 0.5$  (b) OSC  $\lambda = 0.01$ ,  $M_0 = 0$ ,  $M_{50} = 0.5$



(c) CWT Morse wavelet, time bandwidth 60 (d) STFT, Hann, window time bandwidth 350, overlap 340



(e) BurgAR, Hann window 350, overlap 340

**Figure 2** Spectro-temporal estimation on simulated data. The red dashed lines represent ground truth frequency bands.

**Table 1** CPU time cost of each spectro-temporal estimation methods. The times are recorded in a MacBook laptop with Core i5 CPU and Matlab 2017b.

	FourierKS	OscKS	CWT	STFT	BurgAR
$\Delta t = 0.1$	3.39	0.18	0.08	0.07	0.36
$\Delta t = 0.01$	9.18	0.95	1.32	0.30	2.58

### 3.2 ECG Spectro-Temporal Feature Engineering

Our aim is now to find the spectro-temporal features of ECG signals such that it can be classified by deep convolutional neural networks (CNNs). In Fig. 3 we show the overall proposed scheme from input (ECG) to output (predicted label).

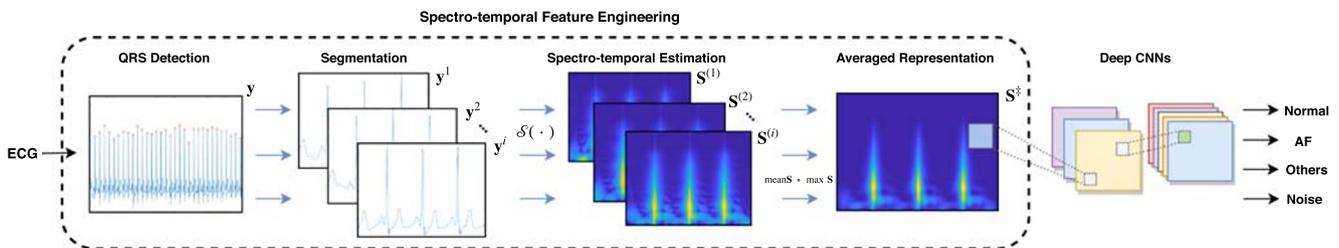
The first step is QRS detection and ECG segmentation in which the raw ECG signal is divided into fixed-length segments aligned by their central R peaks. Next, the spectro-temporal data matrix for each segment is calculated using Eq. 13. The data matrices are then averaged and normalized to generate a fixed-length spectro-temporal feature matrix. In the final step, the 2D feature matrix (spectro-temporal image) is fed into a deep CNN for classification.

The logic behind the segmentation and averaging steps in the feature engineering procedure (dashed area in Fig. 3) is threefold. First, it can handle the problem of ECG recordings with different length, and generate fixed-length spectro-temporal feature matrices. Second, it can capture enough information from ECG recording to be classified by CNNs. For example, since the central R peaks in each segments are aligned, after averaging we expect sharp edges corresponding to QRS complexes in feature matrices (spectro-temporal image) for Normal rhythms. However, for AF rhythms we expect the blurred area in spectro-temporal images due to the variable R-R intervals. For, noisy segments we do not expect any clear area for QRS complexes, and for Other classes based on the underlying arrhythmia one can expect different patterns in spectro-

temporal images (see Fig. 4). Finally, the third reason to use the segmentation and averaging steps is to decrease the effect of noise in ECG recordings. In the following we discuss different steps of feature engineering in detail.

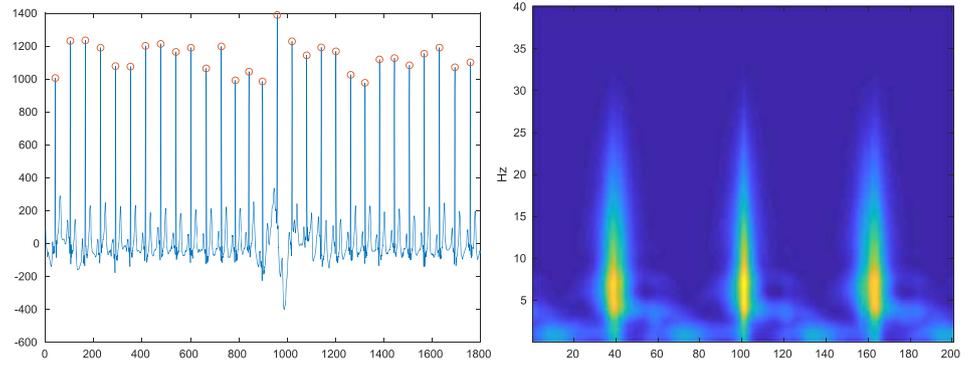
In this work, for QRS detection, we use a modified version of Pan-Tompkins algorithm. The original Pan-Tompkins algorithm [30] is sensitive to burst noise, and it easily misinterprets noise with R peak. To address this limitation at least partially, we slightly modify the original algorithm such that it iteratively checks the number of detected R peaks and if that number is smaller than a threshold, it ignores the detected R peaks and their neighbourhood samples in the ECG signal, and again applies the Pan-Tompkins algorithm on the rest of the signal. In this way, if there are few instances with high-amplitude burst noise, our algorithms can handle those. One example which illustrate this modification is shown in Fig. 5.

The next step is segmentation in which the fixed-length ECG segments are extracted from the original signal such that each segment potentially covers three QRS complexes. The segmentation process is described as follows: if  $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^T \in \mathbb{R}^N$  is the original ECG signal and  $\bar{p}_i \in \{1, 2, \dots, N\}$  is the position of  $i$ th R peak in  $\mathbf{y}$ , then  $\bar{\mathbf{p}} = [\bar{p}_1 \ \bar{p}_2 \ \dots \ \bar{p}_D]^T$  holds the positions of all R peaks, and  $D$  is the total number of R peaks in  $\mathbf{y}$ . Now, to extract  $D - 2$  ECG segments we associate each  $\bar{p}_i, i \in \{2, \dots, D - 1\}$ , to a segment of  $\mathbf{y}$  such that it potentially covers three adjacent QRS complexes. To do so, we collect  $\beta$  samples before and after each  $\bar{p}_i$ . Following this procedure, the ECG segment associated to  $i$ th R peak can be extracted from  $\mathbf{y}$  as  $\mathbf{y}^{(i)} = [y_{\bar{p}_i - \beta} \ \dots \ y_{\bar{p}_i} \ \dots \ y_{\bar{p}_i + \beta}]^T$ , and using Eq. 13, the spectro-temporal data matrix corresponding to this ECG segment is  $\mathbf{S}^{(i)} \in \mathbb{R}^{M \times (2\beta + 1)}$  where  $M$  and  $2\beta + 1$  are frequency and time steps, respectively. It is worth noticing that these two parameters (i.e.,  $M$  and  $2\beta + 1$ ) determine the size of the matrix  $\mathbf{S}$  in Eq. 13. The choice of parameter  $\beta$  is important, as it regulates the length of output and how much takes into average. Usually,  $\beta$  should cover at least three QRS complexes for good evidence of R-R intervals.



