Biomedical Signal Processing and Control 63 (2021) 102200

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Prediction of cardiac arrhythmia using deterministic probabilistic finite-state automata

Zhi Li [a,*], Harm Derksen [b], Jonathan Gryak [a,c], Cheng Jiang [g], Zijun Gao [a], Winston Zhang [a], Hamid Ghanbari [e], Pujitha Gunaratne [h], Kayvan Najarian [a,c,d,f,g]

[a] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA
[b] Department of Mathematics, University of Michigan, Ann Arbor, MI, USA
[c] Michigan Institute for Data Science, University of Michigan, Ann Arbor, MI, USA
[d] Department of Emergency Medicine, University of Michigan, Ann Arbor, MI, USA
[e] Department of Internal Medicine - Cardiology, University of Michigan, Ann Arbor, MI, USA
[f] Michigan Center for Integrative Research in Critical Care, University of Michigan, Ann Arbor, MI, USA
[g] Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA
[h] Toyota Motor North America, Ann Arbor, MI, USA

ARTICLE INFO

Keywords:
Signal processing
Machine learning
Probabilistic finite-state automata
Cardiac arrhythmia prediction

ABSTRACT

This paper introduces a novel method for classifying and predicting cardiac arrhythmia events via a special type of deterministic probabilistic finite-state automata (DPFA). The proposed method constructs the underlying state space and transition probabilities of the DPFA model directly from the input data. The algorithm was employed in the prediction of two types of cardiac events, supraventricular tachycardia (SVT) and atrial high-rate episodes (AHRE), with its performance compared to five other well-established methods. In all experiments, the proposed method achieved over 0.8 AUC for both SVT and AHRE prediction.

1. Introduction

The electrocardiogram (ECG) is a type of physiological signal with many clinical applications in cardiac arrhythmia, including diagnosis confirmation, monitoring drug effects, and rate control [1]. Despite its great value, it is often challenging to extract relevant information from ECG signals, even for a well-trained medical expert, due to variability in signal morphology. Such variability is ubiquitous and manifests itself in different ways: via the ECG signals themselves, the measurements derived from said signals, and the diagnostic interpretations based on such measurements [2].

Most traditional methods for physiological signal analysis depend heavily on pre-processing to identify peaks with specific morphology. As a result, these methods tend to be less effective on noisy data. In recent years, with the advent of portable arrhythmia monitoring devices, it has become possible to collect ECG data in real time, albeit with more noise. Therefore, in order to enable automated clinical decision making using such ECG sources, it is desirable to introduce new methods that require minimal pre-processing prior to analysis.

In this paper, we introduce a novel deterministic probabilistic finite-state automata (DPFA)-based approach for classifying and predicting cardiac events using ECG signals. The method transforms ECG signals into probabilistic strings over an alphabet which, in this study, contains only three symbols. Each of the three symbols corresponds to a broad morphology type which is present in ECG signals. Using these symbolized ECGs, the algorithm seeks to extract information related to the patterns and sequences of the symbols contained in the probabilistic string. This is achieved by constructing a DPFA from the probabilistic string, after which the DPFA serves as the primary means of classification. This approach frees the resultant classification algorithm from relying on the identification of specific ECG morphological such as peaks.

In order to test its performance, the algorithm was used to predict atrial high-rate episodes (AHRE), a surrogate for atrial fibrillation (AFib), and supraventricular tachycardia (SVT) events by analyzing ECG data several minutes prior to the onset of the cardiac event. In most cases the algorithm outperforms five other more traditional and well-established approaches, including a heart rate variability (HRV) based method using support vector machine (SVM), a method combining discrete wavelet transform (DWT) and principal component analysis (PCA), a deep neural network (DNN), a convolutional neural network (CNN), and a CNN with long-short term memory (LSTM).

* Corresponding author.
E-mail address: zcli@umich.edu (Z. Li).

https://doi.org/10.1016/j.bspc.2020.102200
Received 13 January 2020; Received in revised form 25 August 2020; Accepted 29 August 2020
Available online 2 September 2020

2. Historical perspectives and current trends

Arrhythmia classification using ECG signals typically involves three major tasks: signal pre-processing in order to remove noise and baseline wandering, feature extraction in time and/or frequency domains, and classification of the signals into different arrhythmia types according to the features extracted, often through the training of classifiers via machine learning algorithms.

In the literature, numerous feature extraction techniques have been proposed to analyze and classify ECG signals. These techniques include: heart rate variability (HRV) [3,4], independent component analysis (ICA) [5–7], PCA [8,9], wavelet transform [10–12], and 1-D convolutional neural network (CNN) [13,14]. These feature extraction techniques are used together with learning algorithms for arrhythmia detection and classification. Each feature extraction technique focuses on different aspects of the signal, e.g., HRV features put emphasis on the features of heart beats, while ICA, PCA, and wavelet transform focus on the morphology of the signal. All of these methods have their unique strengths and drawbacks.

The machine learning algorithms used for arrhythmia classification include: support vector machine (SVM) [15–19], auto-regressive modeling [20], hidden Markov model (HMM) [21–23], a set of rules as determined by cardiologists [24,25], optimal path forest [26], and artificial neural networks [8,27–29]. In the past two years, many studies have adopted recurrent neural network (RNN) or long short-term memory (LSTM) as the learning algorithms of choice [30–34]. A number of studies have also applied deep learning methods for the classification of ECG signals [13,35,36]. In general these algorithms do not require specific signal pre-processing, QRS detection, or feature extraction steps. However, this advantage comes with the cost of requiring larger training datasets. This can be challenging as arrhythmia cases are relatively rare in comparison to healthy control cases.

Contrary to the above, to the best of our knowledge few studies have applied DPFA-based methods in the analysis of physiological signals, and especially not in arrhythmia detection and prediction using ECG signals. Deterministic probabilistic finite-state automata (DPFA) are finite-state automata in which every state is assigned a probability to each distinct letter in a fixed finite alphabet, which then determines a unique new state. More generally, in probabilistic finite-state automata (PFA), given a letter and current state, multiple transitions are allowed to other states with potentially different probabilities [37]. PFA have the same expressive power as HMM [37], despite the fact that only the latter has found wide application in ECG signal analysis.

Due to its deterministic assignment of transitions to the alphabet for a given state, DPFA have less expressive power than PFA. On the other hand, it is much easier to estimate the parameter values of DPFA than PFA [37]. As a result, DPFA will likely perform better when running the algorithm in real-time. Some well-developed special cases of DPFA include: n-grams with smoothing [38], Markov chains built by aggregating or mixing n-grams [39], and probabilistic suffix trees (PST) [40]. These algorithms have found applications in natural language processing [41,42] as well as in biomedical research such as protein structural analysis and sequence analysis [43,44].

2.1. Arrhythmia prediction

As detailed above, many algorithms and methods exist for the detection or classification of cardiac arrhythmias. A more challenging application of machine learning in this context is the prediction of cardiac events or arrhythmias before they occur. A reliable cardiac event prediction system could alert patients and clinicians alike of an impending event, thereby enabling timely intervention. However, prediction of arrhythmic events well before their onset is still an open research problem [45].

Therefore in this study we aim to construct a reliable prediction system for cardiac arrhythmias, specifically AHRE as a surrogate of AFib, and SVT, as they are the two most prevalent types of arrhythmia. AFib is the most common sustained cardiac arrhythmia. As of 2014, between 2.7 million and 6.1 million American adults were affected by AFib [46]. AFib is associated with a 5-fold increased risk of stroke [47], a 3-fold increased risk of heart failure, and a 2-fold increased risk of both dementia and mortality [48]. SVT is another common arrhythmia with mechanistic origin in the upper chambers of the heart, occurring with a prevalence of 2.25 cases per 1000 in the general population [49]. An algorithm focusing on AFib and SVT prediction, such as the proposed DPFA-based method, would greatly advance clinical decision making and provide better patient care.

3. Method

The proposed method not only classifies, but more importantly, allows us to predict heart rhythms by first generating two DPFA (arrhythmia versus non-arrhythmia) from the training datasets containing ECG signals, and then comparing the goodness-of-fit (as defined in Section 3.4.2) between each DPFA and the testing data.

3.1. Training method

A schematic diagram of the proposed method is illustrated in Fig. 1. The training dataset consists of annotated ECG signals. From these the algorithm extracts windows that are indicative of future arrhythmia events and others that are not. The positive ECG signals (i.e., pre-arrhythmia) are symbolized into probabilistic strings, which are then fed into the "DPFA Generation" module that constructs the positive DPFA M_+. The same is done for the negative ECG windows (i.e., regions that are neither arrhythmia nor pre-arrhythmia), producing the negative DPFA M_-. The "Symbolization" module is detailed in Section 3.2 and the "DPFA Generation" module is described in Section 3.3.

3.2. Symbolization

In this study, ECG signals are symbolized into strings of ternary words, but the method can be extended to larger alphabets. Let $g(t)$ be the input ECG signal, where t is time in seconds, $0 \leq t \leq L$, and L is the length of the ECG recording. $g(t)$ can be thought of as a continuous signal, but in implementation it is generally discrete with a high frequency (typically ≥ 200 Hz) depending on the ECG source. The symbolization process for a sample ECG signal is illustrated in Fig. 2.

A brief description of the main steps of the symbolization process is below, with pseudocode provided in Algorithm 1:

(a) The first step is to capture morphological features of width less than the duration of a typical QRS-complex: this is achieved by first subtracting the moving average $\frac{1}{2h_0}\int_{t-h_0}^{t+h_0} g(r)dr$ over intervals of width $2h_0$. The signal is then smoothed by convolving with the triangle function $\Lambda_{h_0}(t) = \max\{h_0 - |t|, 0\}$ and normalized by dividing by the local L^2 norm

$$\left|\int_{t-h_1}^{t+h_1} g^2 * \Lambda_{h_1}(r)dr\right|^{\frac{1}{2}}$$

over intervals of width $2h_1$.

The parameter h_0 controls the window size for baseline removal, while h_1 controls the length of the filter whose purpose is to capture the variability of magnitudes among adjacent heartbeats, while ensuring comparable average magnitudes over longer intervals. Therefore h_0 should be approximately the duration of a heart beat. Since a normal heart rate is around 60–100 beats per minute, so h_0 was chosen to be 2 s. h_1 needs to be longer than the duration of two or three heartbeats, but not too large as to reduce the local nature of normalization, so h_1 was chosen to be 5 s. Both parameters can also be tuned during the training steps for the purpose of facilitating the capture of local information

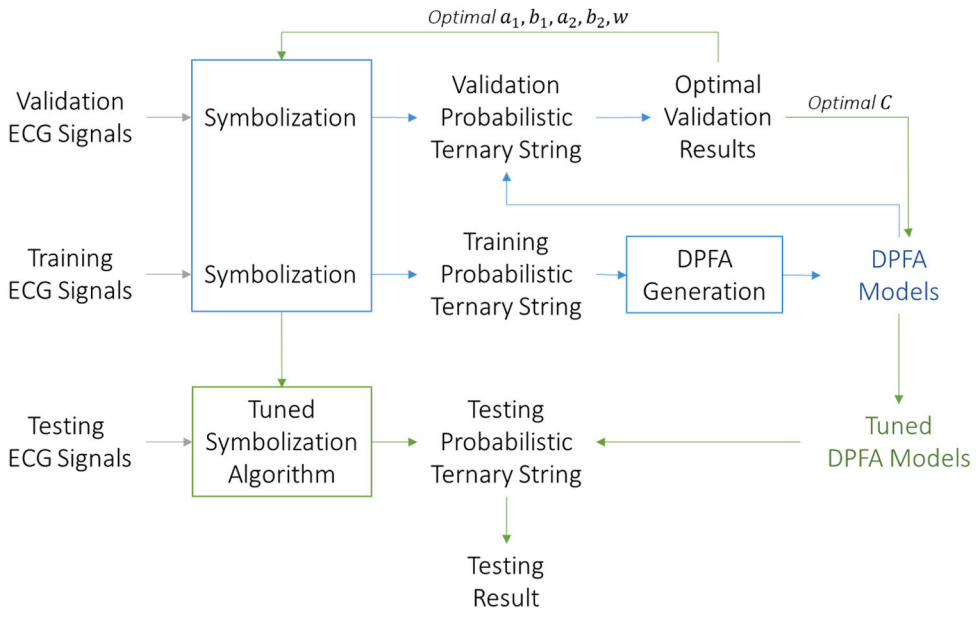

Fig. 1. A schematic diagram depicting the training and testing steps for the proposed method.

