DENS-ECG: A deep learning approach for ECG signal delineation

Peimankar, Abdolrahman; Puthusserypady, Sadasivan

Published in:
Expert Systems with Applications

Link to article, DOI:
10.1016/j.eswa.2020.113911

Publication date:
2021

Document Version
Peer reviewed version

Citation (APA):
Peimankar, A., & Puthusserypady, S. (2021). DENS-ECG: A deep learning approach for ECG signal delineation. *Expert Systems with Applications*, *165*, Article 113911. https://doi.org/10.1016/j.eswa.2020.113911

DENS-ECG: A Deep Learning Approach for ECG Signal Delineation

Abdolrahman Peimankar, Sadasivan Puthusserypady

Department of Health Technology, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

Abstract

Objectives: With the technological advancements in the field of tele-health monitoring, it is now possible to gather huge amount of electro-physiological signals such as the electrocardiogram (ECG). It is therefore necessary to develop models/algorithms that are capable of analysing these massive amount of data in real-time. This paper proposes a deep learning model for real-time segmentation of heartbeats. *Methods:* The proposed DENS-ECG algorithm, combines convolutional neural network (CNN) and long short-term memory (LSTM) model to detect onset, peak, and offset of different heartbeat waveforms such as the P-waves, QRS complexes, T-waves, and No waves (NW). Using ECG as the inputs, the model learns to extract high level features through the training process, which, unlike other classical machine learning based methods, eliminates the feature engineering step. *Results:* The proposed DENS-ECG model was trained and validated on a dataset with 105 ECG records of length 15 minutes each and achieved an average sensitivity and precision of 97.95% and 95.68%, respectively, using a stratified 5-fold cross validation. Additionally, the model was evaluated on an unseen dataset to examine its robustness in QRS detection, which resulted in a sensitivity of 99.61% and precision of 99.52%. *Conclusion:* The empirical results show the flexibility and accuracy of the combined CNN-LSTM model for ECG signal delineation. *Significance:* This paper proposes an efficient and easy to use approach using deep learning for heartbeat segmentation, which could potentially be used in real-time tele-health monitoring systems.

Keywords: Convolutional neural network (CNN); Deep learning; Electrocardiogram (ECG); Long short-term memory (LSTM); Signal delineation.

1. Introduction

Analysis of electrocardiogram (ECG) signals is one of the most important steps in the diagnosis of cardiac disorders. In order to achieve high diagnostic accuracies, the ECG analysis tools/software require the knowledge about the location and morphology of different segment waveforms (P-QRS-T) in ECG records. For example, atrial fibrillation (AFIB) is one of the most common cardiac arrhythmias in elderly population (Iwasaki et al., 2011; Markides & Schilling, 2003) and P-wave absence is one of the important and clinically useful features for the detection of AFIB (Couceiro et al., 2008; Fukunami et al., 1991). This makes P-wave delineation of great importance in cardiac clinical practice. In addition, most of the developed state-of-the-art algorithms for analysing

Email addresses: apeima@dtu.dk (Abdolrahman Peimankar), sapu@dtu.dk (Sadasivan Puthusserypady)

ECG records and arrhytmias detection are also dependant on the detection of QRS complexes (R-peaks) (Tateno & Glass, 2001; Huang et al., 2010; Peimankar & Puthusserypady, 2018; Andersen et al., 2019; Alonso-Atienza et al., 2012; Hagiwara et al., 2018; Khalaf et al., 2015; Ceylan et al., 2009; Luz et al., 2013; Homaeinezhad et al., 2012). Furthermore, there are other studies that use statistical features extracted from RR intervals for the purpose of different arrhythmias classification (De Chazal et al., 2004; Mar et al., 2011; Ye et al., 2015; Perlman et al., 2015; Khamis et al., 2018; Pławiak, 2018; Martis et al., 2012; Khorrami & Moavenian, 2010). It should be noted that these methods usually utilise the annotated databases such as the Physionet datasets to validate their performance (Goldberger et al., 2000). These methods become extremely cumbersome to test on the newly collected raw ECG signals in real-world applications.

The latest advancements in tele-health monitoring systems provide the opportunity to collect huge amount of ECG data. One of the most common ways for physicians/cardiologists to analyse ECG waveforms is through visual examination of these recordings. However, in most cases, it is difficult and extremely time consuming to analyse such huge amount of data. Subjectivity is another big concern in such approaches. In order to interpret ECG, the morphology of its three most important component waveforms, namely, the P-wave, QRS complex, and T-wave, are to be assessed to help diagnosing different heart diseases. Though, the clinical details of various heart diseases are beyond the scope of this paper, analysing the features of ECG waveforms can help experts to interpret them effectively. For example, in a normal sinus rhythm (NSR), the P-wave is a relatively small wave due to the small muscle mass of atria. The normal P-wave duration is usually less than 0.12 seconds. The most important wave in the QRS complex is the R-wave, which is of high interest in ECG interpretation. The QRS duration for NSR is usually less than 0.1 seconds and an abnormal QRS is wider than 0.12 seconds. The T-wave always follows a QRS complex and is often referred to as the most difficult ECG waveform to assess, which can be easily misinterpreted (Khan, 2008). The duration of the T-wave is usually between 0.1 to 0.25 seconds. Therefore, it is of great interest to develop a reliable software for ECG signal delineation in order to specify the accurate location of different segment waves, such as the P-wave, QRS complex, and T-wave, for subsequent use in clinical diagnostics.

Various state-of-the-art algorithms for ECG delineation have been introduced in the literature. Most of these algorithms are based on classical machine learning and digital signal processing techniques. Martínez et al. (2004) proposed a wavelet based ECG delineation algorithm. This algorithm was validated on two different datasets achieving a high performance. However, this algorithm relies on wavelet transform that requires a threshold to be set in order to achieve a desired results on new datasets. Dubois et al. (2007) used generalised orthogonal forward regression with Gaussian mesa function models for automatic ECG waves extraction. However, the performance of this algorithm is not as good as other state-of-the-art models especially for the detection of smaller waveforms such as T-waves. Lin et al. (2010) proposed a Bayesian model for P- and T- waves detection, which showed higher accuracy compared to previously published algorithms but at a higher computational cost. In another study, Poli et al. (1995) proposed a method based on genetic algorithm for QRS detection. However, evolutionary based methods, such as genetic algorithm, suffer from falling into local optima. Furthermore, there are other wavelet based models (Ieong et al., 2012; Cuiwei Li et al., 1995; Yochum et al., 2016), which also require a threshold to be set to achieve their optimum performance similar to Martínez et al. (2004).

Feature engineering is an essential step in classical machine learning methods where it requires to define specific features from the data and use them as inputs to train the models for classification (LeCun et al., 2015). On the other hand, deep learning models bypass this step by automatically extracting the relevant features directly from the data and outperforms many state-of-the-art models within different fields and applications of machine learning. This is because, deep learning models are capable of extracting highly abstract features from signals without the need for prior domain knowledge and expertise (LeCun et al., 2015; Nielsen, 2015).

The applications of deep learning in tele-health and biomedical engineering are growing exponentially in recent years. For example, these methods were applied successfully to ECG arrhythmias classification (Pourbabaee et al., 2018; Saadatnejad et al., 2020; Özal Yildirim, 2018; Porumb et al., 2020; Acharya et al., 2017; Peimankar & Puthusserypady, 2019) and electroencephalogram (EEG) signal classification (Phan et al., 2018; Kshirsagar & Londhe, 2018; Dose et al., 2018). It has been shown in the literature that a combination of convolutional neural networks (CNN) and long short-term memory (LSTM) can enhance the classification/prediction performance as both the CNN and LSTM learn different (complex) functions from the input signals in the training phase(Geras et al., 2015; Xingjian et al., 2015; Sainath et al., 2015; Petersen et al., 2019; Shi et al., 2016).

In this paper, a novel deep combined CNN-LSTM model, named as the DENS-ECG algorithm, is proposed to automatically extract features from ECG records. These features are subsequently used to distinguish between three main ECG component waveforms (i.e. P-wave, QRS complex, and T-wave) in each heartbeats. In addition, unlike classical ECG peak detection and delineation algorithms, the proposed deep learning model requires only minimum parameters tuning and preprocessing step. In other words, the proposed model can be generalized well to be applied on new datasets in practice. To the best of our knowledge, this is the first study, where a combined CNN-LSTM model is used for ECG signal delineation.

The remainder of this paper consists of 4 sections. In Section 2, the background of the deep learning model and the proposed DENS-ECG algorithm for ECG signal delineation are briefly described. The experimental results are presented in Section 3. Section 4 presents the discussion and comparison with other state-of-the-art methods, followed by the conclusions in Section 5.