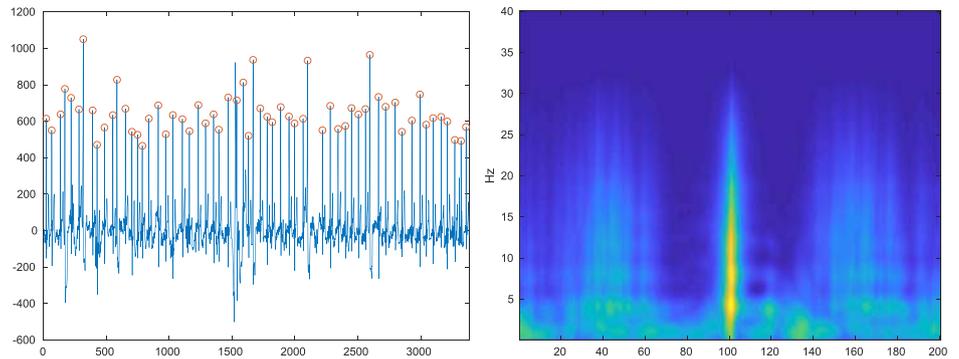
**Figure 3** Generalized overall processing scheme for ECG analysis.

**Figure 4** Results of representation averaging (right side) on four types of ECG signals (left side), using proposed spectro-temporal method. Red circles indicate detected R peaks.



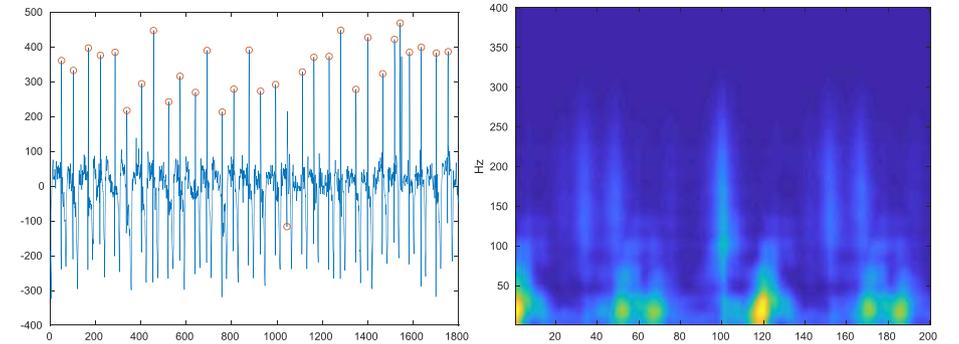
(a) Rec. 5569 (Normal)

(b) Rec. 5569 (Normal)



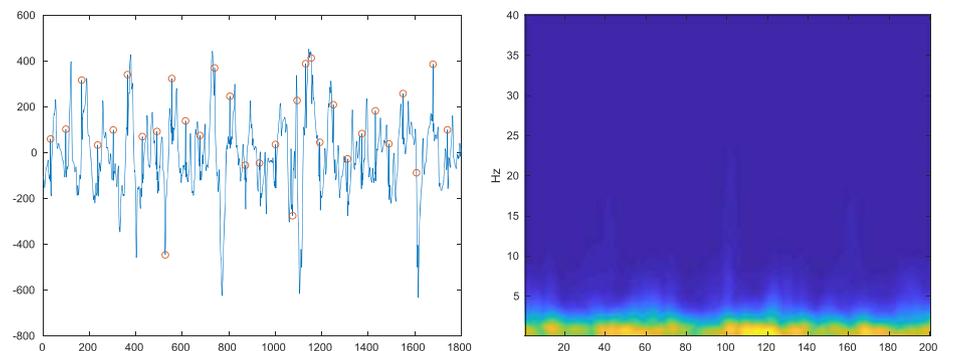
(c) Rec. 5587 (Atrial Fibrillation)

(d) Rec. 5587 (Atrial Fibrillation)



(e) Rec. 5586 (Others)

(f) Rec. 5586 (Others)



(g) Rec. 5507 (Noise)

(h) Rec. 5507 (Noise)

**Algorithm 1** Averaging representation.

**Input:** Signal  $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$   
**Output:** Spectro-temporal feature  $\mathbf{S}^\ddagger \in \mathbb{R}^{M \times (2\beta+1)}$

- 1: Perform Pan-Tompkins on  $\mathbf{y}$  and obtain  $\bar{\mathbf{p}}$
- 2:  $D = \text{size}(\bar{\mathbf{p}})$
- 3: **if**  $D \leq \delta$  **then**
- 4:      $\bar{\mathbf{y}} = \mathbf{y}$
- 5:      $\bar{y}_{i-\alpha:i+\alpha} = 0$  for all  $i \in \bar{\mathbf{p}}$
- 6:     Perform Pan-Tompkins on  $\bar{\mathbf{y}}$ , and obtain new  $\bar{\mathbf{p}}, D$
- 7: **end if**
- 8: **for all**  $1 < i < D$  **in**  $\bar{\mathbf{p}}$  **do**
- 9:     Perform Spectro-temporal estimation on  $\mathbf{y}^{(i)}$  to get  $\mathbf{S}^{(i)}$
- 10: **end for**
- 11: **return**  $\mathbf{S}^\ddagger = \frac{\sum_{i=2}^{D-1} \mathbf{S}^{(i)}}{D-2} \circ \max_{2 \leq i \leq D-1} \mathbf{S}^{(i)}$

The spectro-temporal feature matrix  $\mathbf{S}^\ddagger$  is obtained by averaging over all spectro-temporal data matrices and multiplying with their maximum mask:

$$\mathbf{S}^\ddagger = \frac{\sum_{i=2}^{D-1} \mathbf{S}^{(i)}}{D-2} \circ \max_{2 \leq i \leq D-1} \mathbf{S}^{(i)}. \tag{26}$$

The reason for adding a max operation in Eq. 26 is that it could, at least in certain extent, help preserving intricate details of spectro-temporal data that were potentially lost during averaging across every segments, and also normalizing the data.