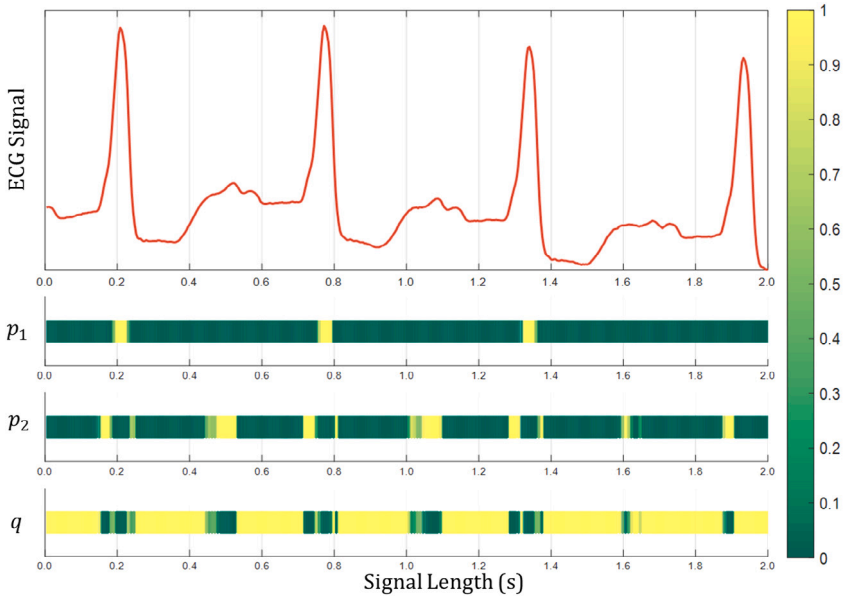

Fig. 2. An example of ECG signal symbolization into a ternary alphabet: $\{p_1, p_2, q\}$.

for use by the DPFA algorithm. For different applications these parameters may need to be modified.
Finally a piecewise linear filter is applied by forming the convolution

$$g * \phi(t) = g^{(-2)} * \phi^{(2)}(t),$$

where $g^{(-2)}$ denotes the second antiderivative of the signal $g(t)$ and $\phi^{(2)}(t)$ denotes the second derivative of the piecewise linear function $\phi(t)$. The expression on the right hand side has the advantage that $\phi^{(2)}$ is a sum of delta functions, which is better behaved when working with discrete signals.

(b) The second step is to apply downsampling to discretize the signal into $x_1 x_2 \dots x_n$ by

$$x_i = \max\left\{ g(t) \;\middle|\; (i-1)w \le t \le iw \right\},$$

where w is the downsampling window size tuned during the training step and $n = \lfloor L/w \rfloor$.

(c) The next step is to locally normalize the discrete signal by dividing x_i by the local maximum $\max\{x_{i-\lfloor h_2/w \rfloor}, \dots, x_{i+\lfloor h_2/w \rfloor}\}$ over intervals of width $2h_2$, where $h_2 = 1$ second. h_2 was chosen to be long enough to cover several QRS complexes.

(d) In the final step, the probabilities p_{i1}, p_{i2} and q_i are obtained from the normalized signal $\tilde{x}_1 \tilde{x}_2 \dots \tilde{x}_n$ via soft-thresholding

$$\begin{cases} p_{i1} &= \psi_1(\tilde{x}_i) \\ p_{i2} &= \big(1 - \psi_1(\tilde{x}_i)\big) \cdot \psi_2(\tilde{x}_i) \\ q_i &= 1 - p_{i1} - p_{i2}, \end{cases}$$

where the soft-thresholding functions are chosen to be piecewise linear functions $\psi_1, \psi_2 : [0,1] \to [0,1]$ of the form

$$\psi_j(x) = \begin{cases} 1 & \text{if } x > b_j \\ \frac{x - a_j}{b_j - a_j} & \text{if } a_j \le x \le b_j \\ 0 & \text{if } x < a_j \end{cases}$$

The parameters a_1, b_1, a_2, b_2 are all tuned in the training step. A grid search was performed for a_1 and b_1 with $a_1 < b_1$ from 0.1 through 0.9 with 0.1 increment. A grid search for a_2 and b_2 was performed in a similar manner. The parameter values used in this study were $a_1 = .6$, $b_1 = .8$, $a_2 = .025$ and $b_2 = .05$ for AHRE prediction and $a_1 = .4$, $b_1 = .8$, $a_2 = .04$ and $b_2 = .06$ for SVT prediction.

In summary, p_{i1}, p_{i2}, and q_i represent the probability of dominant patterns and local shapes depending on values of a_1, b_1, a_2, and b_2.

Algorithm 1 ECG Symbolization

Input: $g(t)$, $0 \le t \le L$ ▷ ECG
$h_0, h_1, h_2 > 0$ ▷ time parameters
$\phi(t)$, supported locally near $t = 0$ ▷ piecewise linear filter
$w > 0$ ▷ downsampling window size
$\psi_1, \psi_2 : [0,1] \to [0,1]$ ▷ soft-thresholding functions
Output: $\mathbf{P} = \mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_n = \begin{bmatrix} p_{11} \\ p_{12} \\ q_1 \end{bmatrix}\begin{bmatrix} p_{21} \\ p_{22} \\ q_2 \end{bmatrix} \cdots \begin{bmatrix} p_{n1} \\ p_{n2} \\ q_n \end{bmatrix}$ ▷ probabilistic string
for all $t \in [0, L]$ **do**
$g(t) \leftarrow g(t) - \frac{1}{2h_0}\int_{t-h_0}^{t+h_0} g(r)dr$ ▷ subtract moving average
$g(t) \leftarrow g * \Lambda_{h_0}(t)$ ▷ triangular smoothing
$g(t) \leftarrow |\int_{t-h_1}^{t+h_1} g^2 * \Lambda_{h_1}(r)dr|^{-\frac{1}{2}} g(t)$ ▷ normalization
$g(t) \leftarrow g^{(-2)} * \phi^{(2)}(t)$ ▷ piecewise linear filter
end for
$n \leftarrow \lfloor L/w \rfloor$ ▷ length of $\mathbf{P}$
for $i \leftarrow 1, n$ **do**
$x_i \leftarrow \max\{g(t) \mid (i-1)w \le t \le iw\}$ ▷ downsampling
end for
for $i \leftarrow 1, n$ **do**
$\tilde{x}_i \leftarrow x_i / \max\{x_j \mid i - \lfloor h_2/w \rfloor \le j \le i + \lfloor h_2/w \rfloor\} \in [0,1]$ ▷ local normalization
$p_{i1} \leftarrow \psi_1(\tilde{x}_i)$
$p_{i2} \leftarrow (1 - \psi_1(\tilde{x}_i))\psi_2(\tilde{x}_i)$
$q_i \leftarrow 1 - p_{i1} - p_{i2}$ ▷ soft-thresholding
end for

3.3. DPFA generation

The output of the symbolization module is a probabilistic string over the alphabet Σ. Such a probabilistic string defines a word distribution, for which the DPFA generation module produces a DPFA that best approximates this distribution. The DPFA is constructed in two steps: first by building a frequency prefix tree (FPT) and then performing state merging within the FPT.

3.3.1. The word distribution defined by a probabilistic string

Given the alphabet $\Sigma = \{a_1, \dots, a_d\}$ consisting of d symbols, let Σ^* denote the set of all words $a_{j_1}a_{j_2}\dots a_{j_m}$ of finite length, with ε representing the empty word. A word distribution is defined to be a function $f : \Sigma^* \to \mathbb{R}$ such that

(a) $f(w) \ge 0$, and
(b) $f(w) = \sum_{j=1}^{d} f(wa_j)$

for all $w \in \Sigma^*$.

Let $\mathbf{P} = \mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_n$ be a probabilistic string of length n consisting of d-dimensional probability vectors

$$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ \vdots \\ p_{id} \end{bmatrix}, \quad p_{ij} \ge 0 \text{ and } \sum_{j=1}^{d} p_{ij} = 1;$$

where p_{ij} denotes the probability that the ith letter of the probabilistic string is equal to $a_j \in \Sigma$. Then the probabilistic string defines a word distribution $f_{\mathbf{P}}$ by

$$f_{\mathbf{P}}(a_{j_1}a_{j_2}\dots a_{j_m}) = \sum_{i=1}^{n-m+1}\prod_{k=i}^{i+m-1} p_{ij_k} \tag{1}$$

for all non-empty $a_{j_1}a_{j_2}\dots a_{j_m} \in \Sigma^*$ and $f_{\mathbf{P}}(\varepsilon) = n$ (see Algorithm 2).

Algorithm 2 Word Distribution of a Probabilistic String

Input: $\mathbf{P} = \mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_n$ ▷ probabilistic string
Output: $f_{\mathbf{P}}(w) \in \mathbb{R}$ ▷ $w = a_{j_1}a_{j_2}\dots a_{j_m} \in \Sigma^*$
$w \leftarrow \varepsilon$
$p_{n+1}(w) \leftarrow 0$
for $i \leftarrow 1$ to n **do**
$p_i(w) \leftarrow 1$
end for
for $k \leftarrow 0$ to $m-1$ **do**
$w \leftarrow a_{j_{m-k}}w$
$p_i(w) \leftarrow p_{ij_{m-k}}p_{i+1}(w)$
end for ▷ $p_i(w) = \mathbb{P}(w$ occurs at the ith position of $\mathbf{P})$
$f_{\mathbf{P}}(w) \leftarrow \sum_{i=1}^{n} p_i(w)$

3.3.2. Constructing the FPT

A frequency prefix tree (FPT) is a tree-like automaton $T = \langle Q_0, \Sigma, \varepsilon, \text{Freq}\rangle$ with alphabet Σ, state space $Q_0 \subset \Sigma^*$, and initial state $\varepsilon \in Q_0$, which is also equipped with a frequency function $\text{Freq} : Q_0 \to \mathbb{R}$. Here the transition function is given by concatenation of words, which assigns $qa_j \in \Sigma^*$ to $(q, a_j) \in Q_0 \times \Sigma$ whenever $qa_j \in Q_0$.

