IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 29, 2019, accepted November 12, 2019, date of publication November 25, 2019,
date of current version December 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955738

Inter-Patient CNN-LSTM for QRS Complex Detection in Noisy ECG Signals

BROSNAN YUEN, XIAODAI DONG, (Senior Member, IEEE), AND TAO LU
Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada

Corresponding author: Xiaodai Dong (xdong@ece.uvic.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2018-03778, and in part by the Undergraduate Student Research Awards.

ABSTRACT In this paper, a convolutional neural network (CNN) with long short-term memory (LSTM) is designed to detect QRS complexes in noisy electrocardiogram (ECG) signals. The CNN performs feature extraction while the LSTM determines the QRS complex timings. A multi-layer perception (MLP) after the LSTM is added to format the QRS complex detection predictions. With a unique data preparation procedure that includes proper design of training dataset, the proposed CNN-LSTM can achieve superior inter-patient testing performance, which means the testing and training datasets do not share any same patient ECG records. This generalization ability characteristic is critical to automated ECG analysis in an age of big data collected from noisy wearable ECG devices. The MIT-BIH and the European ST-T noise stress test databases are used to validate the effectiveness of the proposed algorithm in terms of sensitivity (recall), positive predictive value (precision), F_1 score and timing root mean square error of R peak positions.

INDEX TERMS Artificial neural networks, electrocardiogram (ECG), QRS complex, feedforward neural networks, multi-layer neural network, convolutional neural networks, recurrent neural networks.

I. INTRODUCTION

Electrocardiogram (ECG) is the most important and prevalent tool in diagnosing cardiovascular diseases. With the advancement of wearable technology, Internet of things (IoT) and mobile health, mobile wearable ECG for real-time long-term monitoring becomes increasingly possible anywhere and anytime in patients' hands. The direct result is that vast amounts of ECG data will be generated. The sheer volume of ECG recordings is prohibitive for cardiologists to handle. Therefore, accurate and automated ECG analysis is in urgent need to process the explosively growing number ECG recordings collected by wearable devices.

Computer aided ECG analysis is a field that has been developed for more than four decades. Numerous algorithms were devised and proposed for QRS complex detection and heartbeat classification in the literature [1], [2] and references therein. QRS complex detection is the critical first step, as QRS complex is the most prominent portion of a heartbeat signal and its detection facilitates the subsequent ECG analysis. In addition to heartbeat classification, basic parameters, such as RR, QT, PR intervals, etc., derived after QRS detection, are required for every ECG recording and reveals important information about heart functions. Therefore, literature is abundant with QRS complex detection. Techniques used in QRS complex detection range from signal derivative and digital filters [3]–[7], wavelet transforms [8]–[12], Hilbert transforms [13]–[15], matched filters [16], [17], compressed sensing [18], [19], to machine learning and neural networks (NN) approaches [20]–[28]. Among the many classical derivative and digital filter algorithms after the first Pan and Tompkins method [3], GQRS [7] is a simple one with superior performance by using adaptive search intervals and amplitude thresholds. Reference [10] uses wavelet transform and dynamic amplitude thresholding for QRS complex detection. The wavelet transform eliminates noise and other peaks from the ECG recordings, after which the generated pulse trains are scanned for the QRS complex peaks using the dynamic amplitude thresholding. This method has the advantage of being easy to implement and not needing a training phase. However, the wavelet transform uses a fixed filter pattern, which has the disadvantage of not adapting to different types of QRS complexes. Similarly, papers [29]–[31] employ noise filtering techniques to extract QRS complexes.

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Angiulli.

A quadratic filter with dynamic amplitude thresholding is constructed in [31] for QRS complex detection, which has the same advantages and disadvantages of the wavelet transform filter.

There is a long history of using neural networks for ECG analysis. ECG signals are non-linear and non-stationary in nature, and hence methods that can adapt to changes are needed. Neural networks have such potential. Advancements in neural networks lead to new opportunities and design. Recently, Zihlmann *et al.* [32] propose a convolutional neural network (CNN) followed by a long short-term memory (LSTM) network for ECG disease classification. Jun *et al.* [33] claim that a CNN with a fully connected layer classifies arrhythmia in ECG recordings. Rajpurkar *et al.* [34] developed a 34-layer CNN for detecting arrhythmias in arbitrary length ECG time-series. For applying neural networks to QRS detection, [23] implements the first multi-layer perceptron (MLP) for QRS complex matched filtering. In [25], Xiang et al. utilize a CNN followed by a dense layer for QRS complex detection. The CNN filters the ECG signal, while the dense layer predicts the QRS complexes. The CNN has the advantage of adapting to different types of QRS complexes, but it does not directly predict the timing information of R peaks. Paper [26] segments the QRS complexes by removing the regions outside of the QRS complexes using the first CNN. Then the second CNN finds the starts and ends of the QRS complexes. Paper [27] implements an MLP with radial basis functions for QRS complex detection. Radial basis functions are better at filtering noise when compared to the regular sigmoid functions.

Despite of significant efforts, there are still unsolved challenges of QRS complex detection. First, when heavy noise, motion artifact and baseline wanders are present, robust algorithms are yet to be developed. In wearable device-based ECG measurements, signals can often be very noisy. Second, QRS complex varies from person to person and even within one person's recording. For training-based methods such as NN, detection of new records not previously in the training dataset leads to unsatisfied performance. As mobile wearable ECG adoption increases, many patients' data are not labeled and not included in the training database. To address these challenges, this paper proposes, for the first time according to the authors' knowledge, a CNN-LSTM for QRS complex detection with the objectives of not only high classification accuracy but also small timing error. Moreover, the CNN-LSTM model developed has the ability to generalize to new patients' records. The CNN captures visual patterns and filters noises, while the LSTM detects timings of the QRS complexes. After that, an MLP formats the timing predictions and outputs the final QRS complex detection result. Finally, this paper performs inter-patient testing on the CNN-LSTM by training and testing on different ECG patient recordings. Inter-patient testing verifies the CNN-LSTM's generalization ability.

The rest of the paper is organized as follows. Section II discusses several related QRS complex detection algorithms in detail. Section III on data preparation shows the inter-patient test environment and the test parameters. The proposed CNN-LSTM neural network is presented in Section IV and the simulations section, Section V, compares the performance metrics of the CNN-LSTM to other QRS complex detection algorithms. Error analysis of CNN-LSTM is conducted in Section VI. Conclusions are given in Section VII. Finally, a review of neural networks is found in the Appendix/Section VII.

II. RELATED QRS COMPLEX DETECTION ALGORITHMS

In this section, the following related QRS complex detection algorithms are presented: Pan and Tompkins [3], GQRS [7], Wavedet [8], Xiang et al.'s CNN [25], and Chandra et al.'s CNN [28]. The advantages and disadvantages of each algorithm are also described.

A. PAN AND TOMPKINS

The Pan and Tompkins algorithm [3] is the first real time QRS complex detection algorithm, in which a bandpass filter is applied to reduce the noises in the ECG signals, and adaptive filters are used to detect the QRS complexes. The adaptive filters consist of an amplitude filter, a slope filter, and a width filter. In order to be marked as a QRS complex, an ECG peak must simultaneously meet all of the following criteria: the peak's amplitude must be greater than an amplitude threshold, the peak's slope must be greater than a slope threshold, and the peak's width must fall within the range of a QRS complex width. The amplitude filter rejects the low amplitude signals, while the slope filter and the width filter eliminate the P waves and T waves. The advantages of the Pan and Tompkins algorithm are the fast processing times and low complexity. However, the filters used in the algorithm need to be engineered by hand, which requires a lot of time and expertise. Furthermore, the handcrafted filters can not adapt to different patients and environments.

B. GQRS

GQRS [7] is a classical QRS complex detection algorithm. Firstly, it calculates the means and the standard deviations of the RR intervals and the QRS complex amplitudes of the previously detected QRS. Secondly, the algorithm forms an adaptive search interval using the statistics of the RR intervals. Thirdly, the model creates an adaptive amplitude filter using the statistics of the QRS complex amplitudes. Finally, the adaptive amplitude filter is applied to the current adaptive search interval in order to detect the QRS complex. GQRS has the advantage of adapting slightly better than the Pan and Tompkins algorithm, which resulted in a better performance. However, GQRS still fails at detecting some of the QRS complexes because of its inability to adapt properly in noisy signals.