2. Materials and Methods

2.1. Dataset

In this study, PhysioNet QT database (QTDB) was used to train and validate the performance of the proposed algorithm (Laguna et al., 1997). The QTDB includes records from seven different databases such as European ST-T database, MIT-BIH Sudden Death database, MIT-BIH Normal Sinus Rhythm database, and MIT-BIH Supraventricular Arrhythmia database. This is a common database with a mix of female and male aged 20 to 88 years. There are 105 records in the QTDB and each has a length of 15 minutes with a sampling frequency of 250 Hz. In addition, the MIT-BIH Arrhythmia Database (MITDB) was used to test the model, which is also a common database with a mix of female and male aged 23 to 89 years. It contains 48 half-hour ECG recordings, which were sampled at 360 Hz (Moody & Mark, 2001). PyhsioNet's WFDB python package (Goldberger et al., 2000) was used to read the signals and their corresponding annotations.

2.2. Pre-processing

First, each record was filtered and segmented accordingly. The signals were zero-phase filtered using a 3rd order Butterworth band-pass (0.5 – 40Hz) filter to remove baseline wanders and high frequency noises (Christiano & Fitzgerald, 2003). The records were then segmented into smaller chunks of 1000 samples. Each chunk of 1000 samples equals to approximately five heartbeats. This helps the model to see a sequence of heartbeats to provide a more robust outcome and learn the underlying features of the different ECG waves effectively. It should be emphasized that 84 out of 105 records of QTDB were used for training the model and the remaining 21 records were used for evaluating the model.

2.3. Deep Learning Model Structure

As discussed, the model used in this study for ECG signal delineation is a combination of two well-known deep network structures. The first part of the model consists of three 1D convolutional layers, which extracts the high abstract features from ECG segments. The second part of the model includes two deep LSTM layers to process the features extracted by the convolutional layers. Lastly, the output of the second LSTM layer is passed through a dense layer with four neurons, which provide the posterior probabilities corresponding to each of the four classes. It is worth noting that the dense layer is a time distributed layer to keep the continuity of the ECG records.

2.3.1. CNN layer

Unlike traditional neural networks, in which each neuron is connected to every neuron in the adjacent layer, CNNs are able to exploit any existing spatial and temporal patterns in the data (LeCun et al., 2015). For this purpose, CNNs take advantage of four key attributes, which are: 1) establishing local connections; 2) shared weights; 3) very large number of layers/filters; and 4) reducing the complexity of the network (LeCun et al., 2015). For example, in 1D CNNs, different filters are defined by sliding a fixed window over the signal. The length of the window used in CNNs for the convolution process is known as the kernel size, denoted as k_{size}. The outputs of these convolutions (between the filters and specific regions of the input signal) are the neurons in the resulted feature maps as illustrated in Figure 1. The weights of these connections and an overall bias are learned during the training process. It should be noted that there is only one set of weights corresponding to each feature map as shown in Figure 1. This convolution process can be expressed as follows (Goodfellow et al., 2016):

$$a_{ij}^{m} = \varphi\left(b_i + \sum_{k=1}^{M} w_{ik} x_{j+k-1}\right) = \varphi(b_i + \mathbf{w}_i^T \mathbf{x}_j), \tag{1}$$

where a_{ij}^{m} is the activation or the output of the jth neuron of the ith filter for the mth convolutional layer, M is the kernel size, φ is the neural activation function, b_i is the shared bias of the ith filter, $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \ldots \quad w_{iM}]^T$ are the shared weights of the ith filter, and $\mathbf{x}_j = [x_j \quad x_{j+1} \quad \ldots \quad x_{j+M-1}]^T$ are the corresponding M inputs. The outputs of the neurons, a's, are the filtered version of the input time series, which learns the same features at different locations as the filter (kernel) slides over the input signal. Applying various filters to the input time series leads to different feature maps in the output of the activation functions. As shown in Figure 1, the number of feature maps are equal to the number of applied filters ($n_{filters}$), which are each defined by a set of M shared weights and a single bias (the biases b_i's are not shown in the figure).

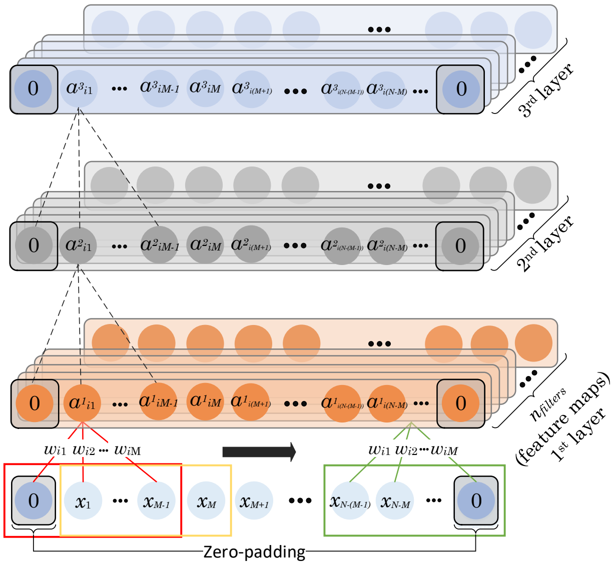

0
a³i1 ··· a³iM-1 a³iM a³i(M+1) ··· a³i(N-(M-1)) a³i(N-M) ···
0
3rd layer
0
a²i1 ··· a²iM-1 a²iM a²i(M+1) ··· a²i(N-(M-1)) a²i(N-M) ···
0
2nd layer
0
a¹i1 ··· a¹iM-1 a¹iM a¹i(M+1) ··· a¹i(N-(M-1)) a¹i(N-M) ···
0
nfilters
(feature maps)
1st layer
wi1 wi2 ··· wiM
wi1 wi2 ··· wiM
0
x1 ··· xM-1 xM xM+1 ··· xN-(M-1) xN-M ···
0
Zero-padding

Figure 1: Schematic diagram of the convolution process. A kernel of size $k_{size} = M$ moves across the input signal. The corresponding weights $(w_{i1}, \ldots, w_{i2}, w_{iM})$ are fixed for all the convolution operation. The signals in each layer are appropriately zero-padded to keep the dimensions same as in the input layer. For example, if kernel size (k_{size}) M is odd, $(M-1)/2$ zeros are padded to each end of the signal, otherwise the zero-padding size is $M/2$.

In general, the computational complexity of a 1D CNN is lower than a 2D CNN. Several vector-matrix operations are performed in the 1D CNN layers. For instance, a 1D convolution is a matrix multiplication of size $N_f \times N_s$, where N_f and N_s are the length of the filter and input signal, respectively (Vaswani et al., 2017). Therefore, the total computational complexity of a 1D CNN layer is approximated by,

$$n_{filters} N_f N_s, \tag{2}$$

where $n_{filters}$, as mentioned above, is the number of filters applied in the corresponding layer.

2.3.2. LSTM layer

Recurrent Neural Networks (RNNs) are designed to work with sequential time-series which are capable of learning dependencies in sequential information. However, it has been shown that learning long-term dependencies are very challenging (Bengio et al., 1994). LSTM networks, a special type of RNNs, are capable of addressing the problem of unstable gradient and can handle long-term dependencies (Hochreiter & Schmidhuber, 1997). As depicted in Figure 2, there are three main parts in a LSTM block: (i) forget gate (f_n), (ii) input gate (i_n), and (iii) output gate (o_n). Forget and output gates are mainly responsible to remove or add information to the memory block in the following way:

$$f_n = \varphi(b_f + \mathbf{u}_f^T \mathbf{a}_n + \mathbf{w}_f^T \mathbf{h}_{n-1}), \tag{3}$$

$$i_n = \varphi(b_i + \mathbf{u}_i^T \mathbf{a}_n + \mathbf{w}_i^T \mathbf{h}_{n-1}), \tag{4}$$

where $\mathbf{a}_n$ is the input sequence to the LSTM at time step n, which is actually the output of the last CNN layer here, and $\mathbf{h}_{n-1}$ is the output sequence at time step $n-1$. The $\mathbf{u}_f$, $\mathbf{w}_f$, $\mathbf{u}_i$, and $\mathbf{w}_i$ represent the weight vectors and b_f and b_i are bias terms. These should be learned in the training phase of the LSTM. In addition, since $0 \leq \varphi(\cdot) \leq 1$, this controls the contribution of each unit in the memory block. Therefore, the memory c_n is updated as:

$$c_n = f_n c_{n-1} + i_n \tilde{c}_n, \tag{5}$$

where

$$\tilde{c}_n = \tanh(b_c + \mathbf{u}_c^T \mathbf{a}_n + \mathbf{w}_c^T \mathbf{h}_{n-1}). \tag{6}$$

Finally, the output vector h_n is computed as:

$$h_n = o_n \tanh(c_n), \tag{7}$$

where

$$o_n = \varphi(b_o + \mathbf{w}_o^T \mathbf{a}_n + \mathbf{u}_o^T \mathbf{h}_{n-1}). \tag{8}$$

Here, $\mathbf{u}_o$ and $\mathbf{w}_o$ are the weight vectors of the output gate, and b_o is the output bias. From (7) and (8), in addition to input and previous output gate, the current memory plays an important role in the output gate. This provides LSTM with the ability to keep or forget the existing memory efficiently (Chung et al., 2014).