In particular, a word distribution $f : \Sigma^* \to \mathbb{R}$ on the alphabet Σ defines an FPT with full state space $Q_0 = \Sigma^*$ and frequency function $\text{Freq} = f$. In general, such FPTs have infinite state spaces. However, in the special case when $\text{Freq}(w) = 0$ for all but finitely many $w \in \Sigma^*$, the FPT can be restricted to the finite state space

$$Q_0 = \{w \in \Sigma^* \mid \text{Freq}(w) \ne 0\}$$

without losing any information.

One such special case is when the word distribution f arises from a probabilistic string $\mathbf{P}$ over the alphabet Σ with finite length. This is because $f_{\mathbf{P}}(w) = 0$ for all words $w \in \Sigma^*$ whose length exceeds the length of $\mathbf{P}$.

3.3.3. Cutoff and state merging

A deterministic probabilistic finite-state automaton (DPFA) is an automaton $M = \langle Q, \Sigma, \varepsilon, P, T\rangle$ with state space Q, alphabet Σ, initial state $\varepsilon \in Q$ and transition function $T : Q \times \Sigma \to Q$, which is also equipped with a probability function $P : Q \times \Sigma \to \mathbb{R}$ such that

(a) $P(q, a_j) \ge 0$, and
(b) $\sum_{j=1}^{d} P(q, a_j) = 1$

for all $q \in Q$.

Given an FPT $T = \langle Q_0, \Sigma, \varepsilon, \text{Freq}\rangle$, one can construct a DPFA $M = \langle Q, \Sigma, \varepsilon, P, T\rangle$ by the largest suffix merging (LSM) algorithm (Algorithm 3); a brief description of which is provided below:

(a) the algorithm requires a cutoff parameter C such that $0 < C < \text{Freq}(\varepsilon)$;
(b) the state space Q of the DPFA consists of all $q \in Q_0$ which have frequency $\text{Freq}(q) > C$;
(c) the alphabet Σ and initial state $\varepsilon \in Q$ are the same as the FPT;
(d) the probability function is defined to be

$$P(q, a_j) = \frac{\text{Freq}(qa_j)}{\text{Freq}(q)}$$

for all $q \in Q$, which is well-defined since $\text{Freq}(q) > C > 0$;
(e) if $\text{Freq}(qa_j) > C$, then the transition function is defined to be $T(q, a_j) = qa_j \in Q$;

(f) otherwise $T(q, a_j) = s \in Q$ is defined to be the largest suffix s of qa_j with $\text{Freq}(s) > C$.

In summary, the LSM algorithm selects those states $q \in Q_0$ of T with sufficiently high frequency to be in the state space M, and then defines the transition state $T(q, a_j)$ to be the largest suffix of qa_j that is itself contained in the state space of M.

Fig. 3 provides an example of an FPT, while Fig. 4 depicts the DPFA constructed by the LSM algorithm. For example, in Fig. 4 the transition state $T(a, aba)$ is the largest suffix of $abaa$ contained in the state space. Since $abaa$ and aba are not in the state space, $T(a, aba) = aa$.

The mechanism for state merging in the LSM algorithm reflects the time-dependent nature of the underlying data (i.e., ECG signals). In particular, the transition state $T(q, a_j)$ is found by repeatedly deleting the left-most letter from the string qa_j, which represents the process of forgetting information from the most distant past, in order to update the system to the current state.

Algorithm 3 Largest Suffix Merging

Input: $T = \langle Q_0, \Sigma, \varepsilon, \text{Freq}\rangle$ ▷ FPT
$0 < C < \text{Freq}(\varepsilon)$ ▷ cutoff parameter
Output: $M = \langle Q, \Sigma, \varepsilon, P, T\rangle$ ▷ DPFA
$Q \leftarrow \{\varepsilon\}$ ▷ initial state
for all $q \in Q_0$ **do**
  while $\text{Freq}(q) > C$ **do**
    for $j \leftarrow 1, d$ **do**
      $P(q, a_j) \leftarrow \dfrac{\text{Freq}(qa_j)}{\text{Freq}(q)}$ ▷ probability function
      $T(q, a_j) \leftarrow qa_j$
      while $\text{Freq}\big(T(q, a_j)\big) \le C$ **do**
        $T(q, a_j) \leftarrow$ largest suffix of $T(q, a_j)$
      end while ▷ transition function
      $Q \leftarrow Q \cup \{T(q, a_j)\}$ ▷ state space
    end for
  end while
end for

3.3.4. Implementation

When training the DPFA $M_{\mathbf{P}}$ from the input probabilistic string $\mathbf{P}$, there is no need to construct the states of the FPT $T_{\mathbf{P}}$ with frequency less than C, and the LSM algorithm is executed to merge the states of $T_{\mathbf{P}}$ immediately after they are constructed.

The combined algorithm has computational complexity $O(dns)$ and needs total memory $O(nl)$, where d is the size of the alphabet Σ, n is the length of the probabilistic string $\mathbf{P}$, s is the size of the state space Q, and l is the length of the longest word in the state space Q. Both s and l depend on the cutoff parameter C.

3.4. Classification method

The classification scheme contains the training phase and the testing phase. By the end of the training phase, the algorithm has already learned the DPFA M_+ and M_- for the "positive ECG signals" and "negative ECG signals" classes respectively. Then to classify a given ECG signal $g(t)$ from the testing dataset, we first symbolize the signal into a probabilistic string $\mathbf{P}_g$, and then compare the goodness-of-fit between $\mathbf{P}_g$ and M_+ versus $\mathbf{P}_g$ and M_-. The goodness-of-fit is measured using the expected likelihood between word distributions.

3.4.1. The word distribution defined by a DPFA

Let $M = \langle Q, \Sigma, \varepsilon, P, T\rangle$ be a DPFA, then M defines a word distribution $f_M : \Sigma^* \to \mathbb{R}$ on the alphabet Σ by

$$f_M(a_{j_1} a_{j_2} \dots a_{j_m}) = \prod_{k=1}^{m} P(q_{k-1}, a_{j_k}) \tag{2}$$

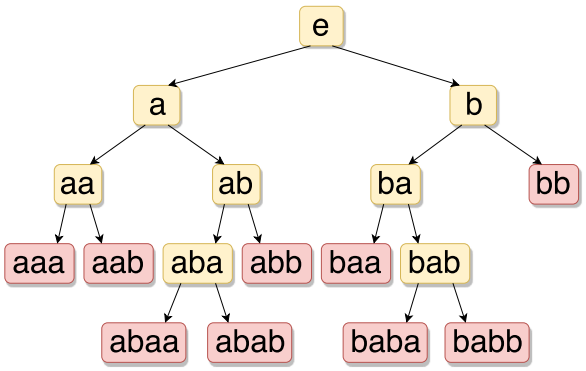

Fig. 3. FPT - The "DPFA generation" module searches through the tree. The yellow words have frequency at least C, while the red words have frequency less than C. This example will generate a DPFA with 8 states.

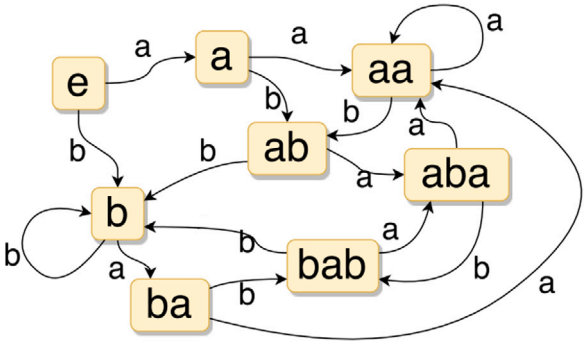

Fig. 4. The DPFA generated from the previous figure. The transition state $T(a, aba)$ is the largest suffix of $abaa$ contained in the state space. Since $abaa$ and aba are not in the state space, $T(a, aba) = aa$.

for all non-empty $a_{j_1} a_{j_2} \dots a_{j_m} \in \Sigma^*$ and $f_M(\varepsilon) = 1$, where the state $q_k \in Q$ is defined recursively by $q_0 = \varepsilon$ and $q_k = T(q_{k-1}, a_{j_k})$ (see Algorithm 4).

Algorithm 4 Word Distribution of a DPFA

Input: $M = \langle Q, \Sigma, \varepsilon, P, T\rangle$ ▷ DPFA
Output: $f_M(w) \in \mathbb{R}$ ▷ $w = a_{j_1} a_{j_2} \dots a_{j_m} \in \Sigma^*$
$w \leftarrow \varepsilon$
$q \leftarrow \varepsilon$
$f_M(w) \leftarrow 1$
for $k \leftarrow 1, m$ **do**
  $f_M(w) \leftarrow f_M(w) P(q, a_{j_k})$
  $w \leftarrow w a_{j_k}$
  $q \leftarrow T(q, a_{j_k})$
end for

3.4.2. Compute expected likelihood

The expected likelihood $\text{EL}(f_{\mathbf{P}}, f_M)$ is defined as the expected value $\mathbb{E}_{f_{\mathbf{P}}}[\mathcal{L}(f_M)]$ of the likelihood function $\mathcal{L}(f_M | w)$ over all words $w \in \Sigma^*$:

$$\text{EL}(f_{\mathbf{P}}, f_M) = \sum_{w \in \Sigma^*, f_{\mathbf{P}}(w) \neq 0} \mathcal{L}(f_M \mid w) f_{\mathbf{P}}(w) = \sum_{w \in \Sigma^*, f_{\mathbf{P}}(w) \neq 0} f_M(w) f_{\mathbf{P}}(w), \tag{3}$$

which is computed with respect to the word distribution $f_{\mathbf{P}}(w)$.

The summation in (3) is well-defined as there are only finitely many words $w \in \Sigma^*$ with $f_{\mathbf{P}}(w) \neq 0$ since $\mathbf{P}$ is a probabilistic string of finite length. As a measure of goodness-of-fit, a greater value of expected likelihood implies a better fit between the word distributions.

3.4.3. Classification

If $\mathrm{EL}(f_{\mathbf{P}_g}, f_{M_+}) > \mathrm{EL}(f_{\mathbf{P}_g}, f_{M_-})$, then the ECG signal $g(t)$ is classified as "positive ECG signal", otherwise it is classified as "negative ECG signal".

3.4.4. Implementation

The likelihood values $\mathcal{L}(f_{M_\pm}|w)$ can be calculated simultaneously with the word distribution $f_{\mathbf{P}_g}(w)$ as the algorithm searches through all the words $w \in \Sigma^*$ until $f_{\mathbf{P}_g}(w) = 0$. Once the values $\mathcal{L}(f_{M_\pm}|w)$ and $f_{\mathbf{P}_g}(w)$ are found for all relevant $w \in \Sigma^*$, one then applies formula (3) to compute the expected likelihoods $\mathrm{EL}(f_{\mathbf{P}_g}, f_{M_\pm})$.

The values $\mathrm{EL}(f_{\mathbf{P}_g}, f_{M_\pm})$ can be extremely small, which can lead to rounding errors. Thus, $\log \mathrm{EL}(f_{\mathbf{P}_g}, f_{M_\pm})$ is used instead of $\mathrm{EL}(f_{\mathbf{P}_g}, f_{M_\pm})$.