C. WAVEDET

Wavedet [8] is a wavelet based QRS complex detection algorithm. It performs wavelet decomposition on the ECG signals, which produces a time series of frequencies. After the decomposition, a matched filter detects the QRS complexes by looking at the patterns of the wavelet coefficients. The matched filter allows for the analysis of many different signals at varying frequencies and time intervals, thus enabling the separation of the QRS complex signals from the non QRS complex signals. For the final QRS complex detection, it uses an adaptive amplitude filter. Wavedet performs better than GQRS under low noise conditions due to its multi-resolution analysis, but performs poorly under high noise conditions due to its ineffective matched filter and adaptive amplitude filter. The matched filter is unable to filter out the noises as it can not distinguish the false QRS complexes from the actual QRS complexes. Furthermore, the amplitude filter can not tell the difference between the noises and the actual QRS complexes just by looking at the amplitudes.

D. AUTOMATIC QRS COMPLEX DETECTION USING TWO-LEVEL CONVOLUTIONAL NEURAL NETWORK

Xiang et al.'s paper [25] detects QRS complexes using a 2-layer CNN. The first ECG channel is obtained by applying a difference filter to the original input ECG signal. The second ECG channel is produced by applying a moving average filter and a difference filter to the original input ECG signal. After filtering, two 1×5 pixel CNN kernels are applied to the ECG channels. For the second CNN layer, it uses a 1×5 pixel CNN kernel. Finally, the MLP layers make the final QRS complex predictions. Xiang et al.'s CNN is fast and produces great results under low noise conditions. However, Xiang et al.'s CNN is ineffective under high noise conditions due to its difference filter. The difference filter is a highpass filter that allows high frequency noise through, which introduces classification errors and decreases the performance of the algorithm.

E. ROBUST HEARTBEAT DETECTION FROM MULTIMODALDATA VIA CNN-BASED GENERALIZABLE INFORMATION FUSION

Chandra et al.'s paper [28] uniquely features an inter-patient testing scheme. In the testing scheme, the patients in the training set differ from the patients in the testing set. This testing scheme proves the generalization ability of their algorithm. Their neural network has a 1-layer CNN and an MLP. The CNN has 2 filters with a kernel size of 29 pixels. The MLP has one 200-neuron hidden layer and employs a sigmoid activation function. The model performs slightly better than Xiang et al.'s CNN due to the former's large CNN kernel size and the former's greater number of neurons. However, it was not designed for high noise conditions, and hence its performance degrades in very noisy data that often happen in wearable ECG devices.

III. DATA PREPARATION

As stated in the introduction, data preparation provides the testing and training environment to compare the various QRS complex detection algorithms. The MIT-BIH arrhythmia database [35], [36] and the European ST-T database [37] were selected for the training and testing of the QRS complex detection algorithms. The MIT-BIH database was sampled at 360 Hz, or equivalently 1 sample per 2.78 ms. In order to maintain a consistent sample rate, the European ST-T database was upsampled from 250 Hz to 360 Hz. The databases have relatively clean ECG recordings. To simulate the noisy wearable ECG devices, noise was added to the ECG recordings using the PhysioToolkit Noise Stress Test [38] software. In this paper, only the first 640,000 samples of each ECG recording were used due to the constraints of the PhysioToolkit Noise Stress Test [38]. The worst case signal to noise ratio (SNR) for most wearable ECG devices ranges from 12 dB SNR to 0 dB SNR. As a result, only the 12 dB SNR and the 0 dB SNR ECG recordings were used.

The following labels were selected for QRS complex detection: N, •, L, R, A, a, J, S, V, F, e, j, E, /, f, and Q. After the selection, the labels were converted into floats. For every individual sample that has a QRS complex label, $y = 1.0$ was assigned to that individual sample, which usually corresponds to the R peak position or very close to the R peak. The floats $y = 0.0$ were assigned to all other samples in the recording. There is only one $y = 1.0$ label for each QRS complex. All detection algorithms were restricted to using only the primary ECG lead for QRS complex detection, while Other ECG leads were not used. The usage of only the primary ECG lead was also done to mimic wearable single channel ECG devices.

Some of the ECG recordings in the databases have inconsistent label positioning. A portion of the QRS complexes were labeled at the R peak, while other QRS complexes were labeled at the start of the Q wave. For this paper, the QRS complexes labeled at the R peak were used. Moreover, a few ECG recordings have QS complexes instead of QRS complexes. The detection of QS complexes is out of the scope of this paper. The following correct ECG recordings from the MIT-BIH database were used for training and testing: 100, 101, 102, 103, 104, 105, 106, 109, 112, 113, 115, 116, 118, 119, 121, 122, 123, 201, 202, 208, 209, 212, 213, 214, 215, 217, 219, 220, 221, 222, 228, 230, 231, 232, and 234. Furthermore, the following correct ECG recordings from the European ST-T database were used for training and testing: e0103, e0104, e0111, e0112, e0113, e0115, e0116, e0118, e0123, e0127, e0136, e0147, e0151, e0154, e0159, e0161, e0166, e0170, e0203, e0204, e0206, e0207, e0208, e0210, e0212, e0303, e0306, e0404, e0406, e0408, e0409, e0410, e0411, e0417, e0418, e0509, e0601, e0606, e0607, e0609, e0610, e0611, e0612, e0613, e0615, e0704, e0818, and e1304. Patients with multiple ECG recordings in the database had only one ECG recording included in this study. The datasets were concatenated into one dataset and randomly shuffled during the 1×10 fold testing phase. After shuffling, 14 ECG recordings were used as the training dataset and the remaining ECG recordings were grouped as the testing dataset. This way the patients from the training dataset differ from the patients in the testing dataset, realizing interpatient testing to minimize bias towards the training dataset. The following recordings were used for the MIT-BIH NST cross validation set: 107, 117, 124, and 205. These recordings were selected because they already have significant noise artifacts or ECG deformations present.

IV. PROPOSED CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

In this paper, we propose a CNN-LSTM for the detection of QRS complexes in noisy ECG signals. The algorithm takes in a 2 channel ECG signal. Note that channel 1 is the filtered version of the primary ECG lead, and channel 2 is the gradient of channel 1. To mimic wearable ECG devices, the model does not use any other ECG lead besides the primary ECG lead. The model predicts QRS complexes by producing a delta function at the location of the R peak.

In the pre-processing phase, a Butterworth highpass filter $n = 3, f_c = 5$ Hz is applied to the primary ECG lead in order to obtain channel 1. The Butterworth filter reduces the baseline wandering of the ECG signals by attenuating the signals below $f_c = 5$ Hz. After obtaining channel 1, a difference filter is applied to the channel 1 in order to obtain channel 2, as given by

$$y[t] = x[t] - x[t-1] \tag{1}$$

where $x[t]$ is the input ECG signal with respect to time t and $y[t]$ is the filtered output signal with respect to time t. The difference filter enhances signals that have large gradients. As the QRS complexes have large gradients, the difference filter enhances the QRS complexes. After the filtering, channel 1 and channel 2 are independently normalized in order to compensate for the differing patients and ECG devices. First, each ECG recording is divided into ECG segments of 1,280 samples each. Second, each segment is normalized using the mean of the local maximums.

The architecture of the CNN-LSTM is shown in Fig. 1. It is made from a 2-layer 2D CNN, a 2-layer LSTM, and a 3-layer MLP. The purpose of the CNN layers is to extract the visual features from the ECG signals. Moreover, the CNN layers are able to filter noise from the ECG signals. The visual features extracted by the CNN layers are sent to the LSTM layers, which predict the future QRS complexes using the previous QRS complexes. Furthermore, the LSTM layers smooth out high frequency noise present in the ECG signals. The timing predictions from the LSTM layers are sent to the MLP layers, which apply thresholding to the timing predictions in order to produce the final QRS complex predictions.

The CNN-LSTM architecture is superior to the CNN counterpart because the former takes into account of the temporal correlations between the ECG samples through the LSTM. QRS complexes are quasi-periodic signals. If the period of the QRS complexes is known and position of the latest QRS complex is known, the position of the next QRS complex could be predicted. The LSTM enables the prediction of the next QRS complex position by using the previous QRS complex position and the visual features from the CNN.