Bidirectional LSTM (BiLSTM) is a variant of the LSTM, which can process a sequence of data in both directions. Unlike LSTM, BiLSTM can also exploit the future context (Graves & Jaitly, 2014). It consists of two hidden layers, which are fed forward to the output layer (Graves & Jaitly, 2014). The outputs of BiLSTM are a function of forward and backward pass along with their corresponding weights and biases.

The computational complexity of an LSTM layer depends on the number of hidden units, N_h, and the number of input neurons, N_a. Each LSTM cell performs four matrix-vector multiplications of size $N_a \times N_h$, four matrix-vector multiplications of size $N_h \times N_h$, and several vector operations of size N_h (Saadatnejad et al., 2020). Therefore, the total computational complexity of an LSTM cell can be expressed as,

$$\alpha N_a N_h + \beta N_h N_h + \gamma N_h + \zeta, \tag{9}$$

where α, β, γ, and ζ are constants, which vary based on the cell type. Thus, the computational complexity of a BiLSTM layer can be computed as,

$$2 \times (\alpha N_a N_h + \beta N_h N_h + \gamma N_h + \zeta). \tag{10}$$

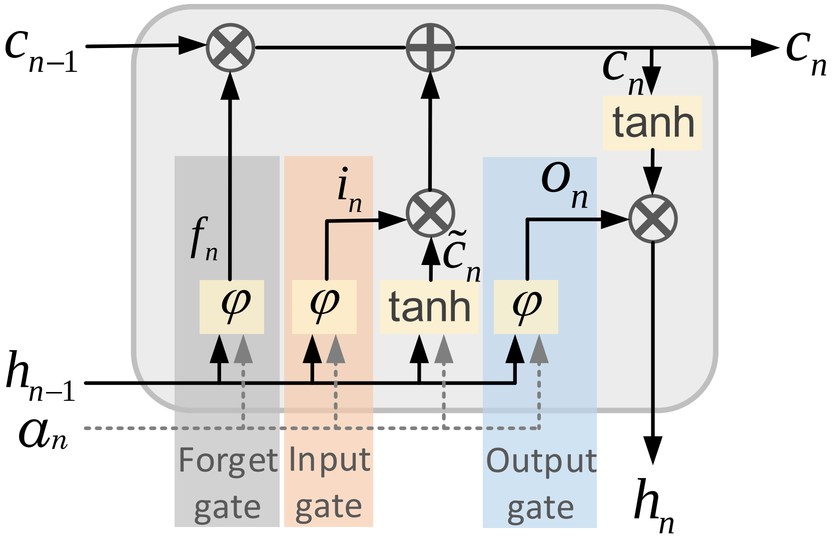


Figure 2: Schematic diagram of LSTM memory block.

2.3.3. Model training

In order to find the optimum network parameters (e.g. weights and biases) and consequently achieve the optimal performance, the network needs to be trained appropriately. It is a non-convex optimisation problem, which can be solved by using a cost function ($\mathcal{J}$) in an iterative process (Goodfellow et al., 2016; Kingma & Ba, 2014; Dauphin et al., 2014). One of the most commonly used cost function for multi-class classification problems is the categorical cross-entropy loss (CCEL), which consists of a softmax and a cross-entropy loss (CEL). The CCEL is utilised to output a probability over the different classes. For this purpose, the class labels are one-hot encoded, which converts all the elements of the label vector into zero except for the true class. The CEL function can be formulated as follows:

$$\mathcal{J}_{CEL} = -\sum_{i=1}^{N_c} y_i \log(z_i), \tag{11}$$

where z_i is the computed score from the network corresponding to the true class, N_c is the number of classes, and y_i is non-zero only for the true class. Thus, the CCEL can be calculated as:

$$\mathcal{J}_{CCEL} = -\sum_{i=1}^{N_c} \log(\sigma(\mathbf{z})_i), \tag{12}$$

where

$$\sigma(\mathbf{z})_i = softmax = \frac{e^{z_i}}{\sum_{j=1}^{N_c} e^{z_j}}, \quad i = 1, \dots, N_c. \tag{13}$$

2.4. Classification

The output of the LSTM layer, which are the extracted features from ECG signals, is fed into a *TimeDistributed* dense layer with four neurons with *softmax* activation functions. The latter ensures a classification per time stamp such that the sum of the neuron outputs is equal to 1, i.e., they can be interpreted as posterior probabilities. The output of the dense layer for the i^{th} sample is classified to one of the four classes (P, QRS, T, or NW) as follows:

$$\hat{y}_i = c_j \quad \text{if} \quad P(y_i = c_j|x_i) = \operatorname{argmax} P(c_1, \ldots, c_k|x_i), \tag{14}$$

where $\hat{y}_i$ is the predicted class for the i^{th} sample x_i and P represents the posterior probability.

2.5. Deep ECG Delineation Framework

125 The proposed deep learning model utilizes the advantages of ensemble learning technique to delineate ECG signals. The flowchart for the proposed DENS-ECG algorithm is illustrated in Figure 3, which is described step by step as follows:

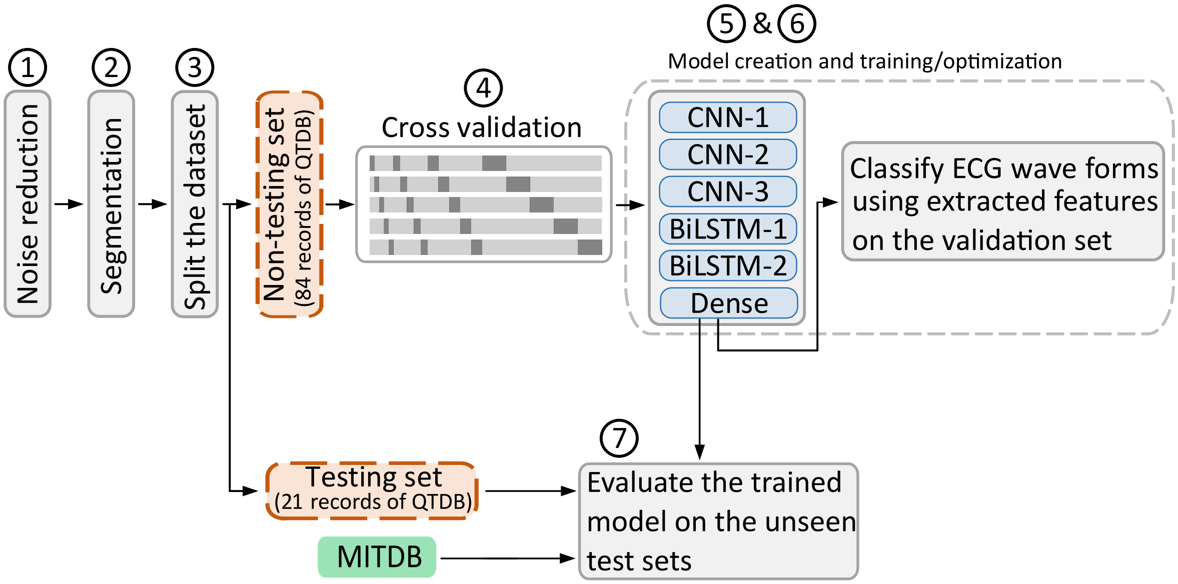


Figure 3: Flowchart of the proposed DENS-ECG algorithm.