4. Preliminary analysis on Arrhythmia detection

To validate the performance of the algorithm, a preliminary analysis on arrhythmia detection was performed using publicly available benchmark databases. The databases include the MIT-BIH Arrhythmia Database (mitdb) [50] and the Creighton University Ventricular Tachycardia Database (cudb) [51]. For ventricular arrhythmia detection, the proposed algorithm achieved an AUC of 0.96 and F1-score of 0.91 for 5-second long signals and an AUC of 0.97 and F1-score of 0.93 for 2-second long signals [52]. However, the algorithm's performance on arrhythmia prediction was not examined, as most of the recordings in the publicly available databases were of insufficient duration.

5. Data sources

Experiments on two different types of arrhythmia prediction, AHRE and SVT, were performed in this study.

A retrospective database with ECG signals from Michigan Medicine cardiac patients with SVT or AFib was used in this study. After excluding recordings with pacemakers, implantable cardioverter defibrillators (ICD), or ventricular assist devices (VAD), there were 181 lead II ECG recordings for patients with SVT and 1210 lead II ECG recordings for patients with AFib. The recordings were sampled at 240 Hz.

6. Pre-processing and Arrhythmia event extraction

The aim of pre-processing is to remove noise from ECG signals. Methods focusing on heartbeat segmentation within the ECG signal tend to require pre-processing [53]. Automated annotation algorithms were used for both SVT and AFib to annotate the arrhythmia events. Pre-processing, noise removal, and arrhythmia event extraction were performed in the same fashion for both types of arrhythmia. The same pre-processing was applied to all methods in order to make the results commensurable. In order to demonstrate that the proposed algorithm does not rely on extensive pre-processing, a set of analysis using raw ECG data without any processing was also performed.

The following section describes the pre-processing and noise removal steps, followed by the arrhythmia annotations method used for each type of arrhythmia.

6.1. Pre-processing

During the signal pre-processing step, a fourth-order Butterworth band-pass filter with cutoff frequencies of 0.5 and 40 Hz is first applied to the raw ECG signal to remove noise, after which a double median filter with orders equal to 0.2 and 0.6 times the sampling frequency is applied to remove baseline wandering. To calculate heart rate and beat duration, the Pan–Tompkins algorithm for QRS detection is used [54, 55].

6.2. Noise removal

After R peak detection, a stepwise noise detection method is performed. There are 3 criteria for noise detection. First, the method calculates the percentage of missing signal in a defined time window (300 s) and checks if the missing signal percentage is above a certain threshold (15%). If the percentage of the missing signal is above the threshold, the signal is classified as being noisy by the first criteria. Secondly, if the signal passes the first criterion then non-linear filter analysis is used to determine missing peaks. If the signal passes both the first and the second criteria, the third step determines if the percentage of missing R peaks is above a certain threshold (15%) in the current window. The window of signal will be classified as noisy if it fails any of these three criteria. Noisy signals will not be used for subsequent annotation. Signals that occur too close to the noisy signal will also be excluded from the analysis.

6.3. Pre-event signal extraction

The aim of this study is to predict the onset of an arrhythmia event using pre-event signals. In order to assess the predictive power of the method, events that occurred too close to previous events were excluded, as the prediction interval should not overlap with arrhythmia events. Events that occurred within 8 min of a noisy signal were also excluded, with the aim to ensure that the prediction interval is out of the noisy signal range. Fig. 5 shows an example of an annotated ECG signal. After annotation, there is an annotated event A and attendant prediction interval B, which is located t_{gap} minutes before the annotated event A with a length of t_{signal} minutes. The interval C classified as noise will be excluded from the analysis, along with the neighboring intervals of 8 min.

In the prediction experiments, different combinations of t_{gap} and t_{signal} for both SVT and AFib were tested.

7. Comparison with other methods

Automatic detection of types of arrhythmia or cardiac conditions encompasses several basic steps, including pre-processing/segmentation, feature extraction, followed by a classifier [56]. Machine learning (ML) approaches for arrhythmia detection can be grouped into two main categories based on feature extraction strategies [45]. The first group uses features extracted from the raw ECG as input followed by classical ML algorithms such as SVM, KNN, and decision trees as classifiers. Such ML algorithms require dimensionality reduction after feature extraction from the ECG signals. The second group uses raw ECG data as input and does not require feature extraction. Information from the original physiological signals can be directly processed by algorithms in this group [57]. ML algorithms like neural networks, including their basic and advanced versions, use the raw data for model training and detection of arrhythmia types. Compared to the first group, algorithms in this group generally have higher computational cost and require larger datasets to achieve equal or greater performance. Our method belongs to the second group, even though it is not based on neural networks.

In medical applications, datasets that are both sufficiently large and well-annotated are not always available due to nature of the various diseases. Thus both groups have their advantages and drawbacks. In this study we selected five methods from both groups for comparison.

The first method is an SVM based arrhythmia classification algorithm with features extracted from heart rate variability (HRV with SVM) by Asl et al. [3]. Following [3], time and frequency domain features were extracted from the signals. However, generalized discriminant analysis (GDA) for feature reduction was not utilized in this study since GDA would reduce the number of features, and there are only two classes to begin with. The SVM model with radial basis function (RBF) kernel was used for classification. The cost was set to 1:5 for the SVT

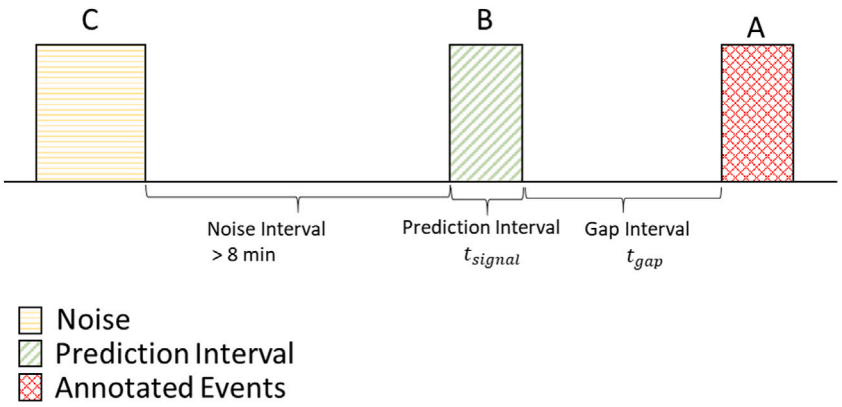

Fig. 5. Intervals of signal are extracted for use in prediction that are t_{gap} minutes prior from the arrhythmia event and 8 or more minutes away from detected noise.

prediction experiment and 1:11 for the AFib prediction experiment. Parameters for SVM were tuned using the training datasets.

The second method for comparison uses the principal components of the Discrete Wavelet Transform (DWT) coefficients as features and SVM as the classifier [9].

The third comparison method is a deep neural network (DNN) created for arrhythmia prediction by Hannun et al. [58]. The authors used a convolutional neural network to detect types of arrhythmia. The network architecture has 34 layers, takes ECG signals as input and outputs arrhythmia classes. The same experimental setup was used as in [58]. The network was trained de novo with random initialization and the Adam optimizer with default parameters.

The fourth and the fifth comparison methods are both based on CNN techniques. The fourth method, due to Acharya et al. [13], utilized a CNN algorithm to automatically detect different ECG segments. The algorithm consists of an eleven-layer deep CNN. The fifth method, due to Tan et al. [59], employed a long short-term memory (LSTM) network (CNN with LSTM) to classify ECG signals.

To compare the performance of the five alternative algorithms, the same training, cross validation, and testing data sets were used.

8. Case study I: SVT prediction

8.1. SVT event annotation

After pre-processing and noise removal as described in Section 6, an automated annotation algorithm was applied to determine SVT episodes with the ECG signal. Time windows consisting of three R-R intervals were annotated as SVT if they satisfied all of the following four SVT annotation criteria. First, an automated annotation algorithm based on the heart rate-duration criteria line was used to annotate high heart rate events. Portions of signals with extremely high heart rate y_2 that last for a short time period x_1, and a relatively lower heart rate y_1 that last for a longer time period x_2, are both considered to be indicative of high heart rate events. Using a straight line passing through the points (x_1, y_2) and (x_2, y_1), the lower bound for heart rates in the criteria region between duration x_1 and x_2 are determined. If any of the heart rates on the duration grid are above the heart rate-duration criteria line, then that portion of the signal lies in the criteria region and will be classified as high heart rate. In the case of SVT, the episodes lie above the heart rate-duration criterion line passing through the points $(30, 150)$ and $(180, 100)$. Secondly, the difference between three consecutive R-R intervals should be less than 50 ms. Thirdly, there should be fewer than 4 P/T like waves within the window. Finally, the cross-correlations with the 500 P-wave templates extracted from the MIT-BIH P-wave database should be less than 0.85. A portion of the annotated SVT episodes were randomly selected and reviewed by a cardiologist to confirm the sensitivity of the annotation algorithm.

8.2. Data partition

A total of 149 SVT events were annotated by the algorithm and a total of 755 non-SVT intervals were randomly selected from the non-SVT regions of the signals. A total of 119 (80%) of the SVT intervals were used in training with the remaining 30 intervals being held out for testing. Within the training data set, 5-fold cross validation was performed for parameter tuning (Fig. 6). Training, cross validation, and testing sets/folds were partitioned on a participant level, meaning that signals from the same participant were only included in one set/fold so as to prevent overfitting.

The five models (each with four folds of training data) with the best parameters tuned during cross validation were then applied to the testing set.

8.3. SVT prediction results

Different combinations of signal intervals (i.e., $t_{signal} = 0.5, 1.0, 2.0$ minutes) and gap intervals (i.e., $t_{gap} = 0.5, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5$ minutes) up to 5 min before the event were used for prediction. These prediction intervals were tested using the DPFA algorithm. Fig. 7 shows a summary of the performance. The AUC is above 0.75 for all prediction intervals. As the time to the SVT event increases, the mean prediction performance gradually declines while variance increases. For the SVT DPFA, an average of 114×3 transition states were generated for the models using 0.5-min long signals, 97×3 transitions states for the 1.0-min long signal models, and 60×3 transition states for the 2.0-min long signal models. For the non-SVT DPFA, an average of 114×3 transition states were generated for the models using 0.5-min long signals, 95×3 transitions states for the 1.0-min long signal models, and 56×3 transition states for the 2.0-min long signal models.

The performance of the DPFA on raw ECG data was compared with its performance on pre-processed data (Fig. 7). The AUC for all prediction intervals are almost the same with or without pre-processing and there is no statistically significant difference between these results. These results indicate that pre-processing does not impact the performance of the DPFA algorithm.

The proposed DPFA algorithm has on average a 10.2% higher AUC than the deep learning method, 32.9% higher than DWT with PCA, 33.3% higher than CNN and 26.4% higher than CNN with LSTM for various prediction intervals. The HRV with SVM algorithm has comparable performance on SVT prediction, with a slightly lower average AUC compared to DPFA. Fig. 8 summarizes the performance of DPFA together with the 5 alternative methods. The McNemar test was also performed to evaluate the performance of the alternative methods against the proposed DPFA algorithm. The McNemar test used the chi-squared test and a 1:5 ratio for the cost for the imbalance of data. The results are shown in Tables 1, 2, and 3. p-values representing the

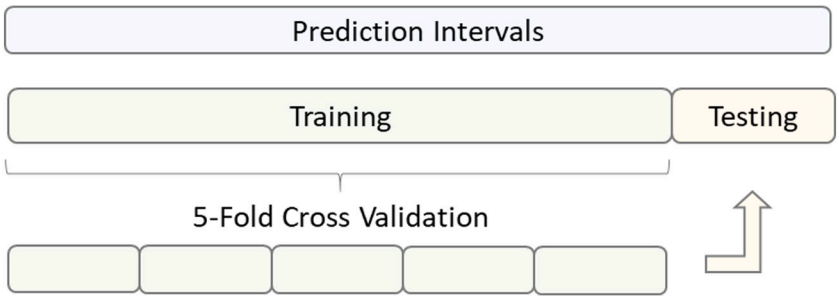

Fig. 6. Data partitioning scheme used for arrhythmia prediction.