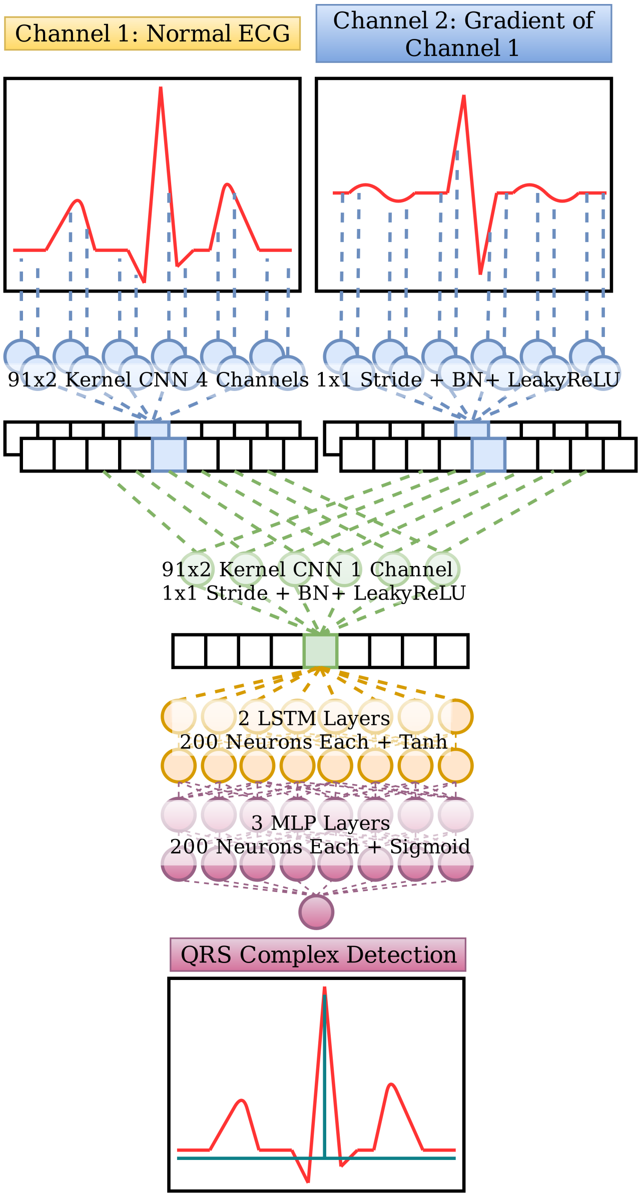

Channel 1: Normal ECG
Channel 2: Gradient of Channel 1
91x2 Kernel CNN 4 Channels 1x1 Stride + BN+ LeakyReLU
91x2 Kernel CNN 1 Channel
1x1 Stride + BN+ LeakyReLU
2 LSTM Layers
200 Neurons Each + Tanh
3 MLP Layers
200 Neurons Each + Sigmoid
QRS Complex Detection

FIGURE 1. The proposed CNN-LSTM architecture.

A. HYPERPARAMETER TUNING

Table 1 shows the hyperparameter tuning of the CNN-LSTM. Firstly, the CNN kernel size is varied until the optimal 91×2 kernel size is found. Secondly, the number of CNN channels, i.e., filters, in the first layer is varied. The optimal number of CNN channels is found to be 4 CNN channels. Thirdly, the number of LSTM and MLP neurons per layer is altered and the optimal number is found to be 200. In order to preserve the information between the LSTM layers and the MLP layers, the number of LSTM neurons per layer must equal the number of MLP neurons per layer. Finally, the optimal number of LSTM layers is found to be 2 LSTM layers, and the optimal number of MLP layers is 3.

TABLE 1. CNN-LSTM hyperparameter tuning.

F_1 score	CNN kernel size	CNN channels	LSTM neurons per layer	LSTM layers	MLP neurons per layer	MLP layers
0.955521	21x2	4	200	2	200	3
0.963075	41x2	4	200	2	200	3
0.974967	61x2	4	200	2	200	3
0.977523	91x2	4	200	2	200	3
0.959403	91x2	1	200	2	200	3
0.977511	91x2	2	200	2	200	3
0.977523	91x2	4	200	2	200	3
0.960813	91x2	6	200	2	200	3
0.973849	91x2	4	50	2	50	3
0.968482	91x2	4	100	2	100	3
0.977523	91x2	4	200	2	200	3
0.970380	91x2	4	300	2	300	3
0.957475	91x2	4	200	1	200	3
0.977523	91x2	4	200	2	200	3
0.968161	91x2	4	200	3	200	3
0.967221	91x2	4	200	2	200	2
0.977523	91x2	4	200	2	200	3
0.964121	91x2	4	200	2	200	4

Note: MIT-BIH NST 12 dB SNR database. Cross validation set.

B. CNN DESCRIPTION

The first CNN layer has a kernel size of 91×2. As the kernel needs to detect QRS complex gradients, the kernel size is set to the size of a QRS complex gradient. The CNN layers' horizontal strides control how much the kernels shift at every time interval. In order to preserve the timing of the ECG signal, the horizontal strides of the CNN layers are set to 1 sample. This makes the kernels shift right by 1 sample at every time interval. When the kernels go out of the bounds of the input matrix, the ends of the input matrix are padded with zeros. The first CNN layer uses 4 channels in order to detect the 4 main QRS complex like waveforms: QRS complex, qRs complex [39], QR complex, and RS complex. The first CNN layer uses the LeakyReLU activation function with $\alpha = 0.02$ given by

$$LeakyReLU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{2}$$

where x is the input matrix to the LeakyReLU function. The LeakyReLU function is fast due to its low computational complexity. Moreover, it prevents the loss from reaching zero. The first CNN layer also uses the batch normalization function given by

$$BN(x) = \frac{x - \mu_x}{\sigma_x} \tag{3}$$

where x is the input matrix, μ_x is the mean of x, and σ_x is the standard deviation of x. Batch normalization helps the neural network to converge faster. The second CNN layer is similar to the first CNN layer, with the only difference being the number of channels. The second CNN layer takes in 4 CNN channels from the first CNN layer and reduces it to 1 channel, which effectively functions as a 4 to 1 pooling layer.

C. LSTM DESCRIPTION

The second CNN layer connects to the first LSTM layer, which predicts the QRS complex timings using the 1D sequence of visual features from the CNN layers. The QRS complex timings allow the LSTM layers to narrow the search spaces for QRS complexes. There are 2 LSTM layers. Each has 200 neurons and uses the tanh function as the activation function. The tanh function has a range of $r \in [-1, 1]$, which allows for the negative and positive feedback in the LSTM layers without exponential feedback, which in turn allows the LSTM layers to remember different past information. The LSTM with the tanh activation function can be viewed as a smoothing filter and hence is able to smooth out high frequency noise present in the ECG signals.

D. MLP DESCRIPTION

The final LSTM layer fully connects to the first MLP layer. The purpose of the MLP layers is to execute the final QRS complex detection. The MLP layers apply thresholding to the QRS complex timing predictions in order to filter out the incorrect QRS complex predictions. There are 3 MLP layers, each having 200 neurons. The MLP layers use the batch normalization function and the sigmoid activation function given by

$$S(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

where x is the input matrix. The sigmoid activation function constrains the MLP layers' output to the continuous interval of $Q \in [0, 1]$. In order to produce a binary output, a final threshold $f_{thres} = 0.9$ is applied to MLP layers' output Q. If $Q > f_{thres}$, then the CNN-LSTM predicts $\hat{y} = 1.0$ to signal the presence of QRS complex, otherwise the CNN-LSTM predicts $\hat{y} = 0.0$ to signal the absence of a QRS complex.

E. LOSS FUNCTION

Neural networks are trained by minimizing a defined loss function. As a result, the choice of the loss function is critical to the performance of the neural network. This work uses the weighted cross-entropy loss function expressed as

$$J(\hat{y}, y) = -\log(S(\hat{y}))(y)(W_{pos}) - \log(1 - S(\hat{y}))(1 - y) \quad (5)$$

where y is the QRS complex label and W_{pos} is the cross-entropy weight. The weighted cross-entropy loss function is chosen because the function allows the designer to change the ratio of false positives (FP) to false negatives (FN) by varying the cross-entropy weight W_{pos}. Each ECG recording has approximately 340 samples in between each pair of QRS complexes. Therefore, the number of true negatives (TN) is far larger than the number of true positives (TP). The imbalance is corrected by setting the cross-entropy weight to $W_{pos} = 340$. Furthermore, the predicted QRS complex detection $\hat{y}$ is matched against the actual QRS complex detection label y. If they both have the same value $\hat{y} \approx y$, then the loss function is small. If they have different values $\hat{y} \neq y$, then the loss function is large. This fulfills the design objective.