1. *Noise reduction*: The ECG signals are filtered to remove noise and baseline wanders.
2. *Segmentation*: In this step, the ECG signals are segmented into chunks of 1000 samples. These segments are then fed into the model as inputs. It should be noted that the continuity of the time series (ECG signal) within each segment is preserved to make it possible for the network to learn the pattern of different waveforms from each input.
3. *Separate the testing set from a non-testing set*: The segmented ECG signals are divided into two sets. The non-testing set is used to train, validate and optimise the P-QRS-T waveforms delineation algorithm and the testing set is considered to evaluate the proposed model. The records in testing and non-testing sets are unique, i.e. no excerpt of the test records is included in the training process of the model.
4. *Cross validation*: The model is trained using 5-fold cross validation technique (Friedman et al., 2001). A stratified 5-fold cross validation (5-fold CV, Figure 4) is used, where the distribution of samples in each fold

is proportional to the size of the corresponding classes in the whole dataset. According to Kohavi (1995), this method of cross validation leads to a more reliable performance in terms of bias and variance compared to traditional cross validation techniques.

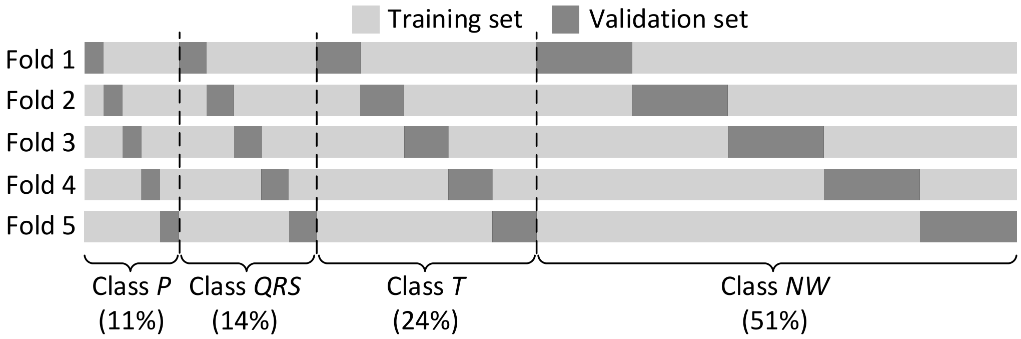
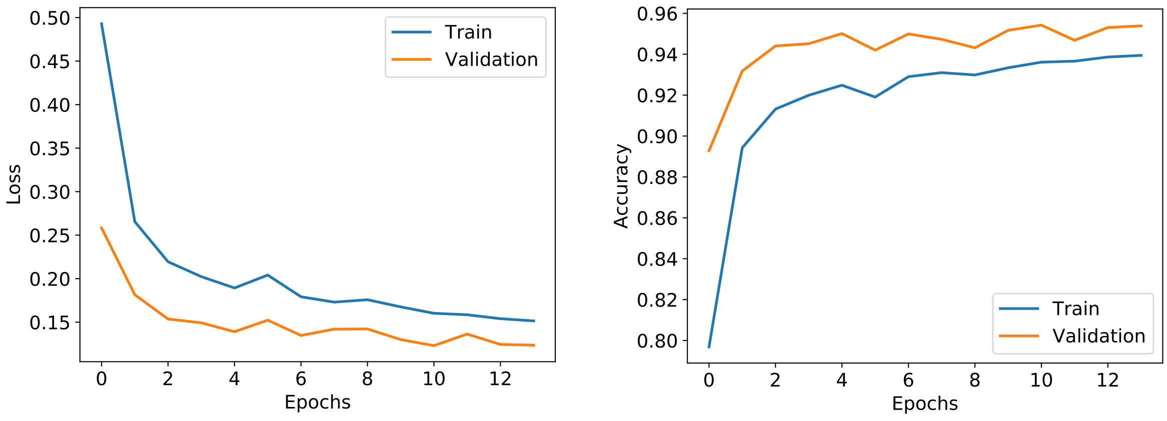


Figure 4: Schematic diagram of a stratified 5-fold cross validation technique. The training and validation sets are split according to the size of the four classes.

5. *Creating the model*: In total, the model has eight layers, which includes the input layer, three 1D convolutional layers followed by two BiLSTM and a dropout layer. Finally, a time distributed wrapper is used for the dense layer, which configures the BiLSTM layer for the sequence prediction. The input segments are fed directly into three successive convolutional layers, which extract the temporal patterns (features) from the ECG signals. A kernel size of $M = 3$ is applied in the three convolutional layers and the corresponding number of filters ($n_{filters}$) are respectively 32, 64, and 128 for the three successive layers. In order to keep the same dimension in the input and convolution layers, zero padding is employed. For example, the output of the first convolutional layer is now a sequence of 32 features with the same dimension of the input signal (time series). This process is repeated for the other two layers. The output of the last convolutional layer is 128 highly abstract feature maps, which are used as inputs for the first BiLSTM layer with $n_{units} = 250$ hidden units. The second BiLSTM layer has $n_{units} = 125$ hidden units. The dropout probability in the dropout layer is set to 0.2. The dropout layer helps to avoid over-fitting problem during the training of the network. The dense layer has 4 hidden units and a *softmax* function is used as an activation function, which assigns a value between 0 and 1 to each sample of the input ECG signals.

6. *Model training and optimisation*: The model is trained using Adam optimisation algorithm (Kingma & Ba, 2014), which is different from the steepest gradient descent (SGD) optimisation algorithm. Adam is used for solving non-convex optimisation problems and is well-suited for large scale network. It has four hyper-parameters, which require to be fine-tuned; 1) the learning rate, α, 2) the exponential decay rate for the first moment estimates, β_1, 3) the exponential decay rate for the second-moment estimates, β_2, and 4) the numerical stability parameter, ϵ. A random search technique is used to find the optimum values of these parameters (Bergstra & Bengio, 2012). This approach is shown to be more efficient compared to the grid search method



(a)

(b)

Figure 5: Training and validation curves of the model for: (a) loss and (b) accuracy.

for hyperparameters optimisation. The proposed DENS-ECG model contains 1,416,044 trainable parameters (weights and biases), which need to be optimised through the learning process. The model is implemented in Python 3.6.4 using *keras* API (Chollet et al., 2015). Figures 5a and 5b show the loss and accuracy of the model during the training phase, respectively. Moreover, an early stopping technique is used to prevent over-fitting of the model. The model stops training if there is no decrease in the value of validation loss after three epochs. Additionally, the dropout process is only applied during the training phase of the model. Therefore, the training becomes more challenging for the network which in turn alleviate over-fitting problem. As illustrated in Figures 5a and 5b, the model achieves higher performance on the validation set than train data, which shows that the model is trained properly without over-fitting.

7. *Evaluate the trained model*: The trained model is then evaluated on the 21 unseen test records from QTDB dataset to examine the performance of the classifier. In addition, the model is tested on the unseen MITDB dataset for QRS detection.

3. Results

In this study, we used the QTDB from PhysioNet to develop the model (Laguna et al., 1997). In total, there are 105 ECG records of which 84 records (80%) are used for training/validation and the remaining 21 records (20%) for testing the model. In addition, the robustness of the proposed model is examined using a different dataset, the MITDB. Since there is only QRS peak annotation available for MITDB, the performance of the proposed DENS-ECG model is reported on detecting the QRS complexes for this dataset.

3.1. Classification Performance Metrics

The key factor in evaluating the performance of any classification system is the capability of the developed model in correctly classifying the new examples. Traditionally, the classification performance of binary problems can be interpreted in a confusion matrix as illustrated in Table 1, which can easily be extended to multi-class problems, as in our case. One of the most commonly used measures to report the performance of classification algorithms is the average accuracy, which can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}. \tag{15}$$

However, in order to report the performance of classifiers on imbalanced datasets, other well-known metrics are used, which can be derived from Table 1 and are formulated as:

$$Sensitivity = \frac{TP}{TP + FN}, \tag{16}$$

$$Precision = \frac{TP}{TP + FP}, \tag{17}$$

$$F - score = (1 + \beta)\frac{Precision \times Sensitivity}{\beta^2 \times Precision + Sensitivity}. \tag{18}$$

When $\beta = 1$, the measure in Eq.(18) is called the balanced F-score (F1-score), which takes both *Precision* and *Sensitivity* into account equally.

Table 1: Confusion matrix.

	Predicted positive	**Predicted negative**
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

Another commonly used qualitative and quantitative metric is the receiver operating characteristics (ROC), which is defined as the ratio between TP rate (Eq. (17)) and FP rate ($FP/(FP + TN)$) (Fawcett, 2006). This method graphically visualises the trade-off between TP rate and FP rate. In case of multi-class classification, each curve actually evaluate the target class versus all the other classes. Beside the curves corresponding to each class, macro- and micro-average curves can also be plotted. The macro- and micro-average of precision and sensitivity are computed as follows (Baeza-Yates et al., 1999):

$$Precision_{micro} = \frac{\sum_{N_c} TP_{N_c}}{\sum_{N_c} TP_{N_c} + \sum_{N_c} FP_{N_c}}, \tag{19}$$

$$Sensitivity_{micro} = \frac{\sum_{N_c} TP_{N_c}}{\sum_{N_c} TP_{N_c} + \sum_{N_c} FN_{N_c}}, \tag{20}$$

$$Precision_{macro} = \frac{\sum_{N_c} Precision_{N_c}}{N_c}, \tag{21}$$

$$Sensitivity_{macro} = \frac{\sum_{N_c} Sensitivity_{N_c}}{N_c}, \tag{22}$$

where N_c is the number of classes as defined earlier.