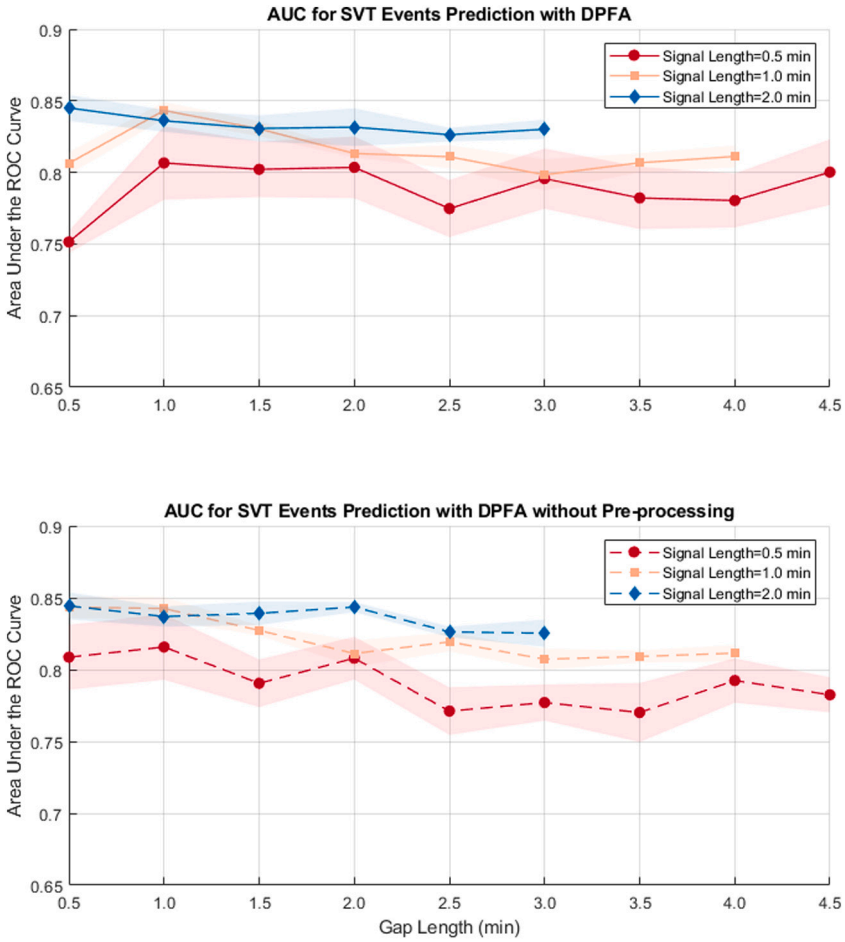

Fig. 7. A comparison of AUC for SVT prediction using the DPFA method with and without pre-processing for various signal lengths and gap intervals.

statistical significance of the different performances in terms of AUC (< .05) are shown in bold font in the referenced tables. There is no statistically significant difference between DPFA and the HRV method. DPFA is significantly better than the deep learning algorithm for 1-min long prediction signals. DPFA is significantly better than the DWT, CNN and CNN with LSTM method for most of the prediction intervals. It is also worth noting that the performance of the DPFA algorithm with raw ECG as input is also higher than the deep learning, DWT, CNN, and CNN with LSTM methods.

9. Case study II: AFib/AHRE prediction

AFib is defined to be a cardiac arrhythmia with three characteristics: "absolutely" irregular RR intervals, no distinct P waves, and variable atrial cycles of length <200 ms (> 300 bpm). However, the qualitative nature of these characteristics implies that AFib detection and diagnosis relies heavily on human input, which impedes the development of automated decision support systems that utilize ECG data. To mitigate this, a heart rate-duration criteria line was used to annotate AFib events with relatively high heart rate (AHRE).

9.1. AFib/AHRE event annotation

Heart rate is calculated based on the R peaks within a time interval, which were detected previously during test interval extraction. Duration is calculated by counting the number of consecutive intervals that a particular heart rate spans. For this study, the time interval was set to 30 s and counted up to 6 intervals, which allows for the duration to range from 30 to 180 s.

The automated annotation algorithm based on the heart rate-duration criteria region as described in Section 8.1 in SVT annotation

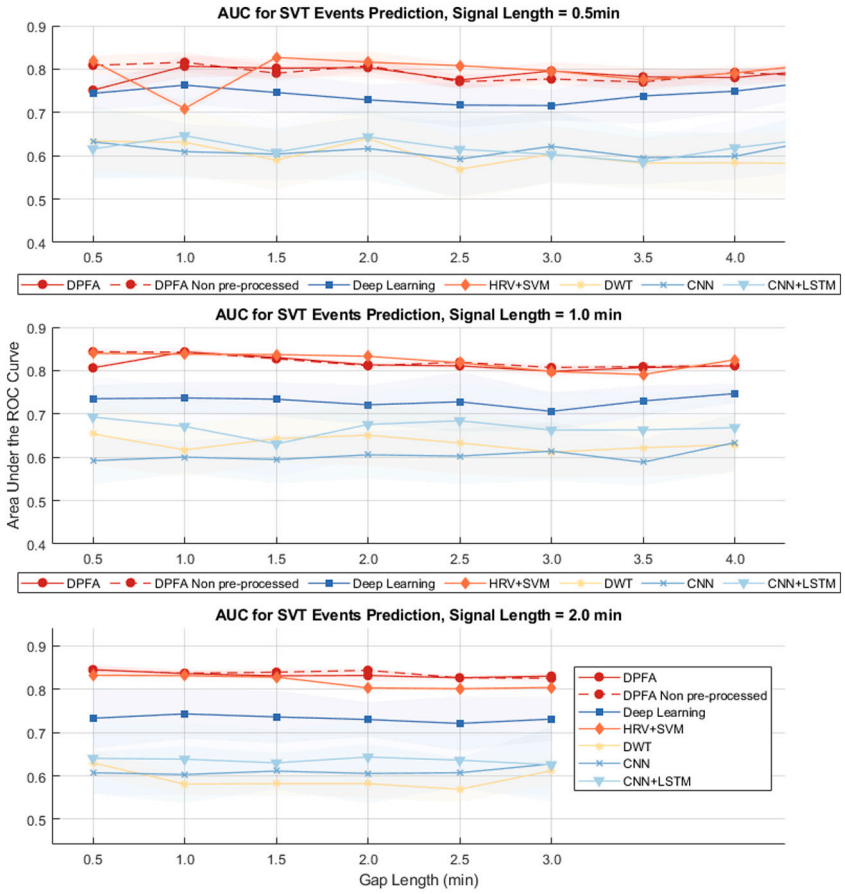

Fig. 8. A comparison of AUC for SVT prediction for all models using various signal lengths and gap intervals.

Table 1
A comparison of AUC for SVT prediction with different gap intervals for half-minute long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.751 (0.008)	0.744 (0.040)	0.090	0.818 (0.004)	0.101	0.634 (0.068)	0.812	0.634 (0.059)	0.897	0.616 (0.069)	0.084
1.0	0.807 (0.026)	0.763 (0.041)	0.542	0.709 (0.012)	0.126	0.631 (0.075)	0.122	0.664 (0.069)	**0.019**	0.646 (0.093)	**0.024**
1.5	0.802 (0.019)	0.746 (0.045)	0.679	0.827 (0.005)	0.869	0.590 (0.068)	0.117	0.664 (0.069)	**0.035**	0.608 (0.070)	**0.027**
2.0	0.804 (0.022)	0.729 (0.036)	0.404	0.816 (0.025)	0.858	0.640 (0.070)	0.407	0.653 (0.050)	0.172	0.644 (0.076)	**0.013**
2.5	0.775 (0.020)	0.717 (0.052)	0.130	0.808 (0.006)	0.805	0.569 (0.075)	0.119	0.630 (0.079)	**0.022**	0.615 (0.106)	**0.022**
3.0	0.796 (0.021)	0.716 (0.035)	0.227	0.796 (0.004)	0.686	0.604 (0.064)	**0.038**	0.646 (0.087)	0.082	0.603 (0.067)	**0.017**
3.5	0.782 (0.022)	0.738 (0.063)	0.364	0.775 (0.009)	0.789	0.583 (0.058)	**0.041**	0.650 (0.066)	0.113	0.585 (0.055)	**0.014**
4.0	0.780 (0.019)	0.749 (0.048)	0.718	0.791 (0.006)	0.591	0.584 (0.068)	**0.005**	0.653 (0.072)	0.126	0.618 (0.064)	**0.014**
4.5	0.800 (0.023)	0.774 (0.029)	0.716	0.814 (0.005)	0.353	0.581 (0.076)	**0.039**	0.684 (0.053)	0.124	0.643 (0.094)	**0.026**

was also used to annotate AHRE events with different heart rate and duration limits. The lower duration limit of 30 s is chosen in accordance with the definition of an AFib event provided by the 2014 AHA guideline [46]. The lower heart rate limit is set to 110 bpm, as this rate has proven to be an effective treatment target for AFib [60]. The higher heart rate limit is set at 160 bpm since a rapid heart rate is more likely to cause symptoms.

Rapid heart rate in AFib may also have an untoward effect on cardiac function, resulting in tachycardia-induced cardiomyopathy [61]. Using this annotation method, it becomes possible to capture the portions of the signal that correspond to AHRE with sufficiently high

Table 2
A comparison of AUC for SVT prediction with different gap intervals for 1 min long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.806 (0.008)	0.735 (0.031)	0.066	0.840 (0.004)	0.342	0.654 (0.072)	1.000	0.680 (0.068)	0.098	0.693 (0.094)	0.566
1.0	0.843 (0.006)	0.737 (0.037)	0.232	0.838 (0.004)	0.519	0.617 (0.061)	**0.010**	0.680 (0.069)	**0.005**	0.671 (0.090)	**0.004**
1.5	0.830 (0.005)	0.734 (0.039)	**0.030**	0.837 (0.002)	0.629	0.643 (0.054)	**0.020**	0.644 (0.076)	**0.007**	0.631 (0.077)	**0.002**
2.0	0.813 (0.003)	0.721 (0.044)	**0.038**	0.833 (0.003)	1.000	0.651 (0.070)	0.079	0.639 (0.075)	**0.005**	0.676 (0.074)	0.074
2.5	0.811 (0.008)	0.728 (0.069)	**0.015**	0.818 (0.002)	0.925	0.633 (0.072)	0.065	0.677 (0.098)	**0.002**	0.684 (0.084)	**0.053**
3.0	0.798 (0.011)	0.706 (0.043)	**0.041**	0.798 (0.003)	0.858	0.612 (0.060)	0.060	0.663 (0.069)	**0.024**	0.663 (0.088)	**0.017**
3.5	0.807 (0.007)	0.730 (0.035)	0.286	0.791 (0.007)	0.831	0.622 (0.066)	0.093	0.644 (0.064)	**0.009**	0.663 (0.084)	**0.022**
4.0	0.811 (0.008)	0.747 (0.022)	0.334	0.825 (0.002)	0.276	0.629 (0.065)	0.334	0.666 (0.086)	**0.049**	0.668 (0.090)	**0.016**