V. SIMULATIONS

In this paper, all algorithms described in Section II are implemented as the comparison basis for the proposed CNN-LSTM. The neural networks are implemented in Python 3 using TensorFlow 1.5 [40], while the other algorithms are implemented in MATLAB using the PhysioNet ECG-Kit [36]. The QRS complex detection algorithms are benchmarked using the noisy dataset described in Section III.

A. EVALUATION METRICS

The true positives (TP), false positives (FP), false negatives (FN), sensitivity (SEN), positive predictive value (PPV), F1 score (F_1), and root mean-squared error (RMSE) of the timings of the QRS complex detection algorithms are recorded. Here, SEN, PPV and F_1 are computed according to the equations below

$$SEN = \frac{TP}{TP + FN} \quad (6)$$

$$PPV = \frac{TP}{TP + FP} \quad (7)$$

$$F_1 = 2\frac{SEN \cdot PPV}{SEN + PPV}. \quad (8)$$

Sensitivity measures the number of false negatives in relation to the actual QRS complexes. Positive predictive value measures the number of false positives among the detected QRS complexes. If a QRS complex detection algorithm performs well, then it must have a high sensitivity $SENS \approx 1$ and a high positive predictive value $PPV \approx 1$. This in turn causes the $F_1 \approx 1$ to be high.

If a QRS complex detection algorithm predicts the R peak of a QRS complex within 50 ms of the R peak of a true QRS complex, then the predicted QRS complex counts as a true positive. If a QRS complex detection algorithm predicts a QRS complex and the R peak of a true QRS complex does not exist within 50 ms of the R peak of the predicted QRS complex, then it is counted as a false positive. If a QRS complex detection algorithm does not predict the R peak of a QRS complex within 50 ms of the R peak of a true QRS complex, then it is counted as a false negative. The true negatives are not relevant as none of the ECG metrics use them.

Another important performance measure is related to the timing accuracy of the R wave, in addition to QRS detection benchmarks. R peak timing error directly impacts the accuracy of RR intervals, PR intervals, and heart rate variability calculations. Here, the RMSE timing metric, given by

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(T_i - \hat{T}_i)^2} \quad (9)$$

is used for the evaluation of the QRS complex detection algorithms, where M is number of QRS complexes, T_i is the QRS complex label time, and $\hat{T}_i$ is the QRS complex prediction time.

B. CNN-LSTM LEARNING CURVE

Fig. 2 shows the learning curve of the CNN-LSTM. The learning curve was generated using the MIT-BIH NST 12 dB SNR database. Some of the ECG segments in the MIT-BIH NST database have low noise, while others have high noise. This discrepancy causes fluctuations in the F_1 score. Furthermore, the CNN-LSTM may perform better in certain ECG recordings, which also leads to more fluctuations in the F_1 score. The fluctuations in the F_1 score account for the large error bars in the learning curve. These errors also plague the other algorithms presented in Table 2, which results in large F_1 score standard deviations. The learning curve narrowing around the 12 to 14 ECG recording mark. This proves the convergence of the model. Moreover, the learning curve also proves the CNN-LSTM is neither underfitting nor overfitting.

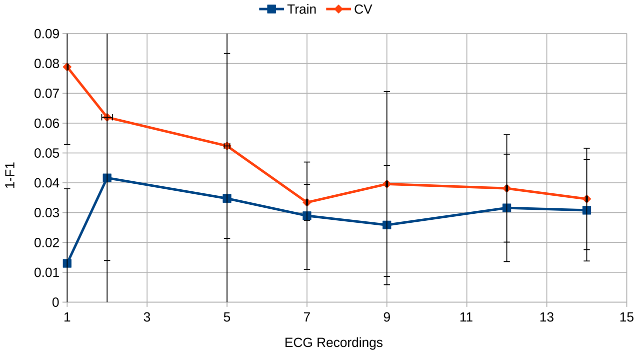


FIGURE 2. CNN-LSTM's learning curve. MIT-BIH NST 12 dB SNR. 2σ error bar.

TABLE 2. MIT-BIH NST algorithm performance, with 12 dB SNR.

Algorithm	TP	FP	FN	SENS	PPV	F_1 score	Timing RMSE
GQRS [7]	46702 ± 1904	8652 ± 1404	1650 ± 626	0.9658 ± 0.013	0.8436 ± 0.025	0.9005 ± 0.019	12.40 ± 0.08
Pantom [3]	40140 ± 4564	2265 ± 2582	8164 ± 3184	0.8305 ± 0.069	0.9462 ± 0.062	0.8844 ± 0.062	6.98 ± 0.80
Wavedet [8]	43809 ± 3556	12590 ± 2226	3945 ± 1830	0.9172 ± 0.039	0.7765 ± 0.044	0.8409 ± 0.041	4.01 ± 0.66
Xiang et al. [25]	45411 ± 6626	2739 ± 2326	2766 ± 5020	0.9419 ± 0.107	0.9444 ± 0.042	0.9418 ± 0.043	1.98 ± 0.80
Chandra et al. [28]	46993 ± 2877	4738 ± 3186	625 ± 751	0.9868 ± 0.016	0.9089 ± 0.058	0.9460 ± 0.032	1.58 ± 0.44
Proposed	46591 ± 3120	2218 ± 1742	1155 ± 1693	0.9757 ± 0.035	0.9550 ± 0.033	0.9650 ± 0.017	1.76 ± 0.50

Note: Confidence interval of 2σ. Timing RMSE units in samples.

TABLE 3. MIT-BIH NST algorithm performance, with 0 dB SNR.

Algorithm	TP	FP	FN	SENS	PPV	F_1 score	Timing RMSE
GQRS [7]	39941 ± 2008	23746 ± 1206	8307 ± 684	0.8277 ± 0.015	0.6270 ± 0.023	0.7135 ± 0.019	12.16 ± 0.09
Pantom [3]	9758 ± 1573	6533 ± 217	37906 ± 2928	0.2048 ± 0.035	0.5981 ± 0.032	0.3049 ± 0.042	5.57 ± 0.20
Wavedet [8]	39106 ± 1911	20716 ± 699	8576 ± 593	0.8201 ± 0.008	0.6536 ± 0.018	0.7274 ± 0.012	4.69 ± 0.42
Xiang et al. [25]	37006 ± 24831	14552 ± 10141	10947 ± 25316	0.7732 ± 0.518	0.7212 ± 0.045	0.7036 ± 0.469	2.64 ± 0.81
Chandra et al. [28]	42216 ± 1268	16952 ± 2431	5641 ± 1158	0.8822 ± 0.020	0.7137 ± 0.031	0.7890 ± 0.023	2.57 ± 0.51
Proposed	43384 ± 3019	14355 ± 2280	4189 ± 1402	0.9117 ± 0.031	0.7513 ± 0.038	0.8237 ± 0.033	2.57 ± 0.56

Note: Confidence interval of 2σ. Timing RMSE units in samples.

TABLE 4. European ST-T NST algorithm performance, with 12 dB SNR.

Algorithm	TP	FP	FN	SENS	PPV	F_1 score	Timing RMSE
GQRS [7]	65835 ± 2726	9511 ± 897	1206 ± 599	0.9819 ± 0.008	0.8737 ± 0.011	0.9247 ± 0.009	12.66 ± 0.30
Pantom [3]	55556 ± 4387	2241 ± 713	11517 ± 2522	0.8281 ± 0.037	0.9612 ± 0.011	0.8896 ± 0.024	10.11 ± 0.55
Wavedet [8]	66374 ± 2135	18451 ± 1752	1616 ± 352	0.9761 ± 0.005	0.7824 ± 0.020	0.8686 ± 0.014	11.81 ± 0.34
Xiang et al. [25]	58909 ± 8037	2283 ± 1666	6752 ± 8403	0.8974 ± 0.126	0.9633 ± 0.022	0.9277 ± 0.063	0.75 ± 0.37
Chandra et al. [28]	66101 ± 2813	3293 ± 1338	297 ± 145	0.9955 ± 0.002	0.9525 ± 0.019	0.9735 ± 0.010	0.83 ± 0.22
Proposed	65802 ± 2847	1789 ± 1087	849 ± 822	0.9873 ± 0.012	0.9735 ± 0.015	0.9803 ± 0.006	1.07 ± 0.27

Note: Confidence interval of 2σ. Timing RMSE units in samples.