Examples of ECG spectro-temporal feature matrices (images) four different classes of ECG signals are shown in Fig. 4, where we used the proposed spectro-temporal estimation method in Section 2.2.

**3.3 Classification**

In the recent ten years, deep learning techniques, especially convolutional neural networks, have achieved great success in detection and classification tasks. Comparing to 1D CNNs models, the progress of CNNs for 2D image applications is more prosperous. The aim here is to leverage advanced CNNs for AF classification using the time-varying spectrum (which is an image).

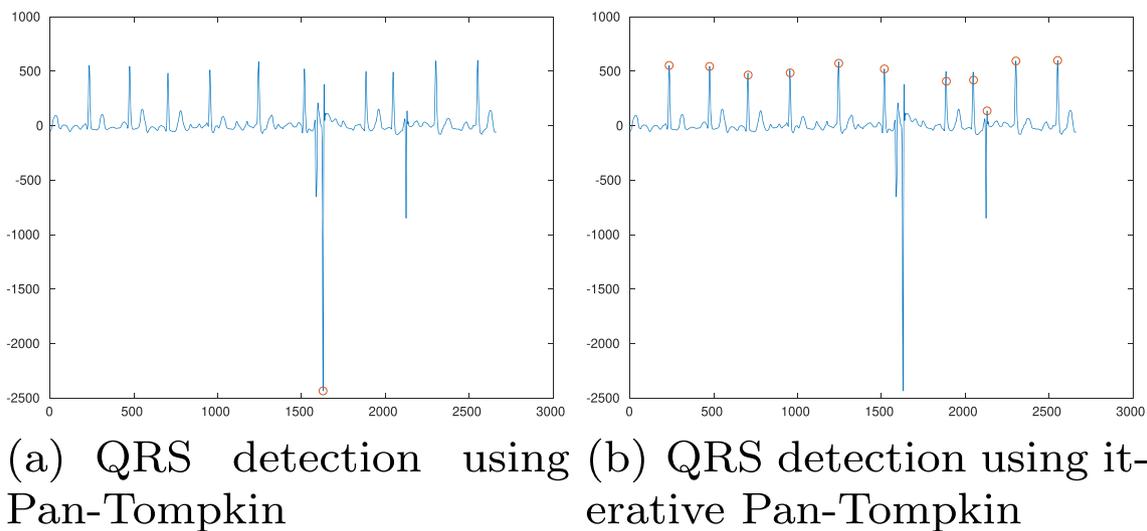
However, one flaw in most of the current network models is that the information during training, principally the gradient, may disappear if the network is exceedingly deep (with many layers), which is usually called “vanishing gradient” [12]. In general way, this root problem can be alleviated by several basic ways, for instance, with pre-training, residual connection, or with properly selected activation functions (e.g., one should not attach ReLu before batch normalization).

Densely connected convolutional networks (DenseNet) [20], which won the 2017 best paper award of CVPR, provide state-of-the-art performance without degradation or over-fitting even when stacked by hundred of layers. DenseNets can be seen as refined versions of deep residual networks (ResNets) [18], where the former one introduces explicit connection on every two and preceding layers in a dense block rather than only adjacent layers, as shown in Fig. 6. Another additional advantage of DenseNet, as mentioned in [20], is the feature reuse.

Considering an  $L$  layers network, and image input  $\mathbf{U}_0$ , the output of  $l$ -th layer is:

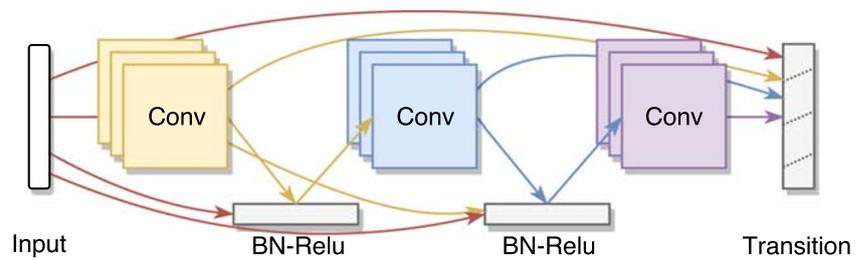
$$\mathbf{U}_l = H_l^{Res}(\mathbf{U}_{l-1}) + \mathbf{U}_{l-1}, \tag{27}$$

$$\mathbf{U}_l = H_l^{Den}([\mathbf{U}_0 \ \mathbf{U}_1 \ \dots \ \mathbf{U}_{l-1}]). \tag{28}$$



**Figure 5** Improvement in QRS detection.

**Figure 6** Dense block: each of the convolutional layer takes all of their preceding outputs as input.



where  $H_l^{Res}$  and  $H_l^{Den}$  are layer operations (e.g., convolution, batch-normalization, or activation) of ResNet and DenseNet respectively, and  $\mathbf{U}_l$  is the output of  $l$ th layer.

The DenseNet we implement here, which we refer as Dense18<sup>+</sup>, is slightly different from the original proposal [20], where we employ and concatenate both max and average global pooling on last layer before fully connected layer, as shown in Table 2. The motivation of this is to try to alleviate the information loss problem caused by pooling operation [38]. In our application, because of the size of input, we remove the initial down-sampling max pooling

layer. Each dense block contains four  $3 \times 3$  convolutional layers, with growth rate of 48 and reduction rate 0.5.

### 3.4 Model Assessment and Evaluation Criteria

To evaluate the performance of the proposed methods, we have conducted experiments on the ECG dataset described in Section 3.1. The classification performance of different methods was assessed by using the scoring mechanism recommended by PhysioNet/Computing in Cardiology (CinC) Challenge 2017 [8] over the whole dataset in 10-fold

**Table 2** Structure of Dense18<sup>+</sup> in this paper. The use of eighteen layers in this application is enough, as the input dimensionality is not significantly large and the training of deeper network is more challenging [26]. The use of 18 layers is also the best result from our hyper-parameter tuning stage.