Furthermore, the area under the curve (AUC) of the ROC curves can be calculated as a scalar metric to evaluate the classification performance. The higher the AUC value, the better the classifier is.

In the following section, the performance of the proposed DENS-ECG model is evaluated using the defined classification metrics. It should be noted that F1-score, Precision (P^+), and Sensitivity (Se) are appropriate metrics for reporting the performance of classification models on imbalanced datasets. The comparison of the proposed method with two other deep learning scenarios is also reported in Section 4.1.

3.2. QRS detection results

The QRS detection performance of the proposed DENS-ECG model on MITDB and QTDB databases are reported in Table 2. It should be noted that the results in this table are corresponding to the unseen 21 records of QTDB and the whole MITDB, which were not used during the training of the model. The proposed model achieves 99.61%, 99.52%, and 99.56% in Se, P^+, and F1-score, respectively, on the well-known MITDB database for QRS detection. The model is also performed well on QTDB for QRS detection. As shown in Table 2, filtering the input signals improves the classification performance substantially. For example, the sensitivity has been increased by more than 3% and 14% on MITDB and QTDB databases, respectively, which confirms the positive effect of filtering. The smaller difference for MITDB database shows that either the R peaks are easier to detect in general or the signal to noise ratio is higher for this database than QTDB database. In addition, the high classification performance on MITDB shows that the proposed model is well generalized, which can be used on different datasets in practice regardless of filtering technique and model parameters. The performance of DENS-ECG model is comparable with other published algorithms (Table 5) and the detailed discussion will be given in Section 4.2.

Table 2: QRS detection performance of the proposed DENS-ECG model for the MITDB and QTDB databases on the test set with and without filtering of the input signals.

Datasets	MITDB			QTDB		
Metrics	Se	P^+	F1-score	Se	P^+	F1-score
Raw signal	96.81	92.01	95.75	85.54	96.24	90.58
Filtered signal	99.61	99.52	99.56	99.7	99.19	99.45

3.3. ECG delineation results

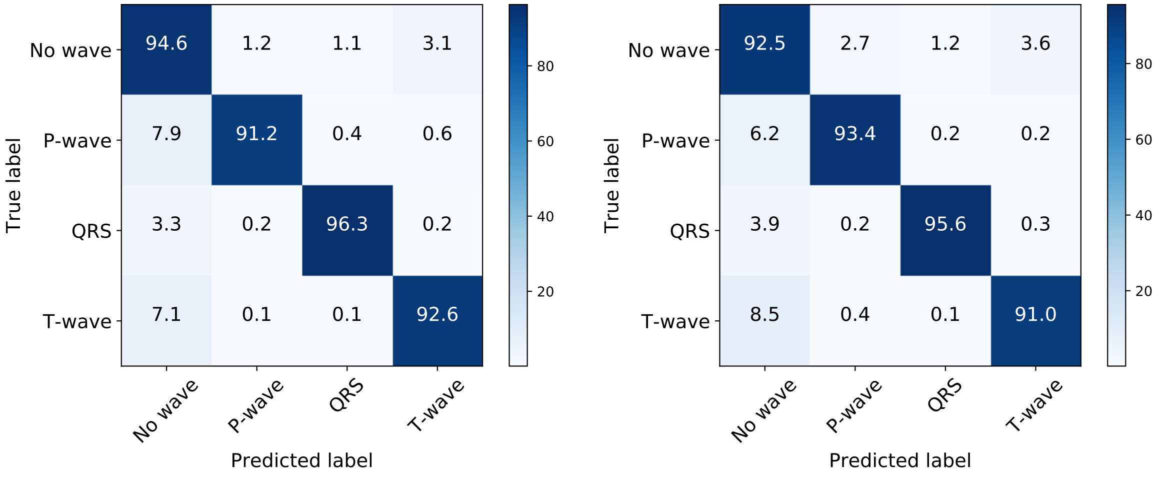
The waveforms delineation performance of DENS-ECG model on QTDB database is given in Table 3. The average performance of the model for detecting start, peak, and end waveforms of each four classes are reported in this table. Overall, the model performs the best on QRS detection followed by T-wave and P-wave, respectively. For example, the precision of T-wave is 5% higher than the P-wave. However, the model sensitivity for P and T-waves detection are comparable to each other, which are more than 96.5% with slightly more favourable for T-wave detection. The F1 score, which represents both Se and P^+, for P, QRS, T, and NW classes equals 93.01%, 99.45%, 96.12%, and 98.55%, respectively.

As shown in Table 3, filtering the input signals improves DENS-ECG performance substantially. For example, the sensitivity of the model has been increased by around 20%, 14%, 13%, and 14% for P, QRS, T, and NW detection. In contrast, the improvement of model precision is not as high as sensitivity when applying filtering to the input signals.

Table 3: Average performance of the proposed DENS-ECG model on the test set for waveform delineation of four classes with and without filtering of the input signals.

Metrics	Se				P^+				F1-score			
Classes	P	QRS	T	NW	P	QRS	T	NW	P	QRS	T	NW
Raw signal	76.80	85.54	82.40	84.39	87.83	96.24	91.43	95.11	81.95	90.58	86.68	89.43
Filtered signal	96.53	99.7	96.81	98.75	89.74	99.19	95.44	98.36	93.01	99.45	96.12	98.55

The confusion matrices of the DENS-ECG model on the 5-fold CV and test set are shown in Figures 6a and 6b. These also confirms that the proposed DENS-ECG model performs better on QRS detection compared to other three classes. Most of the incorrect cases in all three classes (P-wave, QRS, and T-wave) are classified into NW class. In other words, the model does not make incorrect classification between the three main classes (P-wave, QRS, and T-wave). The majority of misclassification cases occur at the start and end of the P-waves, QRS complexes, and T-waves. This is mainly due to the fact that the start and end of these waveforms are adjacent to NW class, which have the same level of amplitude as in NW class and easier to be misclassified with. As an example, 6.2%, 3.9% and 8.5% of P-wave, QRS, and T-wave classes are classified into NW class incorrectly on the test set, respectively. In addition, the small difference between 5-fold CV and test results shows that the model has been trained properly, which does not suffer from over-fitting problem.



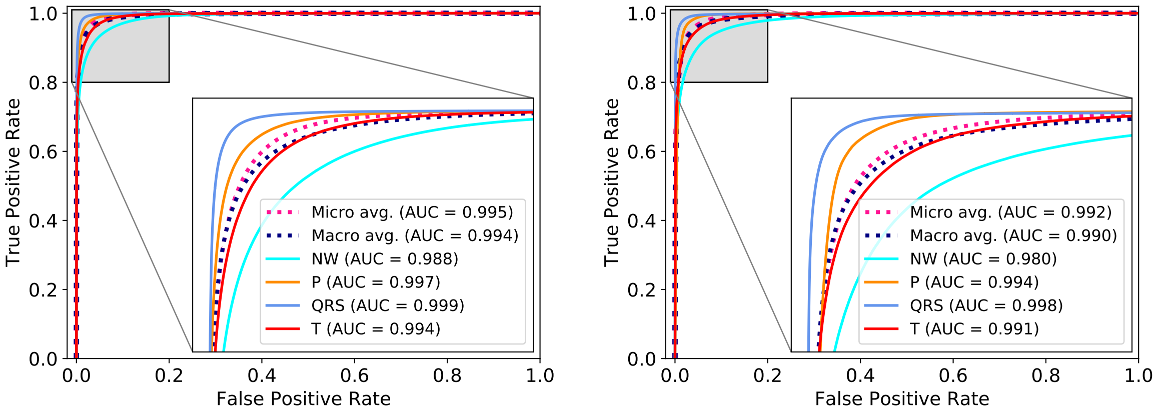
(a)

(b)

Figure 6: Classification performance using confusion matrices for: (a) 5-fold CV and (b) test set of the proposed DENS-ECG algorithm. The numbers are in percentage.

The ROC curve of the DENS-ECG model on the 5-fold CV and test set are plotted in Figures 7a and 7b. As shown in the zoomed areas, the model has the highest AUC on the QRS class compared to other classes. The AUC for the micro and macro-average are 0.992 and 0.99 on the test set, respectively, showing the promising classification performance for the DENS-ECG algorithm.