TABLE 5. European ST-T NST algorithm performance, with 0 dB SNR.

Algorithm	TP	FP	FN	SENS	PPV	F_1 score	Timing RMSE
GQRS [7]	58611 ± 1680	32359 ± 980	8354 ± 854	0.8752 ± 0.012	0.6442 ± 0.012	0.7421 ± 0.011	12.38 ± 0.16
Pantom [3]	15671 ± 2187	8101 ± 762	53038 ± 2386	0.2280 ± 0.030	0.6586 ± 0.035	0.3386 ± 0.038	12.03 ± 0.39
Wavedet [8]	57813 ± 1419	31697 ± 711	9913 ± 624	0.8536 ± 0.007	0.6458 ± 0.010	0.7353 ± 0.007	12.35 ± 0.38
Xiang et al. [25]	57121 ± 6175	17375 ± 7378	8880 ± 4375	0.8650 ± 0.069	0.7683 ± 0.076	0.8130 ± 0.054	1.75 ± 0.45
Chandra et al. [28]	62234 ± 2475	19569 ± 3184	3882 ± 829	0.9412 ± 0.012	0.7609 ± 0.032	0.8414 ± 0.019	1.73 ± 0.23
Proposed	60620 ± 3002	13415 ± 5102	4909 ± 2706	0.9251 ± 0.040	0.8197 ± 0.052	0.8688 ± 0.033	1.65 ± 0.21

Note: Confidence interval of 2σ. Timing RMSE units in samples.

C. RESULTS

Tables 2-5 show the results of the 1×10 fold testing on the MIT-BIH NST and the European ST-T NST databases. For both databases, the proposed CNN-LSTM outperforms GQRS [7], Pan and Tompkins [3], Wavedet [8], Xiang et al.'s CNN [25], and Chandra et al.'s CNN [28] in terms of F_1 score. For example, for the 12 dB SNR MIT-BIH NST database, the proposed CNN-LSTM's F_1 score of 0.9650 is greater than GQRS's F_1 score of 0.9005, Pan and Tompkins's F_1 score of 0.8844, Wavedet's F_1 score of 0.8409, Xiang et al.'s CNN's F_1 score of 0.9418, and Chandra et al.'s CNN's F_1 score of 0.9460. Also shown in these tables, the most recent machine learning based algorithms, [25], [28] and the proposed CNN-LSTM, have clear advantages over the previous filter and wavelet based algorithms, which demonstrates the effectiveness of neural networks. The proposed model performs consistently better than the other NN based QRS complex detection algorithms for noisy data because our CNN-LSTM model has larger CNN kernels than the latter. The larger CNN kernels help the CNN-LSTM to filter out the noise better, thus reducing the number of false positives. Furthermore, the LSTM layers improve the F_1 score of the CNN-LSTM model by predicting the future QRS complexes correctly. Finally, the proposed model has a greater number of neurons than the other NN. The greater number of neurons allows the CNN-LSTM to detect more complex patterns, which improves the F_1 score.

D. WIDE QRS COMPLEXES

Fig. 4 and Fig. 5 show patients with wide QRS complexes. The ECG signals have QRS complex widths of 80 samples (222 ms) and 90 samples (250 ms) respectively. The smaller

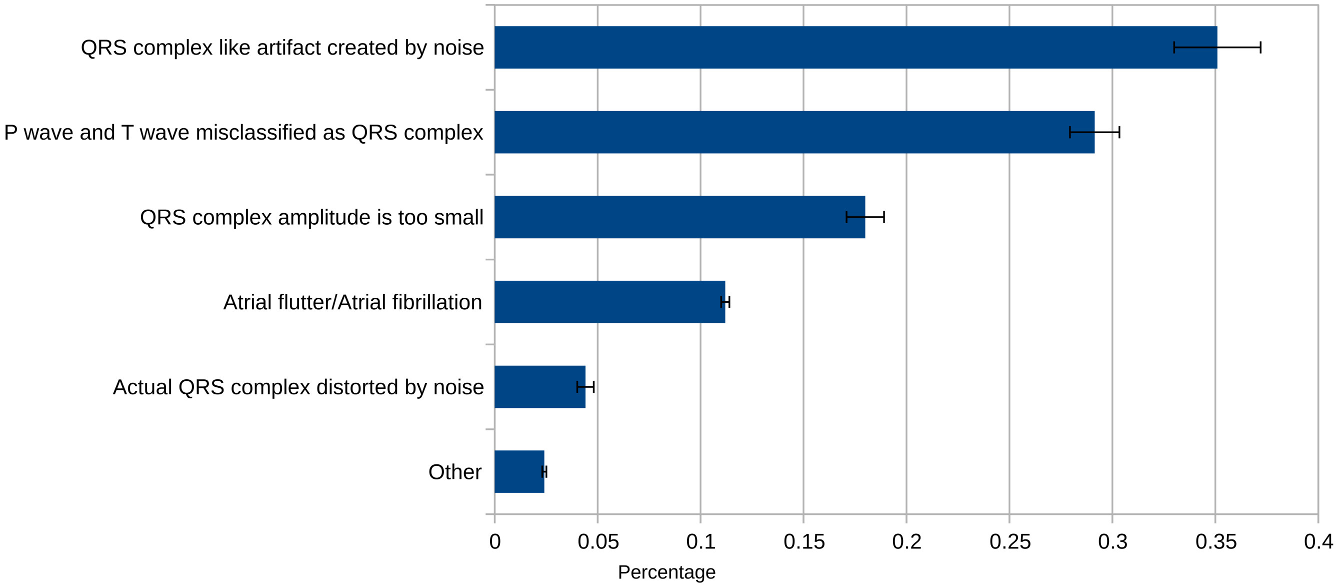


FIGURE 3. **CNN-LSTM's error distribution. MIT-BIH NST 12 dB SNR. 2σ error bar.**

CNN kernel sizes have trouble detecting large QRS complexes because they can not capture the entire QRS complex. Thus, the large 91×2 CNN kernels were used to detect the large QRS complexes. This results in an increase of F_1 as shown in Table 1.

E. CNN-LSTM LIMITATIONS

The proposed CNN-LSTM has a few limitations. The timing RMSE of the model is similar to that of Xiang et al.'s CNN [25] and Chandra et al.'s CNN [28] at low SNRs, but slightly worse at high SNRs. The timing errors of the proposed model are due to the large 91×2 CNN kernels. All CNN kernels have a trade off between spatial frequency uncertainty and position uncertainty. The 91×2 CNN kernels has low spatial frequency uncertainty at the cost of high position uncertainty. Another limitation of the proposed model is the computational complexity. At every time interval n, the CNN performs one convolution with kernel width W and kernel height H at a cost of $O(W, H) = WH$ computations. If the number of channels C is considered, then the cost is $O(W, H, C) = WHC$ computations. The cost for the entire time interval n is $O(n) = WHCn$ computations. With the addition of many CNN layers L_{CNN}, the cost becomes $O(n) = L_{CNN}WHCn$ computations.

Now, consider the computational complexity of the LSTM. For a single gate $G = 1$ at a single time interval $n = 1$, the gate has a cost of $O(m, p) = mp$ computations, where m and p are the height and width of the gate's weight matrix respectively. For multiple gates G and time intervals n, the cost is $O(n) = Gmpn$ computations. With the addition of many LSTM layers L_{LSTM}, the cost becomes $O(n) = L_{LSTM}Gmpn$ computations. The MLP layers have the same weight dimensions as the LSTM layers. Thus, the computational cost of the MLP layers is $O(n) = L_{MLP}mpn$ computations, where L_{MLP} is the number of MLP layers. Finally, the total computational complexity of the CNN-LSTM is

$$O(n) = L_{CNN}WHCn + L_{LSTM}Gmpn + L_{MLP}mpn. \quad (10)$$

The computational complexity of the CNN-LSTM is higher than the computational complexities of other QRS complex detection algorithms. As a result, the proposed model detects QRS complexes at a slower rate than the rest of the QRS complex detection algorithms. The CNN-LSTM also requires more ECG recordings for the training phase. The proposed model requires at least 11 ECG recordings for the training phase as shown in Fig. 2. The rest of the QRS complex detection algorithms only require 100,000 ECG samples for the training phase. These limitations can be largely overcome by today's powerful computing machines such as GPU during training.