Layer Name	Structure	Output Size
Input	Input	(50, 50, 1)
Convolution stride 1	$7 \times 7$ conv (50, 50, 64)	
Dense Block 1	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$	(50, 50, 256)
Transition 1	$1 \times 1$ conv (25, 25, 128)	
Dense Block 2	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$	(25, 25, 320)
Transition 2	$1 \times 1$ conv (12, 12, 160)	
Dense Block 3	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$	(12, 12, 352)
Transition 3	$1 \times 1$ conv (6, 6, 176)	
Dense Block 4	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$	(6, 6, 368)
Pooling		
Concatenate	$\begin{bmatrix} \text{global ave} \\ \text{global max} \end{bmatrix}$ concat	(736)
Fully Connected (Softmax)	4 classes	(4)

cross-validation scheme. The data were partitioned such that the same proportions of each class are available in each fold (stratified cross-validation). Moreover, the F1 score,

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{29}$$

for each class is calculated to summarize the performance of that specific class: Normal ( $F1_N$ ), AF ( $F1_A$ ), Others ( $F1_O$ ), and Noisy ( $F1_{\sim}$ ). Then, as recommended by PhysioNet/CinC 2017 the overall evaluation metric is used as follows:

$$F1_{\text{overall}} = \frac{1}{3}(F1_N + F1_A + F1_O). \tag{30}$$

Finally, the detailed performance is shown by a 4-class confusion matrix whose the diagonal entries are the correct classifications and the off-diagonal entries are the incorrect classifications. This confusion matrix is the result of stacking 10 confusion matrices of the test data in the 10-fold cross-validation.

### 4 Experiments

In principle, any time-frequency analysis method can be used for ECG classification. So, in order to show the benefit of using the proposed spectro-temporal method in Section 2 over other standard time-frequency analysis methods, we have conducted experiments on the ECG dataset. We have compared the results of the proposed method with short-time Fourier transform (STFT), continuous wavelet transform (CWT), and classical power spectral density estimation method. To do so, we used magnitude of STFT, magnitude of CWT, and square root of non-logarithmic power spectral density using Burg autoregressive model (BurgAR) [23] of ECG signal to construct the feature matrices. The settings for spectro-temporal estimation are the same as in Section 5.1. All spectro-temporal feature

matrices (images) are then unifiedly resized (down-sample by local averaging) to  $50 \times 50$  for classifiers.

For the random forest we use 500 decision trees and random selection of 50 features (out of 2500) at each node. In addition, at each node the random forest minimizes the cross-entropy impurity measure.

Different convolutional architectures are examined, and their results are compared to the standard RF classifier. Here we take the original implementations of InceptionV3, ResNet18, ResNet34, and DenseNet from papers [18, 20, 46] without modifications, except that we removed the initial sub-sampling layer. We also construct a plain 18-layer CNN (CNN18) which has the same structure as Dense18<sup>+</sup> but without the dense connections. As to the DenseNet, we only show the results of using 18 layers, because we previously failed to get better results with deeper structures in a hyper-parameter tuning stage. The deep CNNs in Table 3 are all equally trained using Adam optimizer with learning rate 1e-3, weight decay 1e-3, and 60 epochs. The batch normalization is also enabled with batch size 128. In addition, we also performed a hyper-parameter engineering, for example, for the training parameters and the number of layers.

With seven classifiers and five different time-frequency analysis methods, in total we have 35 different combinations whose performance are reported in Table 3. As can be seen from this table the best results (overall scores) belong to our proposed spectro-temporal representation methods (i.e., FourierKS and OscKS) with Dense18<sup>+</sup> classifier. Moreover, Table 4 shows the performance for each ECG classes for Dense18<sup>+</sup> classifier with different time-frequency representation.

The detailed performance of all five methods (i.e., FourierKS, OscKS, CWT, STFT, and BurgAR) with Dense18<sup>+</sup> classifier are reported in five confusion matrices in Fig. 7. Each confusion matrix is row-wise normalized. The diagonal entries show the Recall of each rhythm and off-diagonal entries show the misclassification rates. For

**Table 3** 10-fold cross-validation F1 Score of spectro-temporal estimation methods using different classifiers for classification. Best score for each column and row are rendered bold and italic respectively. The number of parameters of InceptionV3, ResNet18, ResNet34, DenseNet18, and Dense18<sup>+</sup> are 21.81M, 5.08M, 9.59M, 1.58M, and 1.58M, respectively.

<i>F1<sub>overall</sub></i>	Random Forest[5]	CNN18	InceptionV3[46]	ResNet18[18]	ResNet34[18]	DenseNet18[20]	Dense18 <sup>+</sup>
STFT	73.47	72.65	75.66	76.17	76.26	77.39	<i>77.67</i>
CWT	74.91	<b>73.96</b>	76.41	<b>78.57</b>	<b>78.70</b>	78.82	<i>79.63</i>
BurgAR	73.22	71.78	76.45	76.41	76.30	77.58	<i>77.76</i>
FourierKS	75.99	72.74	<b>77.48</b>	78.05	77.99	79.50	<b>80.24</b>
OscKS	<b>76.12</b>	73.07	76.91	77.85	78.19	<b>79.67</b>	<i>80.18</i>

**Table 4** 10-fold cross-validation results of overall and four labels using different spectro-temporal estimation methods on Dense18<sup>+</sup> classifier. Best score for each column are rendered bold.

	Method	$F1_N$	$F1_A$	$F1_O$	$F1_{\sim}$	$F1_{overall}$	$Std_{F1}$
(1)	STFT + Dense18 <sup>+</sup>	88.67	74.49	69.84	53.28	77.67	1.78
(2)	CWT + Dense18 <sup>+</sup>	<b>89.30</b>	77.76	71.82	51.95	79.63	1.76
(3)	BurgAR + Dense18 <sup>+</sup>	88.35	75.17	69.74	<b>56.49</b>	77.76	1.62
(4)	Kalman + Dense18 <sup>+</sup>	89.29	79.18	<b>72.25</b>	52.50	<b>80.24</b>	<b>1.52</b>
(5)	OSC + Dense18 <sup>+</sup>	89.09	<b>79.78</b>	71.68	55.86	80.18	1.55
(6)	Martin [55]	88.8	76.4	72.6	64.5	79.2	N/A
(6)	Zhaohan [51]	87	80	68	N/A	78	N/A

example, the first row of the first confusion matrix shows 92.1% of normal rhythms are correctly classified as normal, but 0.6%, 6.3%, and 1.0% are incorrectly classified as AF, Other, and Noisy.

