Robust ECG R-peak Detection Using LSTM

Juho Laitala
Department of Future Technologies,
University of Turku
Turku, Finland
jtlait@utu.fi

Mingzhe Jiang
Department of Future Technologies,
University of Turku
Turku, Finland
mizhji@utu.fi

Elise Syrjälä
Department of Future Technologies,
University of Turku
Turku, Finland
elmasyr@utu.fi

Emad Kasaeyan Naeini
Department of Computer Science,
University of California Irvine
Irvine, California
ekasaeya@uci.edu

Antti Airola
Department of Future Technologies,
University of Turku
Turku, Finland
ajairo@utu.fi

Amir M. Rahmani
School of Nursing,
Dep. of Computer Science,
University of California Irvine
Irvine, California
a.rahmani@uci.edu

Nikil D. Dutt
Department of Computer Science,
University of California Irvine
Irvine, California
dutt@uci.edu

Pasi Liljeberg
Department of Future Technologies,
University of Turku
Turku, Finland
pasi.liljeberg@utu.fi

ABSTRACT

Detecting QRS complexes or R-peaks from the electrocardiogram (ECG) is the basis for heart rate determination and heart rate variability analysis. Over the years, multiple different methods have been proposed as solutions to this problem. Vast majority of the proposed methods are traditional rule based algorithms that are vulnerable to noise. We propose a new R-peak detection method that is based on the Long Short-Term Memory (LSTM) network. LSTM networks excel at temporal modelling tasks that include long-term dependencies, making it suitable for ECG analysis. Additionally, we propose data generator for creating noisy ECG data that is used to train the robust R-peak detector. Our initial testing shows that the proposed method outperforms traditional algorithms while the greatest competitive edge is achieved with the noisy ECG signals.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**;

KEYWORDS

LSTM, noisy ECG, R-peak detection, data augmentation

ACM Reference Format:
Juho Laitala, Mingzhe Jiang, Elise Syrjälä, Emad Kasaeyan Naeini, Antti Airola, Amir M. Rahmani, Nikil D. Dutt, and Pasi Liljeberg. 2020. Robust ECG R-peak Detection Using LSTM. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3341105.3373945

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03.
https://doi.org/10.1145/3341105.3373945

1 INTRODUCTION

An electrocardiogram (ECG) shows the strength and timing of electrical activity in the heart by measuring from specific sites on the body's skin surface. Among the several standard ECG waves, the QRS complex in the center of an ECG cycle is the most visually distinct part. The correct detection of the R-peak in the QRS complex is the base of the following interpretive analysis such as heart rate extraction and heart rate variability analysis in disease diagnosis, well-being tracking as well as in research studies where autonomic nervous system activities are observed [8, 30, 35].

Among the QRS complex or R-peak detection algorithms developed in the past thirty-five years, the one developed by Pan and Tompkins [25] is the most well-known one serving as both an algorithm benchmark and the basics of several other algorithms developed after it (e.g., [12] and [15]). Many of the previously developed R-peak detection algorithms contain two main parts - signal processing to enhance the presence of QRS complex (i.e., signal derivatives, applying filters or signal transformations) and an amplitude threshold-based peak decision making [5]. Those methods are usually lightweight and could be implemented on embedded devices. However, these algorithms often only perform well with relatively clean ECG signals but are not robust enough to noises and artifacts [7, 26, 27]. On the other hand, the signal quality could vary over time, particularly in wearable devices, due to motions, electrode-skin conductance changes, or the use of dry electrodes [18, 24]. These issues call for more robust methods for R-peak detection that are more resilient to ECG signal noises such as baseline wander and muscle artifact. The need for such solutions is further pronounced if accuracy is of higher priority than real-time requirements (e.g., post-processing) or when processing can be preformed in the back-end (e.g., in the Fog or Cloud layers).

In this paper, we propose a LSTM based R-peak detection approach which is robust against quality fluctuations and high degrees of artifacts in ECG signals. In the evaluation part, our approach outperforms several existing R-peak detection algorithms with our dataset. Since LSTM networks are more capable of dealing with temporal modeling tasks by remembering long-term dependencies compared to other types of neural networks [10, 11, 13], making them more suitable for time series signals. LSTM networks can be bidirectional, multilayered or combination of the two. Bidirectional LSTM [11] is combination of two LSTMs that process the input sequence both in chronological and reversed order and that are connected to the same output layer. Therefore, for a every time step of a given sequence, bidirectional LSTM has information from the time steps preceding and following it. In multilayered LSTM networks, multiple LSTM layers are stacked on top of each other so that the output sequence of one layer forms the input sequence of the next layer [10]. This increases representational capacity of the network and allows it to learn more complex problems.