VI. ERROR ANALYSIS

A detailed error analysis of the CNN-LSTM indicates the following 5 main errors: QRS complex like artifact created by noise, P wave and T wave misclassified as QRS complex, QRS complex amplitude too small, atrial flutter/atrial fibrillation, and actual QRS complex distorted by noise. Fig. 3 shows the CNN-LSTM's error distribution.

A. QRS COMPLEX LIKE ARTIFACT CREATED BY NOISE

This error type occurs when QRS complex like artifacts are introduced by the noises, generated using the PhysioToolkit Noise Stress Test [38], and is the main source of error accounting for 35.12% of total number of errors. The criteria

$$(FP) \wedge (RMSE(ECGclean[t], ECGnoisy[t]) > 0) \quad (11)$$

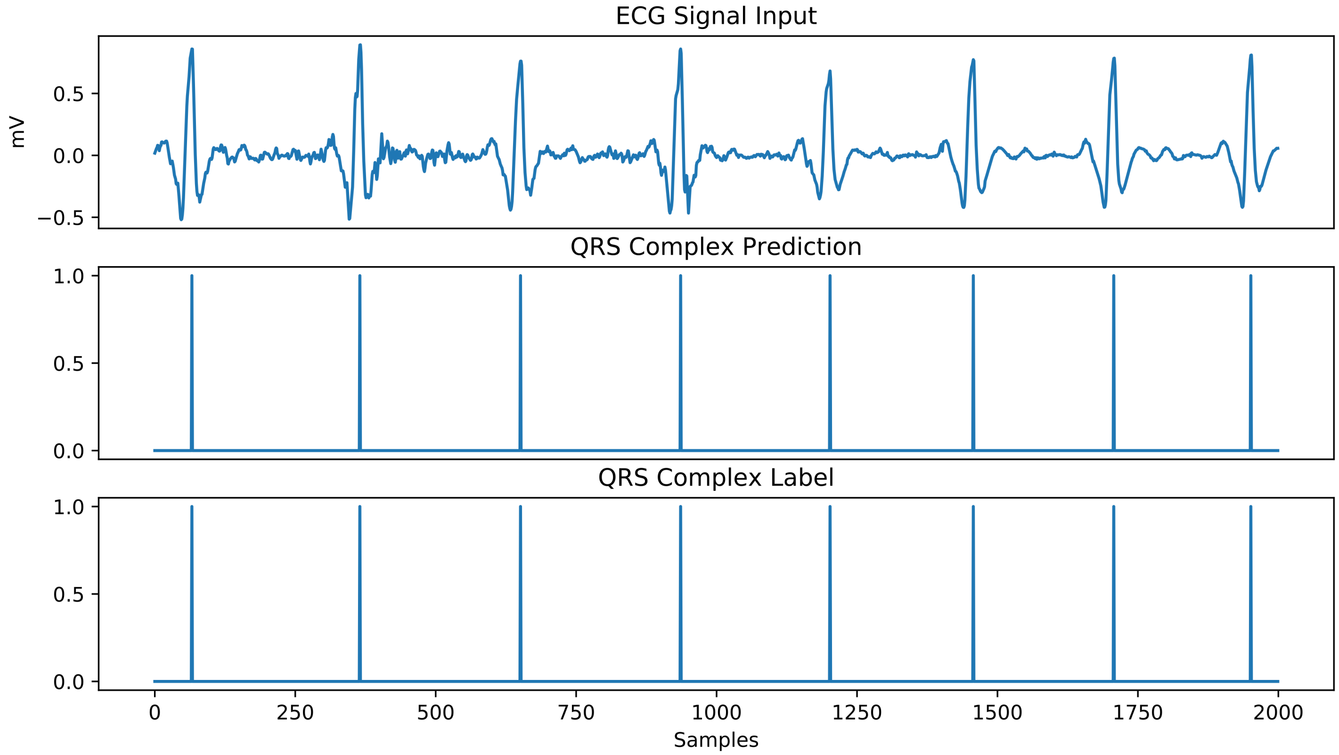


FIGURE 4. MIT-BIH NST CNN-LSTM QRS complex detection. QRS complex width 80 samples (222 ms).

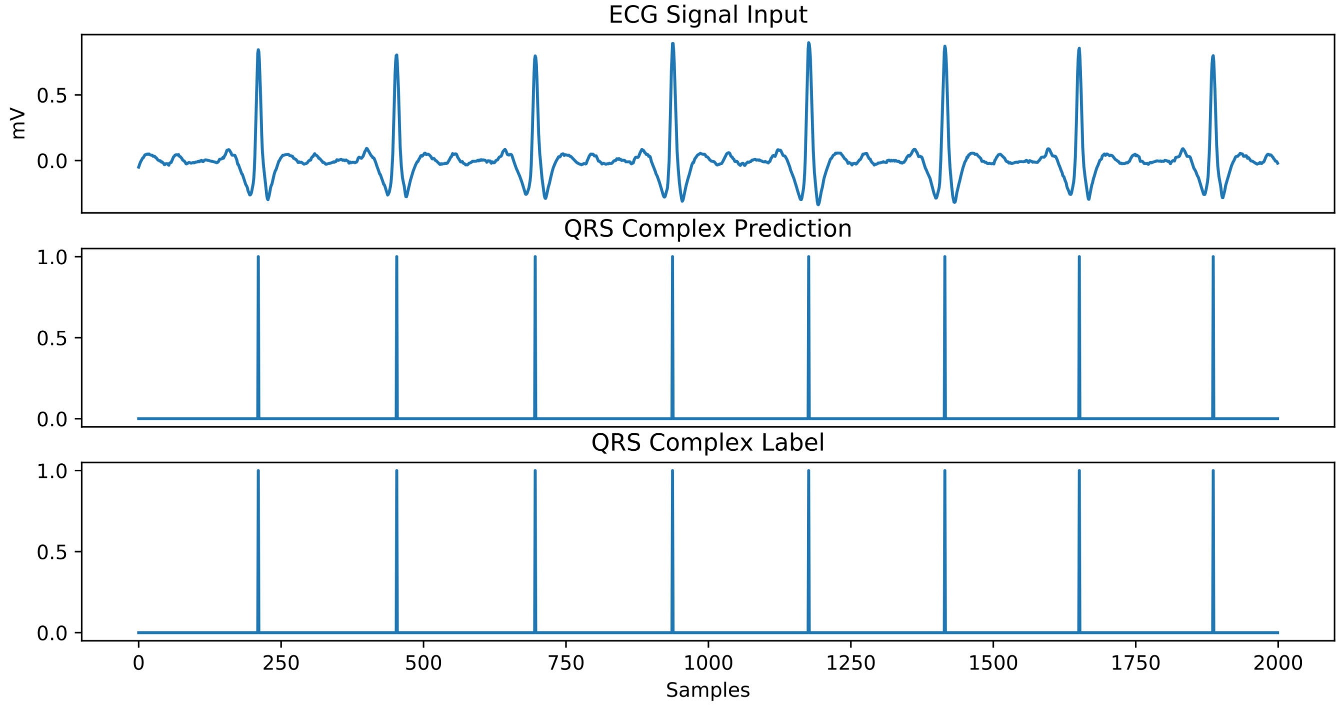


FIGURE 5. MIT-BIH NST CNN-LSTM QRS complex detection. QRS complex width 90 samples (250 ms).

is used to classify the error, where the error is a false positive $FP = True$ and large amounts of noises are introduced $RMSE(ECGclean[t], ECGnoisy[t]) > 0$. $ECGclean[t]$ and $ECGnoisy[t]$ represent the ECG signals before and after the additions of the noises respectively. The generated artifacts are almost indistinguishable from the actual QRS complexes.

The artifacts could be minimized by employing more filters or advanced neural networks. For example, the filters could minimize the number of false positives by rejecting false QRS complexes before they reach the CNN-LSTM.