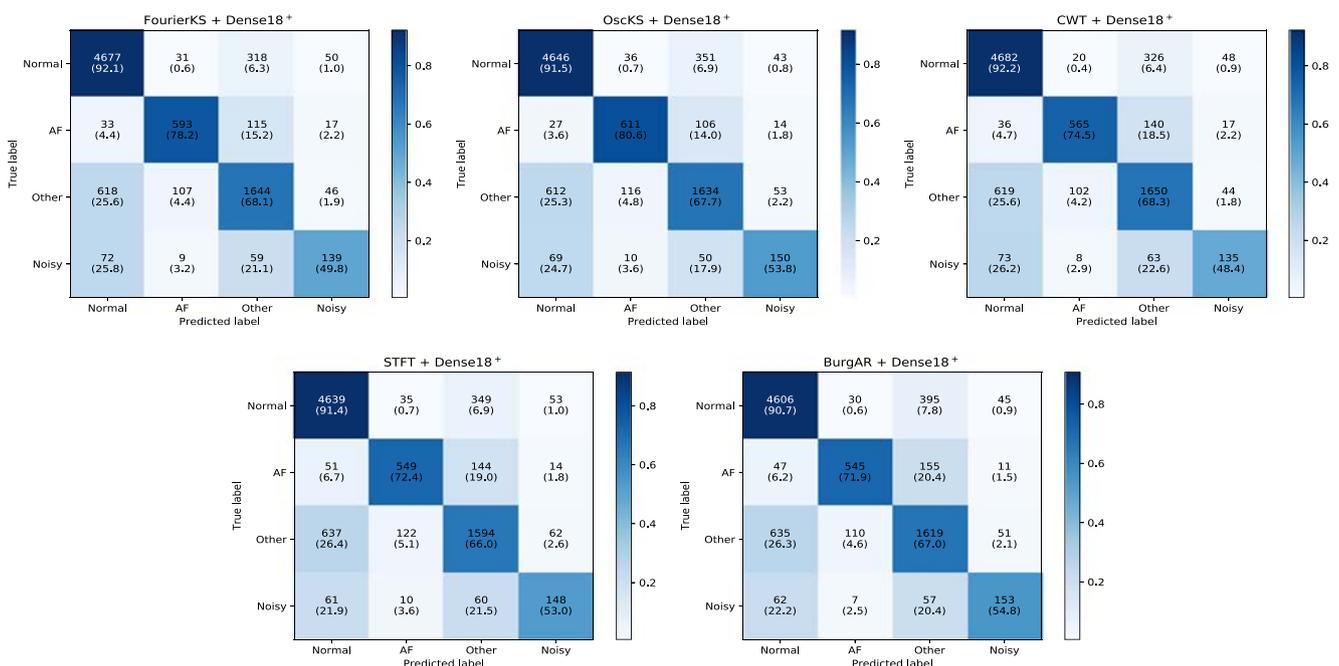
## 5 Discussion

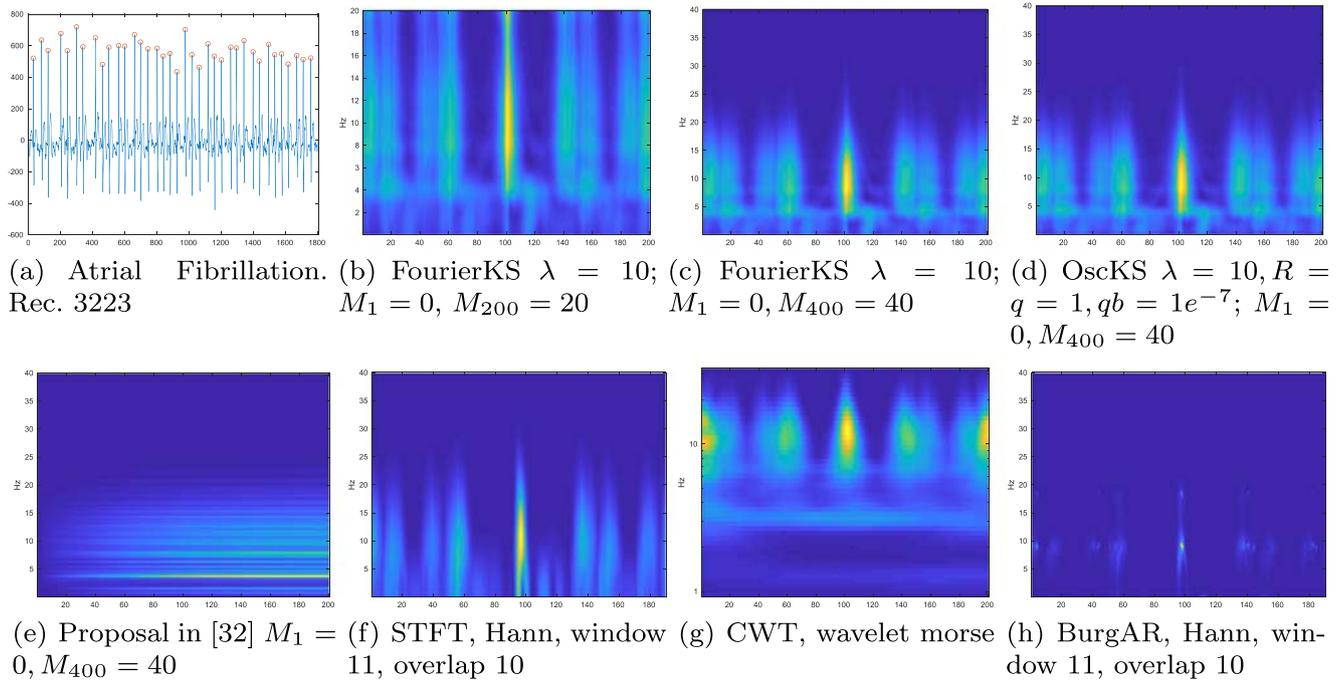
### 5.1 ECG Time-Frequency Analysis Methods