As an example, Figures 8a and 8b show excerpts of DENS-ECG model predictions and its corresponding true

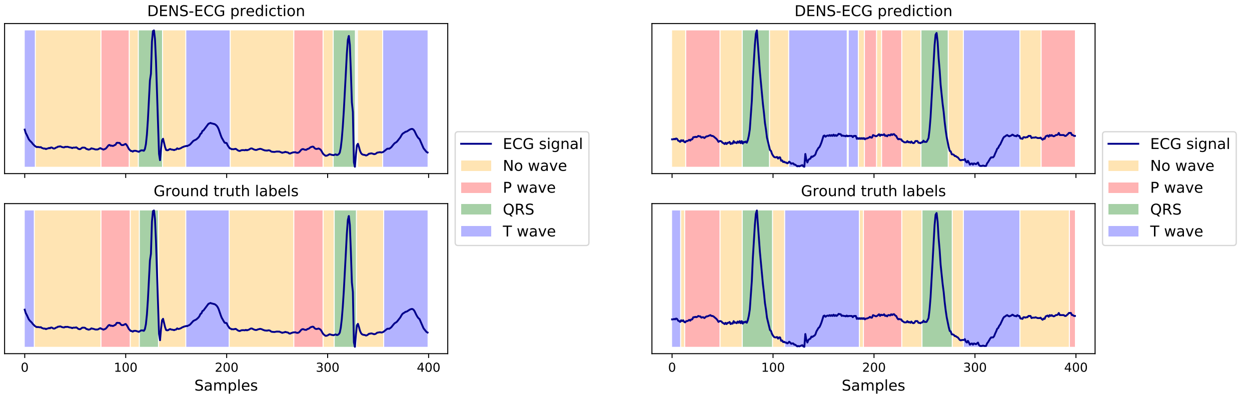


(a)

(b)

Figure 7: ROC curves (solid lines) for four classes using the proposed deep model (DENS-ECG) together with the curves for the micro- and macro-average ROC (dashed lines). (a) 5-fold CV and (b) test set.

annotations (labels). The developed algorithm performance for detecting P, QRS, T, and NW segments confirms the capability of the the deep model on delineation of these waveforms. Figure 8a shows an example of a nearly perfect classification in which all the four waveforms are classified correctly corresponding to the true labels. In contrast, as depicted in Figure 8b, in the first T-wave segment (blue strip), there are some FN predictions. For example, the prediction of the first T wave segment shows a narrow FN detection, which is followed by a correct classification. There is also a similar pattern and some FP predictions for the second P wave segment (red strip). Furthermore, there are some FN predictions for the third P wave segment. In addition, at the beginning of the prediction, there is a FP prediction for a T wave segment.



(a)

(b)

Figure 8: Two excerpts of DENS-ECG model predictions with the corresponding detected P, QRS, T, and NW segments. The true labels (annotations) of the signals are also plotted to compare the results of the classification. (a) A prediction example with high classification rate, (b) A prediction example with some FPs and FNs.

4. Discussion

The proposed DENS-ECG model achieves a good performance on the both MITDB and QTDB databases. The results of the DENS-ECG model show that the proposed model is not only comparable with other state-of-the-art algorithms but also needs less complex parameters tuning to apply on new datasets. The performance of DENS-ECG model is compared with other deep learning models as well as state-of-the-art algorithms for ECG signal delineation.

4.1. Comparison of DENS-ECG with other deep learning approaches

In this section, the performance of the proposed DENS-ECG method is compared with other deep learning models with different number of layers and architectures. This includes models with various number of CNN and BiLSTM layers as well as an end-to-end CNN and BiLSTM model. In all the models, the same parameters and optimisation method as utilised in DENS-ECG are used. As given in Table 4, different combinations of convolutional and BiLSTM layers are examined and their corresponding performance are reported in order to compare with DENS-ECG model, which consists of three convolutional and two BiLSTM layers. It should be noted that the combination of CNN and BiLSTM layers for DENS-ECG model was inspired from our previous research work on the detection of AFIB (Andersen et al., 2019). Unlike DENS-ECG model, which uses convolutional and BiLSTM layers, the end-to-end CNN and LSTM models only use convolutional and BiLSTM layers, respectively, to extract the features and classify the waveforms.

The end-to-end BiLSTM model consists of four layers, which are two BiLSTM layers followed by dropout and a time distributed dense layer. The number of hidden units of BiLSTM layers are the same as DENS-ECG model, which are 250 and 125, respectively. In addition, the dropout value is equal to 0.2 and there are four hidden units in the dense layer. The end-to-end CNN models consists of three convolutional layers followed by a time distributed dense layer. The number of filters in each layer are the same as DENS-ECG model, which are 32, 64, and 128, respectively.

As reported in Table 4, the average F1-score (last column) for all four classes show the better performance of DENS-ECG model with three convolutional and two BiLSTM layers. Although, the performance of the model with two convolutional and two BiLSTM layers is comparable with DENS-ECG model, adding one more convolutional layer leads to the highest possible performance of the model without increasing noticeable complexity to the model. For example, the sensitivity of the DENS-ECG model is higher by around 1% than the model with two convolutional and two BiLSTM layers. This confirms that the DENS-ECG outperforms other architectures for the detection of P-wave, which is an important feature in AFIB detection. The sensitivity of the DENS-ECG model is also generally higher than other deep learning models for the detection of other classes. The end-to-end CNN model has the lowest performance, which achieved 71.83% on average F1-score. The results show that the end-to-end CNN model is especially unable to classify P-waves, which have smaller amplitude compared to other waveforms and more difficult to detect. However, the performance for QRS complex detection is comparable among all models.

On the other hand, the end-to-end BiLSTM model shows better performance than end-to-end CNN model but still lower than other models by around 4% in average F1-score. This shows that the BiLSTM layers have the higher impact on boosting the performance of the delineation model compared to CNN layers, which justifies the use of the combined CNN-LSTM model. Table 4 shows the effectiveness of the proposed DENS-ECG model to extract high abstract temporal features from the ECG signals in order to classify different waveforms. Overall, the performance of the DENS-ECG is higher than other deep learning models for the detection of all waveforms. This comparison shows that the DENS-ECG model architecture is the most optimum among others.

Table 4: Comparison of the classification performance on the test set among DENS-ECG and other deep models architectures.

Metrics	Se				F1-score				Avg. F1-score
Classes	P	QRS	T	NW	P	QRS	T	NW	
1 CNN + 1 BiLSTM	95.79	99.69	97.37	98.48	92.68	99.44	94.37	98.21	96.18
2 CNN + 1 BiLSTM	94.34	99.68	99.51	97.83	92.05	99.45	95.43	98.12	96.26
3 CNN + 1 BiLSTM	95.08	99.65	96.42	98.04	92.35	99.37	95.02	97.97	96.18
2 CNN + 2 BiLSTM	95.48	99.69	96.58	98.12	92.64	99.48	96.44	98.31	96.72
3 CNN	46.54	94.13	85.51	90.26	46.77	88.71	67.74	84.09	71.83
2 BiLSTM	88.12	96.37	91.39	95.04	87.30	95.24	92.62	94.62	92.45
DENS-ECG	**96.53**	**99.70**	**96.81**	**98.75**	**93.01**	**99.45**	**96.12**	**98.55**	**96.78**

4.2. Comparison of DENS-ECG with other state-of-the-art methods

The proposed DENS-ECG model is evaluated on the test set (21 records) of QTDB database for ECG waveform delineation. The model is also compared with other ECG delineation algorithms as given in Table 5.

It should be taken into account that the number of heartbeats (annotations) used for evaluating DENS-ECG in Table 5 is less than other methods since only 21 out of 105 records are used as test set. Although, the performance of DENS-ECG model is not as high as the algorithm reported in Martínez et al. (2004), Bote et al. (2018), and Di Marco & Chiari (2011), it outperforms other algorithms published in Vila et al. (2000) and Dubois et al. (2007) in delineation of P- and T-wave. As it can be seen from Table 5, the capability of the DENS-ECG model in QRS_{on} and QRS_{end} detection is comparable with other models with sensitivity equals to 99.75% and 99.36% for QRS_{on} and QRS_{end}, respectively. The best QRS_{on} and QRS_{end} detection was reported in Di Marco & Chiari (2011), which shows the perfect classification followed by Martínez et al. (2004) and DENS-ECG. Most of the models in Table 5 achieve higher performance in the delineation of P-wave compared to DENS-ECG. For instance, Bote et al. (2018) achieves the highest precision (P^+) for P_{peak} detection equals to 95.11%. However, the performance of the DENS-ECG model in the detection of P_{peak} is comparable with the other best two models published in Martínez et al. (2004) and Di Marco & Chiari (2011). The performance of the DENS-ECG model for the detection of T_{end} is comparable with the model introduced by Bote et al. (2018) and higher than Vila et al. (2000) and Dubois et al. (2007), while it is outperformed by the other two models.