B. P WAVE AND T WAVE MISCLASSIFIED AS QRS COMPLEX

P waves and T waves in the ECG signals sometimes look similar to the QRS complexes, especially when they become larger than QRS complex in amplitude. This error type happens when a P wave or a T wave is misclassified as a QRS complex. The criteria

$$(FP) \land ((label == P) \lor (label == T)) \tag{12}$$

is used to classify the error, where the error is a false positive $FP = True$ and a P wave or a T wave is within 50 ms of the error. The P waves and T waves could be removed using a P wave and T wave detector. However, the detector may introduce more errors.

C. QRS COMPLEX AMPLITUDE TOO SMALL

The CNN-LSTM uses thresholding to detect QRS complexes. If a QRS complex amplitude is above the threshold, then it gets detected; Otherwise it does not get detected. This error type happens when a QRS complex amplitude is too small, which results in a false negative error. The criteria

$$(FN) \land (E[A_{QRS}] > A_{QRS}) \tag{13}$$

is used to classify the error, where the error is a false negative $FN = True$ and the expected value of the QRS complex amplitudes $E[A_{QRS}]$ is greater than the current QRS complex amplitude A_{QRS}. This could be reduced by using better normalization algorithms. However, the normalization algorithms introduce a chicken and egg problem. The QRS complex detection algorithm requires a normalization algorithm in order to increase the QRS detection accuracy. On the other hand, the normalization algorithm needs the actual QRS complex amplitude because the noise peaks could be higher than the actual QRS complexes.

D. ATRIAL FLUTTER/ATRIAL FIBRILLATION

When atrial flutters or atrial fibrillations occur, the ECG signals look like triangular waves or saw-tooth waves. This significantly distorts the QRS complexes and introduces detection errors. The criteria

$$(FN) \land ((label == AFIB) \lor (label == AFL)) \tag{14}$$

is used to classify the error, where the error is a false negative $FN = True$ and the ECG segment is labeled as an atrial flutter or an atrial fibrillation. The misclassification errors may be resolved by increasing the cross entropy weights of the segments that have atrial flutters or atrial fibrillations. Moreover, the false negative errors could be reduced by using a specialized CNN-LSTM just for the detection of the atrial flutters and the atrial fibrillations.

E. ACTUAL QRS COMPLEX DISTORTED BY NOISE

This error type occurs when the actual QRS complex is distorted by the noises generated by the PhysioToolkit Noise Stress Test [38]. The distorted QRS complex does not resemble any normal QRS complex, thus resulting in a classification error. The criteria

$$(FN) \land (RMSE(ECGclean[t], ECGnoisy[t]) > 0) \tag{15}$$

is used to classify the error, where the error is a false negative $FN = True$ and the actual QRS complex is distorted by the noises $RMSE(ECGclean[t], ECGnoisy[t]) > 0$. The error could be minimized by adding more filters to the model. More filters could mean better detection of distorted QRS complexes.

VII. CONCLUSION

This paper has presented a novel CNN-LSTM structure for the detection of QRS complexes in noisy ECG signals. Moreover, an inter-patient training/testing procedure has been devised to prove the generalization ability of the CNN-LSTM. The generalization ability of the CNN-LSTM is particularly useful for automatic analysis of ECG data collected by mobile wearable devices, where manual labeling of individual patients' records is unrealistic. Inside the stacked network, the CNN layers extract visual features and filter out noise from the noisy ECG signals. The LSTM layers predict the QRS complex timings. The subsequent MLP layers execute the final QRS complex detections and format the outputs of the network. Simulations using MIT-BIH NST and European ST-T NST databases have demonstrated that the proposed CNN-LSTM outperforms the existing algorithms in the literature in terms of F_1 score. As a result, the proposed CNN-LSTM is a promising solution for use in noisy wearable ECG devices.

APPENDIXES

REVIEW ON NEURAL NETWORKS

Neural networks are the building blocks for some of the QRS complex detection algorithms. Therefore, a brief review of neural networks is presented below.

A. MULTI-LAYER PERCEPTRON

MLPs [41] are a type of neural network. They excel at classifying many types of data. Moreover, MLPs are proficient at adapting to changing input data and are suitable for detecting QRS complexes in ECG signals. The fundamental equation for the MLPs is given by

$$O_i = A(W_i^T X_i + B_i). \tag{16}$$

For each layer i, X_i, W_i, B_i are the input, weight, and bias matrices respectively. $A(V)$ is the activation function with respect to input matrix V. Also, O_i is the output matrix of layer i. The input data enters at the input matrix X_i. The input X_i is multiplied by the weights W_i and the result is added to the bias B_i. The output O_i is obtained by passing the result through the activation function $A(V)$.

B. LONG SHORT-TERM MEMORY

LSTMs [42] are a special type of neural network that store memories inside of the neurons. LSTMs remember and forget data using the hidden gates. Therefore, LSTMs are suitable for time-series pattern recognition, where the LSTMs predict the future using only the past input data. LSTMs are often employed to predict QRS complexes in time-series ECG signals. The LSTM's equations are given by

$$i_t^l = S(W_i^l[\chi_t; h_{t-1}^l] + b_i^l) \tag{17a}$$
$$f_t^l = S(W_f^l[\chi_t; h_{t-1}^l] + b_f^l) \tag{17b}$$
$$s_t^l = f_t^l s_{t-1}^l + i_t^l \tanh(W_s^l[\chi_t; h_{t-1}^l] + b_s^l) \tag{17c}$$
$$o_t^l = S(W_o^l[\chi_t; h_{t-1}^l] + b_o^l) \tag{17d}$$
$$h_t^l = o_t^l \tanh(s_t^l) \tag{17e}$$

Eqn. (17a) shows the input gate for the LSTM neuron. Input gate controls which information enters the LSTM neuron. Input data χ_t and final LSTM output data h_{t-1}^l are fed into the input gate i_t^l. Using the input gate weights W_i^l and biases b_i^l, the output vector of the input gate i_t^l is determined. Eqn. (17b) shows the forget gate. The forget gate f_t^l determines if the hidden state s_t^l is forgotten or retained. The forget gate f_t^l is computed just like the input gate i_t^l. Eqn. (17c) shows the computation of hidden state vector s_t^l. The forget gate f_t^l controls the retention of the previous state variable s_{t-1}^l. The input gate i_t^l controls the weight of the tanh activation function. Eqn. (17d) shows the output gate o_t^l. The output gate o_t^l controls what information is outputted by the LSTM neuron. Eqn. (17e) shows the final output vector h_t^l of the LSTM neuron. Final output vector h_t^l is computed using the hidden state vector s_t^l and the output gate o_t^l.

C. CONVOLUTIONAL NEURAL NETWORKS

CNNs [43] are used for processing images. At every layer, the CNN applies a convolutional filter to the input vector of the layer. Convolutional filters allow the CNNs to detect spatial patterns in the data. Moreover, CNNs are able to filter out noise from the input data. Therefore, CNNs are good at detecting QRS complexes as QRS complexes have unique spatial patterns. The CNN equations given by

$$V_{y,z}^i = \sum_{j=1}^{N} \sum_{k=1}^{M} W_{j,k}^i X_{j+y-1,k+z-1}^i \tag{18}$$
$$O^i = A(V^i + B^i) \tag{19}$$

depict 2D convolution between the input matrix X^i and the kernel weights W^i. The kernel weights W^i slide across the input data X^i to produce V^i. After computing V^i, V^i is added to bias B^i. Then the resulting matrix passes through the activation function A and produces the output matrix O^i.

ACKNOWLEDGMENT

The GPU is sponsored by NVidia.

REFERENCES

[1] B.-U. Köhler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection," *IEEE Eng. Med. Biol. Mag.*, vol. 21, no. 1, pp. 42–57, Jan./Feb. 2002.

[2] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 1, pp. 85–98, Jan. 1990.

[3] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[4] R. Gupta, M. Mitra, K. Mondal, and S. Bhowmick, "A derivative-based approach for QT-segment feature extraction in digitized ECG record," in *Proc. 2nd Int. Conf. Emerg. Appl. Inf. Technol.*, Feb. 2011, pp. 63–66.

[5] N. M. Arzeno, C.-S. Poon, and Z.-D. Deng, "Quantitative analysis of QRS detection algorithms based on the first derivative of the ECG," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2006, pp. 1788–1791.