We first examine how different spectro-temporal estimation methods perform on an ECG signal through a visual inspection. We take the 3223th recording (Rec. 3223) from CinC 2017 dataset as example, which is labelled as AF. It is shown in Fig. 8a. For the FourierKS and OscKS method, we choose different frequency range ( $M$ ) and smoothing option as shown in Fig. 8b, c and d. We set the length scale  $\lambda$  to

a constant 10, and use 1 for variance of measurement noise  $R$ , and identity for covariance of process noise  $q$ . In theory,  $\lambda$  could be different for each frequency, which could be used to improve the performance. Figure 8e presents results by the original method in [32], which adopts Brownian motion model for the coefficients. For STFT and BurgAR, we apply 11 length 10 overlapping Hann windows for estimation, as shown in Figs. 8f and h. For CWT (Fig. 8g), we use the default Morse wavelet implemented in Matlab.

First, we observe that the estimation results of FourierKS (Fig. 8c) and OscKS (Fig. 8d) are nearly the same except that the base frequency  $a_0$  coefficient estimates are very sensitive to  $q_b$  in the OscKS method. If we compare FourierKS method to STFT, BurgAR, and CWT, which are shown in Fig. 8c, f, h, and g respectively, we can initially conclude several advantages: the result from FourierKS is more smooth and it has higher and more unified

**Figure 7** Normalized confusion matrix on different methods.



**Figure 8** Comparison of different spectrogram estimation methods on Rec. 3223.

resolution on both time and frequency. For STFT and BurgAR, the resolution is confined by window selection, length, and overlap. CWT untangles this problem by scaling and translation of wavelet basis function, but due to uncertainty principle of wavelet signal processing [36], the required resolution in time and frequency can not be met simultaneously (see Fig.8g). Our approaches model the time-varying Fourier series coefficients of signal in state-space, which are free from usage of windows or wavelets.

Another advantage of the proposed OscKS estimation method is that it can be very computationally efficient for implementation when we need to perform estimation many times and the system is fixed (i.e.,  $\mathbf{A}, \mathbf{Q}$  remain unchanged). For example, if one takes the averaging strategy, the spectrum estimation has to be done for every segment and recording. For OscKS method, we merely need to solve  $\mathbf{P}_\infty$  in Eq. 20 once. As we stated in Section 2.2, the computational cost of OscKS method is substantial reduced by deriving a stable covariance.

### 5.2 ECG Classification for AF Detection

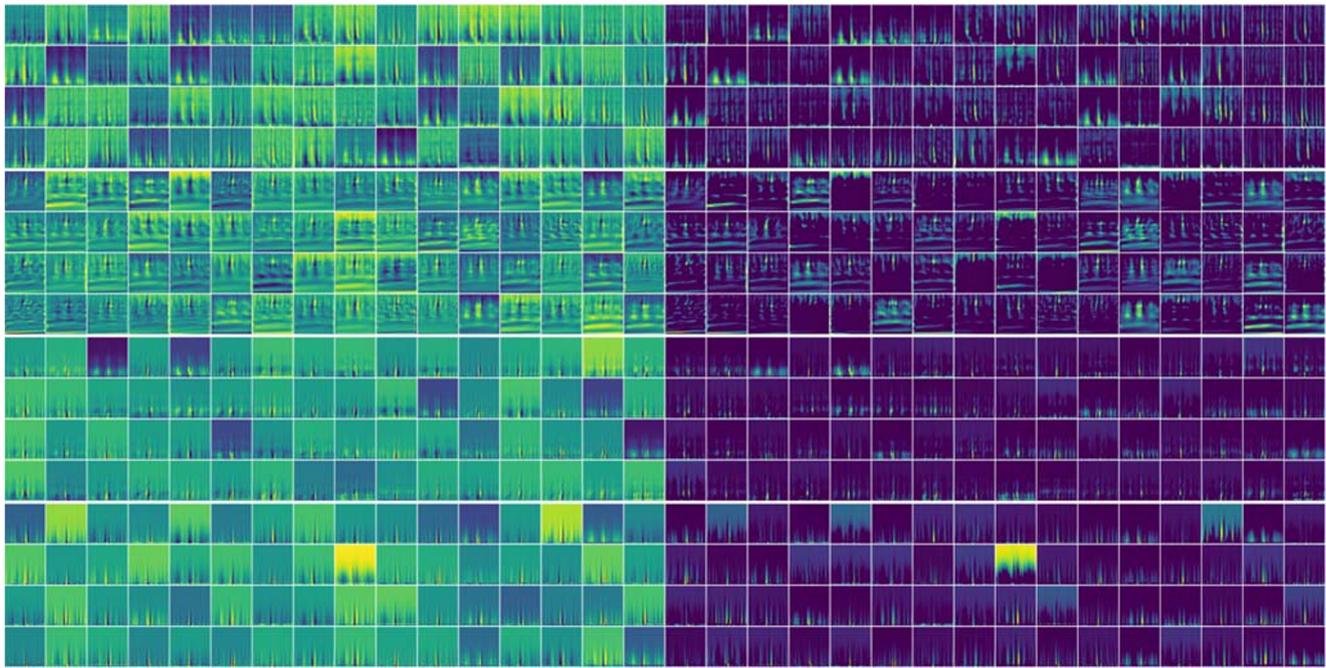
As it is mentioned before, Table 3 shows that the best results belong to our proposed spectro-temporal representation methods (i.e., FourierKS and OscKS) with Dense18<sup>+</sup> classifier. Table 3 also shows that independent of spectro-temporal representation method, Dense18<sup>+</sup> has the highest performance among all classifiers. In contrast, the plain CNN (CNN18) has the lowest scores. In

addition, RF is generally worse than convolutional networks classifiers (except CNN18) probably because in contrary to convolutional networks, RF has not benefited from the existing structure in spectro-temporal representation.