Table 3
A comparison of AUC for SVT prediction with different gap intervals for 2 min long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.845 (0.009)	0.733 (0.069)	0.924	0.832 (0.004)	0.917	0.630 (0.029)	**0.012**	0.663 (0.076)	**0.003**	0.640 (0.083)	**0.002**
1.0	0.836 (0.008)	0.743 (0.057)	0.916	0.831 (0.004)	0.575	0.581 (0.024)	**0.007**	0.664 (0.073)	**0.019**	0.639 (0.083)	**0.007**
1.5	0.831 (0.009)	0.736 (0.060)	0.925	0.828 (0.003)	0.812	0.582 (0.024)	**0.004**	0.659 (0.070)	**0.004**	0.630 (0.076)	**0.001**
2.0	0.832 (0.013)	0.730 (0.040)	0.276	0.803 (0.004)	0.644	0.582 (0.033)	**0.029**	0.673 (0.086)	**0.016**	0.643 (0.083)	**0.006**
2.5	0.826 (0.005)	0.721 (0.062)	0.753	0.801 (0.005)	0.680	0.569 (0.029)	**0.001**	0.654 (0.102)	**0.004**	0.636 (0.077)	**0.004**
3.0	0.830 (0.007)	0.731 (0.050)	0.488	0.804 (0.003)	0.741	0.612 (0.031)	**0.024**	0.668 (0.092)	**0.034**	0.626 (0.069)	**0.013**

severity, either in the form of extremely high heart rate over a span of 30 s, or moderately high heart rate stretching over 180 s. A heart rate-duration criteria region that lies above the line passing through the points $(30, 160)$ and $(180, 110)$ is used. These AHRE events were annotated as surrogate for AFib.

9.2. Data partition

A total of 417 AFib events were labeled by these criteria and a total of 7319 non-AFib intervals were randomly selected from non-AFib regions of the signals. A total of 353 of the AFib intervals were used in training with the remaining 64 intervals being held out for testing. Within the training data set, 5-fold cross validation was performed for parameter tuning (Fig. 6). Similar to the SVT experiment, training, cross validation, and testing sets/folds were partitioned at the participant level.

9.3. AHRE prediction results

Different combinations of signal intervals (i.e., $t_{signal} = 0.5, 1.0, 2.0$ minutes) and gap intervals (i.e., $t_{gap} = 0.5, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5$ minutes) up to 5 min prior to the event were used for prediction. These prediction intervals were tested using the DPFA algorithm. Fig. 9 depicts a summary of the performance. For the AHRE DPFA, around 758×3, 683×3, and 582×3 transition states were generated for 0.5, 1.0 and 2.0-min long AHRE prediction intervals based on the training data set. For the non-AHRE, an average of 626×3 transition states were generated for the models using 0.5-min long signals, 642×3 transitions states for the 1.0-min long signal models, and 459×3 transition states for the 2.0-min long signal models.

The AUC is above 0.80 for all prediction intervals. Performance is nearly steady when gap length increases and the prediction interval moves further away from the AHRE events. Prediction performances with 2-min long signals are slightly better than 1-min long and 0.5-min long signals (Fig. 9). Longer signals contain more local and global patterns and these information may have contributed to the better performances.

The performance of DPFA on raw ECG data was compared with its performance on data with pre-processing (Fig. 9). Similar to the results for SVT prediction, the AUCs for prediction intervals are almost the same with or without pre-processing. Nor is there any statistically significant difference between these results.

For AHRE prediction, the DPFA algorithm has a comparable performance to deep learning with on average a 2% higher AUC. For the other comparison algorithms, DPFA has a 17.9% higher AUC than HRV, 26.1% higher AUC than DWT, 15.4% higher AUC than CNN and 10.3% higher AUC than CNN with LSTM. The results for these experiments are shown in Fig. 10.

The McNemar test was also performed to evaluate the performance of the alternative methods against the proposed DPFA algorithm. The

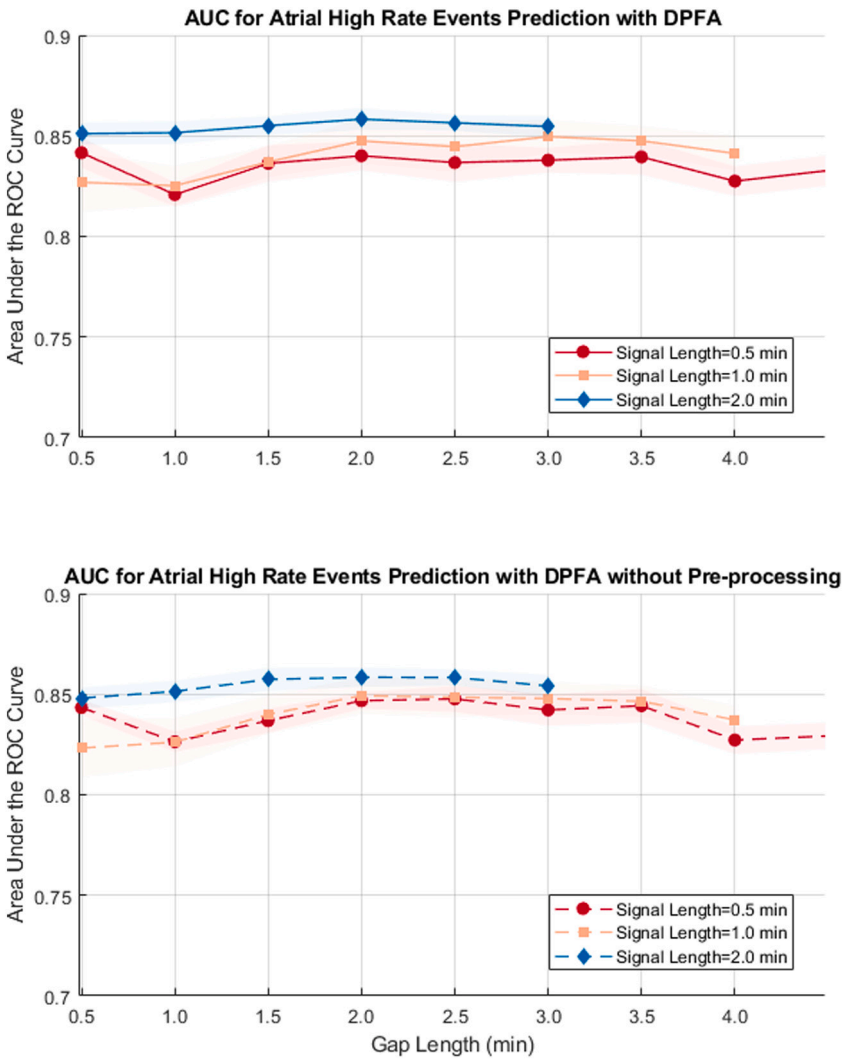

Fig. 9. A comparison of AUC for AHRE prediction using the DPFA with and without pre-processing method for various signal lengths and gap intervals.

McNemar test used the chi-squared test and a 1:11 ratio for the cost for the imbalance of data. Tables 4, 5, and 6 depict the results for half-minute, 1-min, and 2-min long signals with various gap sizes, respectively. *p*-values representing the statistical significance of the different performance in terms of AUC ($< .05$) are shown in bold font in the referenced tables. The deep learning algorithm has no statistically significant difference in performance with respect to the McNemar test. However, the DPFA algorithm has 40.1% lower variance than deep learning. On the other hand, HRV, CNN and DWT have much lower AUC. The results for HRV with SVM were obtained after further excluding certain cases. Two AHRE cases and 20 of control cases were excluded from the test dataset because peak annotation could not be correctly applied to the signals. Although CNN with LSTM has lower AUC than the DPFA model, there is no significant difference in the McNemar test results for 1-min and 2-min long signals. CNN with LSTM performs better than the CNN model.

10. Discussion

Many studies have focused on arrhythmia classification, but very few studies have attempted to predict the onset of an arrhythmia event. This paper proposes an algorithm that is not only useful for classification but also for prediction. As compared to other methods such as HRV, DWT, and deep learning, the method based on the DPFA algorithm has achieved around 0.8 AUC for all prediction intervals for both SVT, which is a small dataset, and AHRE, which is a relatively large dataset.

Remarkably, the proposed novel DPFA algorithm utilizes minimal pre-processing and does not require specific peak identification. The prediction results with raw ECG signals are almost the same as the results with pre-processing. This method represents a valuable alternative to traditional methods that require heavy pre-processing and feature extraction.

We have compared the DPFA algorithms with five other algorithms. The HRV with SVM method has good performance on SVT prediction when the dataset is relatively small. The performance drops when applied to AHRE prediction. Moreover, the automated annotation algorithm for both methods are directly related to the HRV features, which might have unknowingly affected the results. Several cases in AHRE prediction were excluded as a result of difficulties in peak annotation for the HRV with SVM method. The proposed DPFA method, on the other hand, does not require a peak annotation algorithm. As a consequence, the DPFA algorithm is more likely to perform better on other classification problems, where peak locations are not among the primary features.

Neural network based algorithms such as deep learning, CNN, and CNN with LSTM are similar to the DPFA algorithm in that no specific pre-processing is required. However, as shown in the experiments, they require a larger dataset for better performance. The performances of these three algorithms are worse than DPFA and HRV in SVT prediction

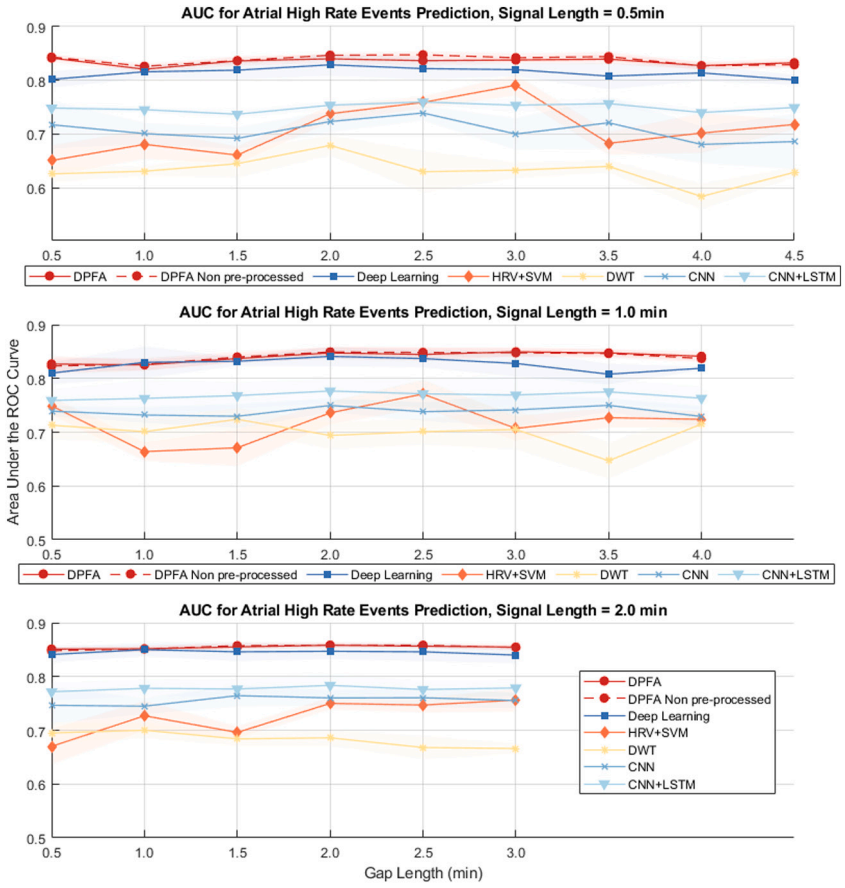

Fig. 10. A comparison of AUC for AHRE prediction for all models using various signal lengths and gap intervals.