Table 5: Comparison of P, QRS, and T waves detection on QTDB dataset between DENS-ECG and other state-of-the-art methods. (N/R: Not Reported, N/A: Not Applicable)

Methods	Parameters	P_{on}	P_{peak}	P_{end}	QRS_{on}	QRS_{end}	T_{peak}	T_{end}
Martínez et al. (2004)	# beats	3194	3194	3194	3623	3623	3542	3542
	Se	98.87	98.87	98.75	99.97	99.97	99.77	99.77
	P^+	91.03	91.03	91.03	N/A	N/A	97.79	97.79
Vila et al. (2000)	# beats	3194	3194	3194	3623	3623	3542	3542
	Se	N/A	N/A	N/A	N/A	N/A	92.6	92.6
	P^+	N/A	N/A	N/A	N/A	N/A	N/R	N/R
Dubois et al. (2007)	# beats	3194	3194	3194	3623	3623	3542	3542
	Se	91.2	91.2	91.2	N/R	N/R	93.6	93.6
	P^+	N/R	N/R	N/R	N/A	N/A	N/R	N/R
Bote et al. (2018)	# beats	3194	3194	3194	3623	3623	3542	3542
	Se	98.12	99.15	99.87	99.50	99.50	99.41	96.98
	P^+	94.26	95.11	96.03	99.78	99.78	98.96	95.98
Di Marco & Chiari (2011)	# beats	3194	3194	3194	3623	3623	3542	3542
	Se	98.15	98.15	98.15	100	100	99.72	99.77
	P^+	91.00	91.00	91.00	N/A	N/A	97.76	97.76
DENS-ECG	# beats	761	761	761	851	851	834	834
	Se	95.49	97.69	96.41	99.75	99.36	97.71	95.87
	P^+	88.77	90.84	89.06	N/A	N/A	96.51	94.43

Table 6: Comparison of QRS detection on MITDB dataset among DENS-ECG and other state-of-the-art methods.

Methods	# beats	TP	FP	FN	Err. (%)	Se	P^+
Martínez et al. (2004)	109428	109208	153	220	0.34	99.80	99.86
Kim & Shin (2016)	109481	109146	137	135	0.43	99.69	99.88
Pan & Tompkins (1985)	109809	109532	507	277	0.71	99.75	99.54
Poli et al. (1995)	109963	109522	545	441	0.90	99.60	99.50
DENS-ECG	**109494**	**109066**	**525**	**428**	**0.87**	**99.61**	**99.52**

The proposed DENS-ECG model can be considered as a fully automated data-driven method, which does need minimum parameters tuning to be used in practice. Furthermore, the model is generalized enough to be applied on 305 different datasets in practice. As reported in Section 3.2 and Table 6, the model performance on MITDB dataset shows its capability in analyzing unseen datasets. In terms of required preprocessing step in DENS-ECG model, it should be also noted that the filtering of MITDB dataset is identical to what was used for training the model on QTDB dataset.

As shown in Table 6, the results of DENS-ECG model in QRS detection is comparable with other algorithms (Se=99.61% and P$^+$=99.52%). The wavelet-based model introduced by Martínez et al. (2004) has the highest performance for both sensitivity and precision (Se=99.8% and P$^+$=99.86%) compared to other algorithms in Table 6 followed by the model proposed by Kim & Shin (2016). From Table 6, the performance of proposed DENS-ECG model is comparable with the well-known Pan & Tompkins (1985) algorithm for QRS detection. As an example, the precision of the DENS-ECG model is equal to 99.52% compared to 99.54% reported in Pan & Tompkins (1985). Furthermore, the model performs slightly better than QRS detection algorithm proposed in Poli et al. (1995).

To use the DENS-ECG model in real-time, a buffer of the previous heartbeats (samples) is needed. The model receives 1000 samples as input, which is equal to four seconds of ECG recorded at 250 Hz. Since the heart rate for an adult is normally between 60 to 80, one complete heartbeat is expected to be seen per second by the model. Therefore, the data can be streamed as an array of 1000 samples, which consists of approximately three previous heartbeats (buffer of 750 samples) plus the current one second (250 samples). It should be noted that the model prediction processing time for 1000 samples is less than one second, which paves the way to use the DENS-ECG model in real-time.

Although the performance of DENS-ECG model is comparable with other state-of-the-art works, there are still some aspects of the model that should be investigated more in the future studies. For example, the proposed model was validated on standard databases, which are relatively less noisy compared to ECGs captured by Holter monitors. Thus, more complex filtering approaches will be needed to overcome this problem when using Holter monitor devices. Furthermore, it was shown that the model is generalized enough to perform well on the unseen records and databases during the validation phase. However, one of the main limitations of the proposed model is the lack of more diverse databases, which have a wider range of ECG morphologies. It is not guaranteed that the model performs best on the new ECG records with previously unseen arrhythmias and different ECG morphologies. Therefore, the model may be required to be retrained, if necessary. To overcome this issue, more arrhythmias, other than the ones existed in QTDB database, should be added to the training set in the future studies.

5. Conclusion

One of the most challenging tasks in ECG waveform delineation has been the detection of P, QRS, and T waves. In this paper, a deep learning approach, named as the DENS-ECG, which combines the CNN-LSTM networks was proposed to predict the ECG waveforms. The laborious feature extraction step was omitted and the filtered ECG segments were directly used as inputs for training the model. The deep CNN-LSTM network was utilised to extract highly abstract temporal features from 1D ECG signals. These features were then used to classify four different waveforms (P, QRS, T and NW). The model was trained using stratified 5-fold cross validation technique. Finally, the trained model was tested on a completely unseen test sets to evaluate the performance of the classification algorithms. The outputs of the model at each time stamp were the posterior probabilities assigned to the four classes. The proposed model shows a high performance on the test sets with an average F1-score of 99.56% and 96.78% on the MITDB and QTDB datasets, respectively. The efficacy of the proposed DENS-ECG model in detecting ECG waveforms provides us with the opportunity to use this algorithm *in house* by cardiologists to analyze ECG recordings in order to diagnose cardiac arrhythmias such as AFIB.

Acknowledgements

The authors would like to thank the support given by the Innovation Fund Denmark (IFD) under Project No. 6153-00009B.

References

Acharya, U. R., Fujita, H., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adam, M. (2017). Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Information Sciences*, *415*, 190–198.

Alonso-Atienza, F., Rojo-Alvarez, J. L., Rosado-Munoz, A., Vinagre, J. J., Garcia-Alberola, A., & Camps-Valls, G. (2012). Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. *Expert Systems with Applications*, *39*, 1956–1967.

Andersen, R. S., Peimankar, A., & Puthusserypady, S. (2019). A deep learning approach for real-time detection of atrial fibrillation. *Expert Systems with Applications*, *115*, 465–473.

Baeza-Yates, R., Ribeiro-Neto, B. et al. (1999). *Modern information retrieval* volume 463. ACM press New York.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*, 157–166.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*, 281–305.

Bote, J. M., Recas, J., Rincón, F., Atienza, D., & Hermida, R. (2018). A modular low-complexity ECG delineation algorithm for real-time embedded systems. *IEEE Journal of Biomedical and Health Informatics*, *22*, 429–441.

Ceylan, R., Özbay, Y., & Karlik, B. (2009). A novel approach for classification of ECG arrhythmias: Type-2 fuzzy clustering neural network. *Expert Systems with Applications*, *36*, 6721 – 6726.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras.

Christiano, L. J., & Fitzgerald, T. J. (2003). The band pass filter. *international economic review*, *44*, 435–465.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, .

Couceiro, R., Carvalho, P., Henriques, J., Antunes, M., Harris, M., & Habetha, J. (2008). Detection of atrial fibrillation using model-based ECG analysis. In *2008 19th International Conference on Pattern Recognition* (pp. 1–5). IEEE.

Cuiwei Li, Chongxun Zheng, & Changfeng Tai (1995). Detection of ECG characteristic points using wavelet transforms. *IEEE Transactions on Biomedical Engineering*, *42*, 21–28.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems* (pp. 2933–2941).

De Chazal, P., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE transactions on biomedical engineering*, *51*, 1196–1206.