Regarding the different spectro-temporal representations STFT and BurgAR have the worst results, and FourierKS, and OscKS have the best performance. In addition, for some classifiers CWT provides the results which are as good as or even better than FourierKS, and OscKS. However, the best results of FourierKS, and OscKS are higher than the best result of CWT.

Table 4 shows that the the proposed ECG classification methods have the best result for Normal rhythm and the worst result for Noisy. The performance of AF and Other are between these two, but typically AF has better performance than Other, probably because Other is an umbrella term that covers many abnormal non-AF rhythms, and we do not have enough samples for each abnormalities to properly train our classifiers.

To examine how different spectro-temporal features act in AF ECG analysis, one elementary-level way is to investigate the feature map and activation of the first convolutional layer. However, this voxel-based “probing” only produces limited explanation [47], and can not fully give the insights. The visualization is shown in Fig. 9. We can see that the feature-map of FourierKS and CWT are more diverse and active than STFT and BurgAR, and they have larger activation on “peaks” and background details. In comparison to FourierKS and CWT, the lower-frequency



**Figure 9** Feature-map (Left 16 columns) and activation (right 16 columns) visualization of first convolutional layer on Rec. 1005 (AF). From top to bottom, every 4 rows are FourierKS, CWT, BurgAR and

STFT respectively. OscKS is not shown here for simplicity, because it has a very similar result to FourierKS.

area are better preserved and exploited for FourierKS method.

### 5.3 Limitations

Typically for AF detection we need at least 30 s ECG data [7]. However, many ECG recordings in the dataset have less than 30 s duration (see Section 3.1) which limits the medical significance of the current study. In addition, the averaging step in feature engineering is robust only when there are enough spectro-temporal segments, which is not the case for very short ECG recordings (see Section 3.2).

## 6 Conclusion

In this paper, we proposed a spectro-temporal representation of ECG signals, based on state-space models, for application in deep network based atrial fibrillation detection. We empirically showed that if we put Gaussian process priors on the Fourier series coefficients, then by estimating the state of the corresponding linear state-space model using Kalman filter/smoother we can outperform other time-frequency analysis methods such as short-time Fourier transform, continuous wavelet transform, and autoregressive spectral estimation for ECG classification.

We also accelerated the estimation of the spectro-temporal representation of signals by using a stochastic

oscillator differential equation model and stationary Kalman filter/smoother. This representation is useful to improve the scalability of the proposed spectro-temporal representation for long ECG recordings. Finally, we have found an efficient convolutional architecture (i.e., Dense18<sup>+</sup>) for AF detection using the spectro-temporal features by comparative evaluation of multiple convolutional neural networks models.

**Acknowledgments** Open access funding provided by Aalto University.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A., Tan, R.S. (2017). A deep convolutional neural

- network model to classify heartbeats. *Computers in Biology and Medicine*, 89, 389–396.
2. Annavarapu, A., & Kora, P. (2016). ECG-Based atrial fibrillation detection using different orderings of conjugate symmetric-complex Hadamard transform. *International Journal of the Cardiovascular Academy*, 2(3), 151–154.
  3. Asgari, S., Mehrmia, A., Moussavi, M. (2015). Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. *Computers in Biology and Medicine*, 60, 132–142.
  4. Babaeizadeh, S., Gregg, R.E., Helfenbein, E.D., Lindauer, J.M., Zhou, S.H. (2009). Improvements in atrial fibrillation detection for real-time monitoring. *Journal of Electrocardiology*, 42(6), 522–526.
  5. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
  6. Bruser, C., Diesel, J., Zink, M.D., Winter, S., Schauerte, P., Leonhardt, S. (2013). Automatic detection of atrial fibrillation in cardiac vibration signals. *IEEE Journal of Biomedical and Health Informatics*, 17(1), 162–171.
  7. Camm, A.J., Kirchhof, P., Lip, G.Y., Schotten, U., Savelieva, I., Ernst, S., Van Gelder, I.C., Al-Attar, N., Hindricks, G., Prendergast, B., et al. (2010). Guidelines for the management of atrial fibrillation: the task force for the management of atrial fibrillation of the european society of cardiology (ESC). *European Heart Journal*, 31(19), 2369–2429.
  8. Clifford, G.D. et al. (2017). AF classification from a short single lead ECG recording: the Physionet/Computing in Cardiology Challenge 2017. *2017 Computing in Cardiology (CinC)*, 44, 1–4.
  9. Datta, S., Puri, C., Mukherjee, A., Banerjee, R., Choudhury, A.D., Singh, R., Ukil, A., Bandyopadhyay, S., Pal, A., Khandelwal, S. (2017). Identifying normal, AF and other abnormal ECG rhythms using a cascaded binary classifier. In *2017 Computing in Cardiology (cinc)*, Vol. 44.
  10. Ehrendorfer, M. (2011). Spectral numerical weather prediction models society for industrial and applied mathematics.
  11. García, M., Ródenas, J., Alcaraz, R., Rieta, J.J. (2016). Application of the relative wavelet energy to heart rate independent detection of atrial fibrillation. *Computer Methods and Programs in Biomedicine*, 131, 157–168.
  12. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, (Vol. 9 pp. 249–256).
  13. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y. (2016). *Deep learning*. Cambridge: MIT Press.
  14. Grewal, M.S., & Andrews, A.P. (2001). *Kalman filtering, theory and practice using MATLAB*. New York: Wiley.
  15. Hagiwara, Y., Fujita, H., Oh, S.L., Tan, J.H., Tan, R.S., Ciaccio, E.J., Acharya, U.R. (2018). Computer-aided diagnosis of atrial fibrillation based on ECG signals: a review. *Information Sciences*, 467, 99–114.
  16. Hannun, A.Y., Rajpurkar, P., Haghpanahi, M., Tison, G.H., Bourn, C., Turakhia, M.P., Ng, A.Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1), 65.
  17. Hartikainen, J., & Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE International workshop on machine learning for signal processing (MLSP)* (pp. 379–384).
  18. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on computer vision and pattern recognition (CVPR)*, Vol. 770–778.
  19. Hong, S., Wu, M., Zhou, Y., Wang, Q., Shang, J., Li, H., Xie, J. (2017). ENCASE: An ENsemble cLASSifier for ECG classification using expert features and deep neural networks. In *2017 Computing in Cardiology (cinc)*, Vol. 44.
  20. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on computer vision and pattern recognition (CVPR)* (pp. 2261–2269).
  21. Joseph, A., Larrain, M., Turner, C. (2017). Daily stock returns characteristics and forecastability. *Procedia Computer Science*, 114, 481–490.
  22. Kailath, T., Sayed, A.H., Hassibi, B. (2000). *Linear estimation*. New Jersey: Prentice Hall.
  23. Kay, S.M., & Marple, S.L. (1981). Spectrum analysis—a modern perspective. *Proceedings of the IEEE*, 69(11), 1380–1419.
  24. Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, (Vol. 25 pp. 1097–1105). Brooklyn: Curran Associates Inc.
  25. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
  26. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems 31* (pp. 6389–6399). Brooklyn: Curran Associates Inc.
  27. Mathews, S.M., Kambhamettu, C., Barner, K.E. (2018). A novel application of deep learning for single-lead ecg classification. *Computers in Biology and Medicine*, 99, 53–62.
  28. Mohebbi, M., & Ghassemian, H. (2008). Detection of atrial fibrillation episodes using SVM. In *2008 30th annual international conference of the IEEE engineering in medicine and biology society* (pp. 177–180): IEEE.
  29. Moody, G.B., & Mark, R.G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3), 45–50.
  30. Pan, J., & Tompkins, W.J. (1985). A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering BME*, 32(3), 230–236.
  31. Pourbabaee, B., Roshtkhari, M.J., Khorasani, K. (2018). Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems Man, and Cybernetics: Systems*, 48(12), 2095–2104.
  32. Qi, Y., Minka, T.P., Picara, R.W. (2002). Bayesian spectrum estimation of unevenly sampled nonstationary data. In *2002 IEEE International conference on acoustics, speech, and signal processing (ICASSP)*, (Vol. 2 pp. 1473–1476): IEEE.
  33. Rad, A.B., & Virtanen, T. (2012). Phase spectrum prediction of audio signals. In *2012 5Th international symposium on communications, control and signal processing*, pp. 1–5. IEEE.
  34. Rad, A.B., et al. (2017). ECG-Based classification of resuscitation cardiac rhythms for retrospective data analysis. *IEEE Transactions on Biomedical Engineering*, 64(10), 2411–2418.
  35. Rajpurkar, P., Hannun, A.Y., Haghpanahi, M., Bourn, C., Ng, A.Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv:1707.01836.
  36. Ricaud, B., & Torrèsani, B. (2014). A survey of uncertainty principles and some signal processing applications. *Advances in Computational Mathematics*, 40(3), 629–650.
  37. Rubin, J., Parvaneh, S., Rahman, A., Conroy, B., Babaeizadeh, S. (2017). Densely connected convolutional networks and signal quality analysis to detect atrial fibrillation using short