Table 4
A comparison of AUC for AHRE Prediction with different gap lengths for half-minute long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.842 (0.007)	0.802 (0.015)	0.852	0.651 (0.028)	**0.004**	0.626 (0.015)	**0.001**	0.718 (0.044)	0.335	0.749 (0.004)	**0.000**
1.0	0.821 (0.006)	0.816 (0.012)	0.202	0.681 (0.027)	0.240	0.631 (0.008)	0.236	0.701 (0.020)	**0.001**	0.746 (0.005)	**0.000**
1.5	0.836 (0.009)	0.819 (0.017)	0.419	0.661 (0.016)	**0.002**	0.645 (0.025)	**0.023**	0.692 (0.020)	**0.000**	0.737 (0.006)	**0.000**
2.0	0.840 (0.007)	0.829 (0.020)	0.701	0.738 (0.015)	**0.007**	0.679 (0.018)	**0.000**	0.723 (0.019)	**0.015**	0.754 (0.007)	**0.000**
2.5	0.837 (0.010)	0.822 (0.020)	0.469	0.759 (0.011)	**0.001**	0.630 (0.038)	0.156	0.739 (0.013)	**0.027**	0.760 (0.004)	**0.000**
3.0	0.838 (0.006)	0.820 (0.012)	0.719	0.791 (0.019)	**0.003**	0.633 (0.014)	0.353	0.700 (0.029)	**0.020**	0.754 (0.011)	**0.000**
3.5	0.840 (0.009)	0.808 (0.024)	0.151	0.683 (0.021)	**0.000**	0.640 (0.011)	**0.000**	0.721 (0.048)	**0.028**	0.757 (0.008)	**0.000**
4.0	0.828 (0.007)	0.814 (0.015)	0.855	0.702 (0.037)	**0.010**	0.584 (0.023)	0.757	0.681 (0.034)	0.133	0.740 (0.007)	**0.003**
4.5	0.833 (0.008)	0.801 (0.009)	0.461	0.718 (0.009)	**0.006**	0.629 (0.009)	**0.061**	0.686 (0.051)	**0.005**	0.750 (0.007)	**0.000**

due to the limited sample size. The prediction performances are better for AHRE when the size of datasets are larger, especially with longer signals. However, the variances of the deep learning and CNN with LSTM methods are higher than the DPFA model. CNN with LSTM has better performance overall compared to CNN. Although the required training time is longer than CNN, LSTM improves the performance of the CNN model. Overall, the proposed DPFA algorithm has consistently good performance on both SVT predictions and AHRE prediction, regardless of the size of the datasets.

Table 5
A comparison of AUC for AHRE Prediction with different gap intervals for 1 min long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.827 (0.014)	0.810 (0.020)	0.751	0.749 (0.017)	**0.000**	0.713 (0.025)	**0.005**	0.739 (0.009)	0.056	0.759 (0.019)	0.780
1.0	0.825 (0.010)	0.830 (0.029)	0.104	0.664 (0.017)	**0.004**	0.701 (0.024)	0.232	0.732 (0.015)	0.345	0.763 (0.034)	0.803
1.5	0.837 (0.008)	0.832 (0.008)	0.105	0.671 (0.034)	0.083	0.724 (0.021)	0.080	0.730 (0.023)	0.081	0.768 (0.024)	0.830
2.0	0.847 (0.011)	0.841 (0.018)	0.265	0.736 (0.021)	**0.000**	0.694 (0.025)	**0.000**	0.750 (0.013)	0.101	0.777 (0.022)	0.822
2.5	0.845 (0.013)	0.837 (0.016)	0.297	0.771 (0.026)	0.190	0.701 (0.024)	0.080	0.738 (0.018)	0.562	0.771 (0.026)	0.902
3.0	0.850 (0.008)	0.828 (0.024)	0.385	0.707 (0.023)	**0.062**	0.705 (0.036)	0.176	0.742 (0.031)	**0.023**	0.769 (0.035)	0.377
3.5	0.848 (0.007)	0.808 (0.019)	0.720	0.727 (0.018)	**0.007**	0.647 (0.033)	**0.000**	0.750 (0.013)	0.217	0.775 (0.022)	0.620
4.0	0.841 (0.008)	0.819 (0.003)	0.294	0.724 (0.015)	**0.000**	0.715 (0.025)	**0.005**	0.729 (0.019)	**0.007**	0.763 (0.021)	0.366

Table 6
A comparison of AUC for AHRE Prediction with different gap intervals for 2 min long signals.

Gap length (min)	DPFA	Deep learning		HRV + SVM		DWT + PCA		CNN		CNN + LSTM	
	Mean (STD)	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value	Mean (STD)	p value
0.5	0.851 (0.005)	0.841 (0.014)	0.219	0.670 (0.033)	0.174	0.695 (0.020)	**0.001**	0.746 (0.034)	0.071	0.772 (0.017)	0.658
1.0	0.852 (0.006)	0.850 (0.012)	0.102	0.727 (0.028)	**0.000**	0.700 (0.014)	**0.000**	0.745 (0.015)	0.096	0.778 (0.015)	0.812
1.5	0.855 (0.005)	0.846 (0.017)	0.961	0.696 (0.011)	**0.001**	0.684 (0.012)	**0.000**	0.764 (0.020)	0.602	0.777 (0.017)	0.949
2.0	0.858 (0.005)	0.847 (0.014)	0.208	0.750 (0.011)	**0.006**	0.686 (0.014)	**0.001**	0.760 (0.014)	0.175	0.784 (0.019)	0.828
2.5	0.856 (0.004)	0.846 (0.017)	0.449	0.747 (0.014)	**0.005**	0.668 (0.021)	**0.005**	0.761 (0.018)	0.211	0.776 (0.015)	0.715
3.0	0.855 (0.005)	0.840 (0.015)	0.434	0.756 (0.021)	**0.006**	0.666 (0.009)	**0.000**	0.755 (0.014)	0.190	0.779 (0.014)	0.957

There are several limitations of the study. The first limitation is the use of the automated annotation algorithm. To validate the performance of the algorithm, the positive cases were verified by a cardiologist, but not the negative cases. In a future study, we aim to have the dataset annotated by clinicians, which is the gold standard for annotations. The proposed DPFA model can be easily trained on other labeled datasets. The performance of the DPFA algorithm has been tested on the MIT-BIH database using clinician-labeled annotations for SVT and AFib vs. all other rhythms. For both experiments, the AUC is above 0.95. Secondly, extremely noisy sections of the ECG signals have been removed from the beginning. These sections have been randomly checked by the clinicians and most of them are not suitable for classifications. For the purpose of annotations that rely on peak detection the signals were pre-processed and noisy sections were excluded. In the future, using events manually annotated by clinicians, these noisy sections can be included as an additional class for prediction instead of removing them. Thirdly, the database used in this study is imbalanced, with many more negative cases than arrhythmia cases. As a result, the costs and weights in the deep learning and SVM models were adjusted accordingly to make the prediction more balanced. In the future it is possible to employ additional data augmentation algorithms to increase the positive cases, which may benefit both the deep learning and DPFA methods. Lastly, the performance of the DPFA algorithm is still limited by the number of cases. Some participants had substantially more cases than others, thus inter-patient ECG signal variability may limit the performance of our algorithm as well as the other alternative algorithms.

11. Conclusion

In this paper a novel DPFA based method is presented to classify and predict arrhythmia events. The proposed method takes a probabilistic string extracted from ECG signals in the training set as input, and via frequency analysis, constructs the underlying state space and transition probabilities of the DPFA model, directly from the input data. When applying the DPFA algorithm to classification and prediction problems, the decisions are based on comparing goodness-of-fit between the test signal and various DPFA. The proposed method achieved over 0.8 AUC for both SVT and AHRE prediction experiments. Compared with other 5 well-established methods, the proposed DPFA algorithm has achieved better classification results, even though in some cases the advantage might not be statistically significant. In addition, the performance of the proposed DPFA algorithm is almost identical with or without any pre-processing on the data. With additional validation, the proposed method could be deployed as a cardiac event prediction system, alerting patients and clinicians alike of an impending event and thereby enabling timely intervention.

CRediT authorship contribution statement

Zhi Li: Conceptualization, Data curation, Formal analysis, Methodology, Writing - original draft. **Harm Derksen:** Conceptualization, Methodology, Project administration, Writing - review & editing. **Jonathan Gryak:** Conceptualization, Data curation, Project administration, Supervision, Writing - review & editing. **Cheng Jiang:** Formal analysis. **Zijun Gao:** Formal analysis. **Winston Zhang:** Formal analysis. **Hamid Ghanbari:** Conceptualization, Data curation. **Pujitha Gunaratne:** Conceptualization, Funding acquisition, Project administration. **Kayvan Najarian:** Conceptualization, Methodology, Funding acquisition, Project administration, Supervision, Writing - review & editing.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.bspc.2020.102200.

Acknowledgment

The research work in this paper was funded by Toyota Motor North America.