Di Marco, L. Y., & Chiari, L. (2011). A wavelet-based ECG delineation algorithm for 32-bit integer online processing. *Biomedical engineering online*, *10*, 23.

Dose, H., Møller, J. S., Iversen, H. K., & Puthusserypady, S. (2018). An end-to-end deep learning approach to MI-EEG signal classification for BCIs. *Expert Systems with Applications*, *114*, 532–542.

Dubois, R., Maison-Blanche, P., Quenet, B., & Dreyfus, G. (2007). Automatic ECG wave extraction in long-term recordings using Gaussian mesa function models and nonlinear probability estimators. *computer methods and programs in biomedicine*, *88*, 217–233.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*, 861–874.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* volume 1. Springer series in statistics New York.

Fukunami, M., Yamada, T., Ohmori, M., Kumagai, K., Umemoto, K., Sakai, A., Kondoh, N., Minamino, T., & Hoki, N. (1991). Detection of patients at risk for paroxysmal atrial fibrillation during sinus rhythm by P wave-triggered signal-averaged electrocardiogram. *Circulation*, *83*, 162–169.

Geras, K. J., Mohamed, A.-r., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., & Sutton, C. (2015). Blending LSTMs into CNNs. *arXiv preprint arXiv:1511.06433*, .

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). Physiobank, Physiotoolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation*, *101*, e215–e220.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (pp. 1764–1772).

Hagiwara, Y., Fujita, H., Oh, S. L., Tan, J. H., San Tan, R., Ciaccio, E. J., & Acharya, U. R. (2018). Computer-aided diagnosis of atrial fibrillation based on ECG signals: a review. *Information Sciences*, *467*, 99–114.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

Homaeinezhad, M., Atyabi, S., Tavakkoli, E., Toosi, H., Ghaffari, A., & Ebrahimpour, R. (2012). Ecg arrhythmia recognition via a neuro-SVM–KNN hybrid classifier with virtual QRS image-based geometrical features. *Expert Systems with Applications*, *39*, 2047 – 2058.

Huang, C., Ye, S., Chen, H., Li, D., He, F., & Tu, Y. (2010). A novel method for detection of the transition between atrial fibrillation and sinus rhythm. *IEEE Transactions on Biomedical Engineering*, *58*, 1113–1119.

Ieong, C., Mak, P., Lam, C., Dong, C., Vai, M., Mak, P., Pun, S., Wan, F., & Martins, R. P. (2012). A 0.83-μw qrs detection processor using quadratic spline wavelet transform for wireless ECG acquisition in 0.35- μm cmos. *IEEE Transactions on Biomedical Circuits and Systems*, *6*, 586–595.

Iwasaki, Y.-k., Nishida, K., Kato, T., & Nattel, S. (2011). Atrial fibrillation pathophysiology: implications for management. *Circulation*, *124*, 2264–2274.

Khalaf, A. F., Owis, M. I., & Yassine, I. A. (2015). A novel technique for cardiac arrhythmia classification using spectral correlation and support vector machines. *Expert Systems with Applications*, *42*, 8361 – 8368.

Khamis, H., Chen, J., Redmond, J. S., & Lovell, N. H. (2018). Detection of atrial fibrillation from RR intervals and PQRST morphology using a neural network ensemble. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5998–6001). IEEE.

Khan, M. G. (2008). *Rapid ECG interpretation*. Springer Science & Business Media.

Khorrami, H., & Moavenian, M. (2010). A comparative study of DWT, CWT and DCT transformations in ECG arrhythmias classification. *Expert Systems with Applications*, *37*, 5751 – 5757.

Kim, J., & Shin, H. (2016). Simple and robust realtime QRS detection algorithm based on spatiotemporal characteristic of the qrs complex. *PloS one*, *11*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, .

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *14th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1137–1143). volume 2.

Kshirsagar, G., & Londhe, N. (2018). Improving performance of devanagari script input-based P300 speller using deep learning. *IEEE Transactions on Biomedical Engineering*, *66*, 2992–3005.

Laguna, P., Mark, R. G., Goldberg, A., & Moody, G. B. (1997). A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ECG. In *Computers in cardiology 1997* (pp. 673–676). IEEE.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*, 436–444.

Lin, C., Mailhes, C., & Tourneret, J.-Y. (2010). P-and T-wave delineation in ECG signals using a bayesian approach and a partially collapsed Gibbs sampler. *IEEE transactions on biomedical engineering*, *57*, 2840–2849.

Luz, E. J. D. S., Nunes, T. M., De Albuquerque, V. H. C., Papa, J. P., & Menotti, D. (2013). ECG arrhythmia classification based on optimum-path forest. *Expert Systems with Applications*, *40*, 3561–3573.

Mar, T., Zaunseder, S., Martínez, J. P., Llamedo, M., & Poll, R. (2011). Optimization of ECG classification by means of feature selection. *IEEE transactions on Biomedical Engineering*, *58*, 2168–2177.

Markides, V., & Schilling, R. J. (2003). Atrial fibrillation: classification, pathophysiology, mechanisms and drug treatment. *Heart*, *89*, 939–943.

Martínez, J. P., Almeida, R., Olmos, S., Rocha, A. P., & Laguna, P. (2004). A wavelet-based ECG delineator: evaluation on standard databases. *IEEE Transactions on biomedical engineering*, *51*, 570–581.

Martis, R. J., Acharya, U. R., Mandana, K., Ray, A., & Chakraborty, C. (2012). Application of principal component analysis to ECG signals for automated diagnosis of cardiac health. *Expert Systems with Applications*, *39*, 11792 – 11800.

Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, *20*, 45–50.

Nielsen, M. A. (2015). *Neural networks and deep learning* volume 2018. Determination press San Francisco, CA, USA:.

Pan, J., & Tompkins, W. J. (1985). A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering*, (pp. 230–236).

Peimankar, A., & Puthusserypady, S. (2018). Ensemble learning for detection of short episodes of atrial fibrillation. In *2018 26th European Signal Processing Conference (EUSIPCO)* (pp. 66–70). IEEE.

Peimankar, A., & Puthusserypady, S. (2019). An ensemble of deep recurrent neural networks for P-wave detection in electrocardiogram. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1284–1288). IEEE.

Perlman, O., Katz, A., Amit, G., & Zigel, Y. (2015). Supraventricular tachycardia classification in the 12-lead ECG using atrial waves detection and a clinically based tree scheme. *IEEE journal of biomedical and health informatics*, *20*, 1513–1520.

Petersen, N. C., Rodrigues, F., & Pereira, F. C. (2019). Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Systems with Applications*, *120*, 426–435.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., & De Vos, M. (2018). Joint classification and prediction CNN framework for automatic sleep stage classification. *IEEE Transactions on Biomedical Engineering*, *66*, 1285–1296.

Pławiak, P. (2018). Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system. *Expert Systems with Applications*, *92*, 334–349.

Poli, R., Cagnoni, S., & Valli, G. (1995). Genetic design of optimum linear and nonlinear QRS detectors. *IEEE Transactions on Biomedical Engineering*, *42*, 1137–1141.

Poli, R., Cagnoni, S., & Valli, G. (1995). Genetic design of optimum linear and nonlinear QRS detectors. *IEEE Transactions on Biomedical Engineering*, *42*, 1137–1141.

Porumb, M., Stranges, S., Pescapè, A., & Pecchia, L. (2020). Precision medicine and artificial intelligence: A pilot study on deep learning for hypoglycemic events detection based on ECG. *Scientific Reports*, *10*, 1–16.

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*, 2095–2104.

Saadatnejad, S., Oveisi, M., & Hashemi, M. (2020). LSTM-based ECG classification for continuous monitoring on personal wearable devices. *IEEE Journal of Biomedical and Health Informatics*, *24*, 515–523.

Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4580–4584). IEEE.

Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, *39*, 2298–2304.

Tateno, K., & Glass, L. (2001). Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of RR and ΔRR intervals. *Medical and Biological Engineering and Computing*, *39*, 664–671.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vila, J. A., Gang, Y., Presedo, J. M. R., Fernández-Delgado, M., Barro, S., & Malik, M. (2000). A new approach for TU complex characterization. *IEEE Transactions on Biomedical Engineering*, *47*, 764–772.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802–810).

Ye, C., Kumar, B. V., & Coimbra, M. T. (2015). An automatic subject-adaptable heartbeat classifier based on multiview learning. *IEEE journal of biomedical and health informatics*, *20*, 1485–1492.

Özal Yildirim (2018). A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. *Computers in Biology and Medicine*, *96*, 189 – 202.

Yochum, M., Renaud, C., & Jacquir, S. (2016). Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT. *Biomedical Signal Processing and Control*, *25*, 46 – 52.