- single-lead ECG recordings. In *2017 Computing in cardiology (cinc)* (pp. 1–4).
38. Sabour, S., Frosst, N., Hinton, G.E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems 30* (pp. 3856–3866). Brooklyn: Curran Associates Inc.
  39. Sannino, G., & Pietro, G.D. (2018). A deep learning approach for ECG-based heartbeat classification for arrhythmia detection. *Future Generation Computer Systems*, 86, 446–455.
  40. Särkkä, S. (2013). *Bayesian filtering and smoothing*. Cambridge: Cambridge University Press.
  41. Särkkä, S., Solin, A., Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, 30(4), 51–61.
  42. Särkkä, S., Solin, A., Nummenmaa, A., Vehtari, A., Auranen, T., Vanni, S., Lin, F.H. (2012). Dynamic retrospective filtering of physiological noise in BOLD fMRI: DRIFTER. *NeuroImage*, 60(2), 1517–1527.
  43. Shashikumar, S.P., Shah, A.J., Li, Q., Clifford, G.D., Nemati, S. (2017). A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology. In *2017 IEEE EMBS International conference on biomedical health informatics (BHI)* (pp. 141–144): IEEE.
  44. Solin, A., & Särkkä, S. (2014). Explicit link between periodic covariance functions and state space models. In Kaski, S., & Corander, J. (Eds.) *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research*, (Vol. 33 pp. 904–912). PMLR, Reykjavik, Iceland.
  45. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of AAAI on Artificial Intelligence* (pp. 4278–4284).
  46. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
  47. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2013). Intriguing properties of neural networks. arXiv:1312.6199.
  48. Teijeiro, T., García, C.A., Castro, D., Félix, P. (2017). Arrhythmia classification from the abductive interpretation of short single-lead ECG records. In *2017 Computing in cardiology (cinc)*, Vol. 44.
  49. Thaler, M. (2017). *The only EKG book you'll ever need*. Philadelphia: Lippincott Williams & Wilkins.
  50. Xia, Y., Wulan, N., Wang, K., Zhang, H. (2018). Detecting atrial fibrillation by deep convolutional neural networks. *Computers in Biology and Medicine*, 93, 84–92.
  51. Xiong, Z., Stiles, M.K., Zhao, J. (2017). Robust ECG signal classification for detection of atrial fibrillation using a novel neural network. *2017 Computing in Cardiology (CinC)*, 44, 1–4.
  52. Yaghoubi, F., Ayatollahi, A., Bahramali, R., Yaghoubi, M., Alavi, A.H. (2010). Towards automatic detection of atrial fibrillation: a hybrid computational approach. *Computers in Biology and Medicine*, 40(11), 919–930.
  53. Zabihi, M., Rad, A.B., et al. (2017). Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier. *2017 Computing in Cardiology (CinC)*, 44, 1–4.
  54. Zhao, Z., Särkkä, S., Rad, A.B. (2018). Spectro-temporal ECG analysis for atrial fibrillation detection. In *2018 IEEE 28th international workshop on machine learning for signal processing (MLSP)* (pp. 1–6): IEEE.
  55. Zihlmann, M., & Perekrestenko, D. (2017). Tschannen, M.: Convolutional recurrent neural networks for electrocardiogram classification. *2017 Computing in Cardiology (CinC)*, 44, 1–4.
  56. Zoni-Berisso, M., Lercari, F., Carazza, T., Domenicucci, S. (2014). Epidemiology of atrial fibrillation: European perspective. *Clinical Epidemiology*, 6, 213–220.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.