[6] N. M. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of first-derivative based QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 2, pp. 478–484, Feb. 2008.

[7] PhysioNet. (2018). *GQRS, QRS Detector and Post-Processor*. [Online]. Available: https://www.physionet.org/physiotools/wag/gqrs-1.htm

[8] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, "A wavelet-based ECG delineator: Evaluation on standard databases," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 570–581, Apr. 2004.

[9] S. Kadambe, R. Murray, and G. F. Boudreaux-Bartels, "Wavelet transform-based QRS complex detector," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 7, pp. 838–848, Jul. 1999.

[10] K. Mourad and B. R. Fethi, "Efficient automatic detection of QRS complexes in ECG signal based on reverse biorthogonal wavelet decomposition and nonlinear filtering," *Measurement*, vol. 94, pp. 663–670, Dec. 2016.

[11] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 1, pp. 21–28, Jan. 1995.

[12] F. Bouaziz, D. Boutana, and M. Benidir, "Multiresolution wavelet-based QRS complex detection algorithm suited to several abnormal morphologies," *IET Signal Process.*, vol. 8, no. 7, pp. 774–782, Sep. 2014.

[13] U. D. Ulusar, R. B. Govindan, J. D. Wilson, C. L. Lowery, H. Preissl, and H. Eswaran, "Adaptive rule based fetal QRS complex detection using Hilbert transform," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 4666–4669.

[14] D. S. Benitez, P. A. Gaydecki, A. Zaidi, and A. P. Fitzpatrick, "A new QRS detection algorithm based on the Hilbert transform," in *Proc. Comput. Cardiol.*, Sep. 2000, pp. 379–382.

[15] F. I. de Oliveira and P. U. Cortez, "A QRS detection based on Hilbert transform and wavelet bases," in *Proc. 14th IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process.*, Sep./Oct. 2004, pp. 481–489.

[16] P. S. Hamilton and W. J. Tompkins, "Adaptive matched filtering for QRS detection," in *Proc. Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc.*, Nov. 1988, pp. 147–148.

[17] D. T. Kaplan, "Simultaneous QRS detection and feature extraction using simple matched filter basis functions," in *Proc. Comput. Cardiol.*, Sep. 1990, pp. 503–506.

[18] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Adaptive dictionary reconstruction for compressed sensing of ECG signals," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 645–654, May 2017.

[19] M. Balouchestani, K. Raahemifar, and S. Krishnan, "High—Resolution QRS detection algorithm for wireless ECG systems based on compressed sensing theory," in *Proc. IEEE 56th Intl. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2013, pp. 1326–1329.

[20] G. Goovaerts, S. Padhy, B. Vandenberk, C. Varon, R. Willems, and S. Van Huffel, "A machine-learning approach for detection and quantification of QRS fragmentation," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 5, pp. 1980–1989, Sep. 2019.

[21] S. S. Mehta and N. S. Lingayat, "SVM-based algorithm for recognition of QRS complexes in electrocardiogram," *IRBM*, vol. 29, no. 5, pp. 310–317, 2008.

[22] S. S. Mehta and N. S. Lingayat, "Identification of QRS complexes in 12-lead electrocardiogram," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 820–828, 2009.

[23] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-network-based adaptive matched filtering for QRS detection," *IEEE Trans. Biomed. Eng.*, vol. 39, no. 4, pp. 317–329, Apr. 1992.

[24] B. Abibullaev and H. D. Seo, "A new QRS detection method using wavelets and artificial neural networks," *J. Med. Syst.*, vol. 35, no. 4, pp. 683–691, 2011.

[25] Y. Xiang, Z. Lin, and J. Meng, "Automatic QRS complex detection using two-level convolutional neural network," *BioMed. Eng. OnLine*, vol. 17, no. 1, p. 13, Jan. 2018, doi: 10.1186/s12938-018-0441-4.

[26] J. Camps, B. Rodríguez, and A. Mincholé, "Deep learning based QRS multilead delineator in electrocardiogram signals," in *Proc. Comput. Cardiol. Conf.*, Sep. 2018, pp. 1–4.

[27] K. Arbateni and A. Bennia, "Sigmoidal radial basis function ANN for QRS complex detection," *Neurocomputing*, vol. 145, pp. 438–450, Dec. 2014.

[28] B. S. Chandra, C. S. Sastry, and S. Jana, "Robust heartbeat detection from multimodal data via CNN-based generalizable information fusion," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 3, pp. 710–717, Mar. 2019.

[29] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, and Y. Lian, "A level-crossing based QRS-detection algorithm for wearable ECG sensors," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 1, pp. 183–192, Jan. 2014.

[30] F. Zhang and Y. Lian, "Electrocardiogram QRS detection using multiscale filtering based on mathematical morphology," in *Proc. 29th Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 3196–3199.

[31] P. Phukpattaranont, "QRS detection algorithm based on the quadratic filter," *Expert Syst. Appl.*, vol. 42, no. 11, pp. 4867–4877, 2015.

[32] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proc. Comput. Cardiol.*, Sep. 2017, pp. 1–4.

[33] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, "ECG arrhythmia classification using a 2-D convolutional neural network," 2018, *arXiv:1804.06812*. [Online]. Available: https://arxiv.org/abs/1804.06812

[34] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," 2017, *arXiv:1707.01836*. [Online]. Available: https://arxiv.org/abs/1707.01836

[35] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[36] 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, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.," *Circulation*, vol. 101, no. 23, pp. E215–E220, 2000.

[37] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. Moody, C. Zeelenberg, and C. Marchesi, "The European ST-T database: Standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography," *Eur. Heart J.*, vol. 13, no. 9, pp. 1164–1172, Sep. 1992.

[38] G. B. Moody, W. K. Muldrow, and R. G. Mark, "A noise stress test for arrhythmia detectors," *Comput. Cardiol.*, vol. 11, no. 3, pp. 381–384, 1984.

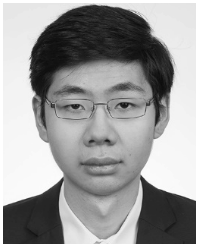
[39] A. L. Goldberger, Z. D. Goldberger, and A. Shvilkin, *Goldberger's Clinical Electrocardiography: A Simplified Approach*. 9th ed. Amsterdam, The Netherlands: Elsevier, 2017.

[40] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.



BROSNAN YUEN is currently pursuing the M.A.Sc. degree in electrical engineering with the University of Victoria, Victoria, Canada. He is currently doing research in robotic systems, ECGs, optics, machine learning, and FPGAs.

XIAODAI DONG (S'97–M'00–SM'09) received the B.Sc. degree in information and control engineering from Xi'an Jiaotong University, China, in 1992, the M.Sc. degree in electrical engineering from the National University of Singapore, in 1995, and the Ph.D. degree in electrical and computer engineering from Queen's University, Kingston, ON, Canada, in 2000.

Since January 2005, she has been with the University of Victoria, Victoria, Canada, where she is currently a Professor with the Department of Electrical and Computer Engineering. From 1999 to 2002, she was with Nortel Networks, Ottawa, ON, Canada, where she worked on the base transceiver design of the third-generation (3G) mobile communication systems. From 2002 to 2004, she was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Her research interests include 5G, mmWave communications, radio propagation, the Internet of things, machine learning, localization, wireless security, e-health, smart grid, and nano-communications. She was the Canada Research Chair (Tier II), from 2005 to 2015. She has served as an Editor for the IEEE Transactions on Communications, from 2001 to 2007, and the IEEE Transactions on Wireless Communications, from 2009 to 2014. She is currently an Editor of the IEEE Transactions on Vehicular Technology.

TAO LU received the B.Sc. degree from the Department of Physics, University of Manitoba, in 1995, the M.Sc. degree from the Department of Electrical and Computer Engineering, Queen's University, Kingston, in 1998, and the Ph.D. degree from the Department of Physics, University of Waterloo, in 2005. He has worked in industry with various companies, including Nortel Networks, Kymata, Canada, and Peleton on optical communications. Before joining UVic, he was a Postdoctoral Fellow with the Department of Applied Physics, California Institute of Technology, from 2006 to 2008. His research interests include optical microcavities and their applications to ultra-narrow linewidth laser source, bio, and nano photonics. He is extending his research on machine learning algorithms, with applications to spectral analysis, the Internet of Things, and indoor localization.