References

[1] T.A. Lankveld, S. Zeemering, H.J. Crijns, U. Schotten, The ECG as a tool to determine atrial fibrillation complexity, Heart 100 (14) (2014) 1077–1084.
[2] B.J. Schijvenaars, G. van Herpen, J.A. Kors, Intraindividual variability in electrocardiograms, J. Electrocardiol. 41 (3) (2008) 190–196.
[3] B.M. Asl, S.K. Setarehdan, M. Mohebbi, Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal, Artif. Intell. Med. 44 (1) (2008) 51–64.
[4] Z. Mei, X. Gu, H. Chen, W. Chen, Automatic atrial fibrillation detection based on heart rate variability and spectral features, IEEE Access 6 (2018) 53566–53575.
[5] S.-N. Yu, K.-T. Chou, Selection of significant independent components for ECG beat classification, Expert Syst. Appl. 36 (2) (2009) 2088–2096.
[6] X. Jiang, L. Zhang, Q. Zhao, S. Albayrak, ECG arrhythmias recognition system based on independent component analysis feature extraction, in: TENCON 2006. 2006 IEEE Region 10 Conference, IEEE, 2006, pp. 1–4.
[7] E. Naseri, A. Ghaffari, M. Abdollahzade, A novel ICA-based clustering algorithm for heart arrhythmia diagnosis, Pattern Anal. Appl. 22 (2) (2019) 285–297.
[8] R. Ceylan, Y. Özbay, Comparison of FCM, PCA and WT techniques for classification ECG arrhythmias using artificial neural network, Expert Syst. Appl. 33 (2) (2007) 286–295.
[9] R.J. Martis, U.R. Acharya, K. Mandana, A.K. Ray, C. Chakraborty, Application of principal component analysis to ECG signals for automated diagnosis of cardiac health, Expert Syst. Appl. 39 (14) (2012) 11792–11800.
[10] S. Mahmoodabadi, A. Ahmadian, M. Abolhasani, ECG feature extraction using Daubechies wavelets, in: Proceedings of the Fifth IASTED International Conference on Visualization, Imaging and Image Processing, 2005, pp. 343–348.
[11] Q. Zhao, L. Zhang, ECG feature extraction and classification using wavelet transform and support vector machines, in: Neural Networks and Brain, 2005. ICNN&B'05. International Conference on, Vol. 2, IEEE, 2005, pp. 1089–1092.
[12] M. Ashtiyani, S.N. Lavasani, A.A. Alvar, M. Deevband, Heart rate variability classification using support vector machine and genetic algorithm, J. Biomed. Phys. Eng. 8 (4) (2018) 423.
[13] U.R. Acharya, H. Fujita, O.S. Lih, Y. Hagiwara, J.H. Tan, M. Adam, Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network, Inform. Sci. 405 (2017) 81–90.
[14] S. Kiranyaz, T. Ince, M. Gabbouj, Real-time patient-specific ECG classification by 1-D convolutional neural networks, IEEE Trans. Biomed. Eng. 63 (3) (2016) 664–675.
[15] C. Ye, M.T. Coimbra, B.V. Kumar, Arrhythmia detection and classification using morphological and dynamic features of ECG signals, in: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, 2010, pp. 1918–1921.
[16] J.A. Nasiri, M. Naghibzadeh, H.S. Yazdi, B. Naghibzadeh, ECG arrhythmia classification with support vector machines and genetic algorithm, in: Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on, IEEE, 2009, pp. 187–192.
[17] M.H. Song, J. Lee, S.P. Cho, K.J. Lee, S.K. Yoo, Support vector machine based arrhythmia classification using reduced features, Int. J. Control Autom. Syst. 3 (4) (2005) 571–579.
[18] N.O. Ozcan, F. Gurgen, Fuzzy support vector machines for ECG arrhythmia detection, in: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE, 2010, pp. 2973–2976.
[19] Y. Bazi, N. Alajlan, H. AlHichri, S. Malek, Domain adaptation methods for ECG classification, in: 2013 International Conference on Computer Medical Applications (ICCMA), IEEE, 2013, pp. 1–4.
[20] D. Ge, N. Srinivasan, S.M. Krishnan, Cardiac arrhythmia classification using autoregressive modeling, Biomed. Eng. Online 1 (1) (2002) 5.
[21] D.A. Coast, R.M. Stern, G.G. Cano, S.A. Briller, An approach to cardiac arrhythmia analysis using hidden Markov models, IEEE Trans. Biomed. Eng. 37 (9) (1990) 826–836.
[22] R.V. Andreao, B. Dorizzi, J. Boudy, ECG signal analysis through hidden Markov models, IEEE Trans. Biomed. Eng. 53 (8) (2006) 1541–1549.
[23] P.R. Gomes, F.O. Soares, J. Correia, C. Lima, ECG data-acquisition and classification system by using wavelet-domain hidden markov models, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, IEEE, 2010, pp. 4670–4673.
[24] M.G. Tsipouras, D.I. Fotiadis, D. Sideris, Arrhythmia classification using the RR-interval duration signal, in: Computers in Cardiology, 2002, IEEE, 2002, pp. 485–488.
[25] T. Exarchos, M. Tsipouras, D. Nanou, C. Bazios, Y. Antoniou, D. Fotiadis, A platform for wide scale integration and visual representation of medical intelligence in cardiology: the decision support framework, in: Computers in Cardiology, 2005, IEEE, 2005, pp. 167–170.
[26] E.J.d.S. Luz, T.M. Nunes, V.H.C. De Albuquerque, J.P. Papa, D. Menotti, ECG arrhythmia classification based on optimum-path forest, Expert Syst. Appl. 40 (9) (2013) 3561–3573.
[27] S.M. Jadhav, S.L. Nalbalwar, A.A. Ghatol, Modular neural network based arrhythmia classification system using ECG signal data, Int. J. Inf. Technol. Knowl. Manage. 4 (1) (2011) 205–209.
[28] Y. Özbay, R. Ceylan, B. Karlik, A fuzzy clustering neural network architecture for classification of ECG arrhythmias, Comput. Biol. Med. 36 (4) (2006) 376–388.
[29] F.A. Elhaj, N. Salim, A.R. Harris, T.T. Swee, T. Ahmed, Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals, Comput. Methods Programs Biomed. 127 (2016) 52–63.
[30] S.L. Oh, E.Y. Ng, R. San Tan, U.R. Acharya, Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats, Comput. Biol. Med. (2018).
[31] S. Chauhan, L. Vig, Anomaly detection in ECG time signals via deep long short-term memory networks, in: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, IEEE, 2015, pp. 1–7.
[32] Ö. Yildirim, A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification, Comput. Biol. Med. 96 (2018) 189–202.
[33] C. Zhang, G. Wang, J. Zhao, P. Gao, J. Lin, H. Yang, Patient-specific ECG classification based on recurrent neural networks and clustering technique, in: Biomedical Engineering (BioMed), 2017 13th IASTED International Conference on, IEEE, 2017, pp. 63–67.
[34] P. Warrick, M.N. Homsi, Cardiac arrhythmia detection from ECG combining convolutional and long short-term memory networks, Computing 44 (2017) 1.
[35] P. Rajpurkar, A.Y. Hannun, M. Haghpanahi, C. Bourn, A.Y. Ng, Cardiologist-level arrhythmia detection with convolutional neural networks, 2017, arXiv preprint arXiv:1707.01836.
[36] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, A. Gertych, R. San Tan, A deep convolutional neural network model to classify heartbeats, Comput. Biol. Med. 89 (2017) 389–396.
[37] E. Vidal, F. Thollard, C. De La Higuera, F. Casacuberta, R.C. Carrasco, Probabilistic finite-state machines-part I, IEEE Trans. Pattern Anal. Mach. Intell. 27 (7) (2005) 1013–1025.
[38] H. Ney, S. Martin, F. Wessel, Statistical language modeling using leaving-one-out, in: Corpus-Based Methods in Language and Speech Processing, Springer, 1997, pp. 174–207.
[39] L. Saul, F. Pereira, Aggregate and mixed-order Markov models for statistical language processing, 1997, arXiv preprint cmp-lg/9706007.
[40] Y. Esposito, A. Lemay, F. Denis, P. Dupont, Learning probabilistic residual finite state automata, in: International Colloquium on Grammatical Inference, Springer, 2002, pp. 77–91.
[41] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai, Class-based n-gram models of natural language, Comput. Linguist. 18 (4) (1992) 467–479.
[42] J. Hakkinen, J. Tian, N-gram and decision tree based language identification for written words, in: IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01, IEEE, 2001, pp. 335–338.
[43] B. Dorohonceanu, C.G. Nevill-Manning, Accelerating protein classification using suffix trees, in: ISMB, Vol. 2000, 2000, pp. 128–133.
[44] H. Oğul, E.Ü. Mumcuoğlu, SVM-based detection of distant protein structural relationships using pairwise probabilistic suffix trees, Comput. Biol. Chem. 30 (4) (2006) 292–299.

[45] A. Rizwan, A. Zoha, I. Mabrouk, H. Sabbour, A. Al-Sumaiti, A. Alomaniy, M. Imran, Q. Abbasi, A review on the state of the art in atrial fibrillation detection enabled by machine learning, IEEE Rev. Biomed. Eng. (2020).

[46] C.T. January, L.S. Wann, J.S. Alpert, H. Calkins, J.E. Cigarroa, J.B. Conti, P.T. Ellinor, M.D. Ezekowitz, M.E. Field, K.T. Murray, et al., 2014 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, J. Am. Coll. Cardiol. 64 (21) (2014) e1–e76.

[47] P.A. Wolf, R.D. Abbott, W.B. Kannel, Atrial fibrillation as an independent risk factor for stroke: the framingham study, Stroke 22 (8) (1991) 983–988.

[48] D. with the special contribution of the European Heart Rhythm Association (EHRA), E. by the European Association for Cardio-Thoracic Surgery (EACTS), and A.F. Members, A.J. Camm, P. Kirchhof, G.Y. Lip, U. Schotten, I. Savelieva, S. Ernst, I.C. Van Gelder, et al., Guidelines for the management of atrial fibrillation: the task force for the management of atrial fibrillation of the european society of cardiology (ESC), Eur. Heart J. 31 (19) (2010) 2369–2429.

[49] L.A. Orejarena, H. Vidaillet, F. DeStefano, D.L. Nordstrom, R.A. Vierkant, P.N. Smith, J.J. Hayes, Paroxysmal supraventricular tachycardia in the general population, J. Am. Coll. Cardiol. 31 (1) (1998) 150–157.

[50] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220.

[51] F. Nolle, F. Badura, J. Catlett, R. Bowser, M. Sketch, CREI-GARD, a new concept in computerized arrhythmia monitoring systems, Comput. Cardiol. 13 (1986) 515–518.

[52] Z. Li, H. Derksen, J. Gryak, M. Hooshmand, A. Wood, H. Ghanbari, P. Gunaratne, K. Najarian, Markov models for detection of ventricular arrhythmia, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2019, pp. 1488–1491.

[53] E.J.d.S. Luz, W.R. Schwartz, G. Cámara-Chávez, D. Menotti, ECG-based heartbeat classification for arrhythmia detection: A survey, Comput. Methods Programs Biomed. 127 (2016) 144–164.

[54] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, IEEE Trans. Biomed. Eng. (3) (1985) 230–236.

[55] H. Sedghamiz, Matlab implementation of pan tompkins ECG QRS detector, 2014.

[56] Z.F.M. Apandi, R. Ikeura, S. Hayakawa, Arrhythmia detection using MIT-BIH dataset: A review, in: 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), IEEE, 2018, pp. 1–5.

[57] O. Faust, E.J. Ciaccio, U.R. Acharya, A review of atrial fibrillation detection methods as a service, Int. J. Environ. Res. Public Health 17 (9) (2020) 3093.

[58] A.Y. Hannun, P. Rajpurkar, M. Haghpanahi, G.H. Tison, C. Bourn, M.P. Turakhia, A.Y. Ng, Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network, Nat. Med. 25 (1) (2019) 65.

[59] J.H. Tan, Y. Hagiwara, W. Pang, I. Lim, S.L. Oh, M. Adam, R. San Tan, M. Chen, U.R. Acharya, Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals, Comput. Biol. Med. 94 (2018) 19–26.

[60] I.C. Van Gelder, H.F. Groenveld, H.J. Crijns, Y.S. Tuininga, J.G. Tijssen, A.M. Alings, H.L. Hillege, J.A. Bergsma-Kadijk, J.H. Cornel, O. Kamp, et al., Lenient versus strict rate control in patients with atrial fibrillation, New Engl. J. Med. 362 (15) (2010) 1363–1373.

[61] V. Fuster, L.E. Rydén, D.S. Cannom, H.J. Crijns, A.B. Curtis, K.A. Ellenbogen, J.L. Halperin, J.-Y. Le Heuzey, G.N. Kay, J.E. Lowe, et al., ACC/AHA/ESC 2006 guidelines for the management of patients with atrial fibrillation, Circulation 114 (7) (2006) e257–e354.