To the best of our knowledge, this is the first study predicting R-peak locations directly from raw ECG signals using LSTM. Only a few neural network solutions are found in the literature, and they use feed forward networks which lack the capability to model time dependency [26, 33].

Training a neural network requires a large volume of labels or annotated data, which is highly expensive particularly when peak-by-peak checking and manual corrections are needed. Therefore, most of the existing R-peak detection methods are developed from benchmark ECG databases. Among those, MIT-BIH Arrhythmia Database [21] is the most frequently used. However, the signals in this database are relatively noise free and thus they alone are not enough for training a robust R-peak detector. To the best of our knowledge, there are only two open databases that contain noisy ECG signals with annotated peaks [22, 27]. However, these databases are not large enough for training accurate neural network models.

As part of the proposed R-peak detection method, we introduce a noisy ECG data generator for training data augmentation. Our data generator mixes data from two open databases to create training data similar to noisy ECG signals recorded in real life. This data is then used to train LSTM network to detect R-peaks. Our method is tested with a separate dataset containing ECG signals from a clinical study with different levels of artifacts. In addition, one signal from the dataset is selected and different amounts of Gaussian noise is added to it. The proposed R-peak detector is compared with several classic R-peak detection methods using the same dataset.

To sum up, the contributions of this work are twofold:

- first, we propose a robust LSTM based solution for R-peak detection with noisy ECG signals;
- second, we propose data generator for generating noisy ECG signals to train the robust R-peak detector.

The implementation is released as open source on GitHub [1]. The rest of the paper is organized as follows: Section 2 introduces the datasets used in our study; Section 3 presents the proposed method in detail; Section 4 evaluates the proposed method and compares it against several other methods; and Section 5 concludes and discusses on this work.

2 DATASETS

Three datasets were involved in this work including two publicly available databases in the model training phase, and one clinical database in the model evaluation phase.

The data used in the training phase consists of two annotated databases, MIT-BIH Arrhythmia database [9, 21] and MIT-BIH Noise Stress Test database [9, 22]. The former one has been widely employed in R-peak or QRS complex detection studies and works as a benchmark database for detection algorithm development and testing [17]. The 48 ECG records from 47 subjects were digitized at 360 Hz, and each record covered 30 minutes. The signals in the database are in general clean and may have different types of arrhythmia. The modified lead II ECG was used in this study. MIT-BIH Noise Stress Test database contains three different noise records that represent typical noise sources in the ambulatory ECG recordings. In this study two noise records, baseline wander (BW) and muscle artifact (MA), were used. The third noise data source was simulated powerline interference, which was 60 Hz sinusoidal wave in this study.

The goal of this study is to develop an R-peak detection algorithm that is robust to noises. Thus for testing the method we use a real-world dataset containing one or several ECG contaminants, such as powerline interference, electromyographic noise, baseline wandering, or electrode motion artifact. The test dataset is part of a larger database[2], where one channel ECG signal was recorded by a portable biopotential acquisition device [28] from postoperative patients during re-examination. In total 103 minutes of lead I ECG recordings sampled at 500 Hz from 7 patients are included in the test phase. The recordings include irregular heart rhythm as well with premature ventricular contractions or arrhythmia. Nearly 19 minutes of signals are with different types of noises and visually detectable R-peaks. The ground truth labels of the R-peak locations (annotations) in the dataset are from a threshold-based automatic R-peak detector followed by manual corrections.

3 METHODS

3.1 Overview

Our development process was two fold (see Figure 1). At first phase, we used publicly available ECG and ECG noise databases to train the LSTM network. The heart of the training process was the generator function that created training data by mixing ECG signals with noise. At second phase, we implemented a wrapper function that uses the model to detect R-peak locations. Also, a filtering function was developed which allows filtering out unnaturally closely occurring R-peaks by using distance and model predictions as a decision criteria. After development, we tested our method against a real-world ECG dataset and also against one ECG signal with variable degrees of additional Gaussian noise. All of the development work was done with the Python 3 programming language while the following external libraries were also utilized: Keras [3], NumPy [23, 31], SciPy [32], TensorFlow [1] and wfdb-python [34].

[1] https://github.com/jtlait/ecg2rr

[2] http://healthscitech.org/

In addition, following external Python libraries were used during the method evaluation: BioSPPy [2], Matplotlib [14] and Pandas [20].

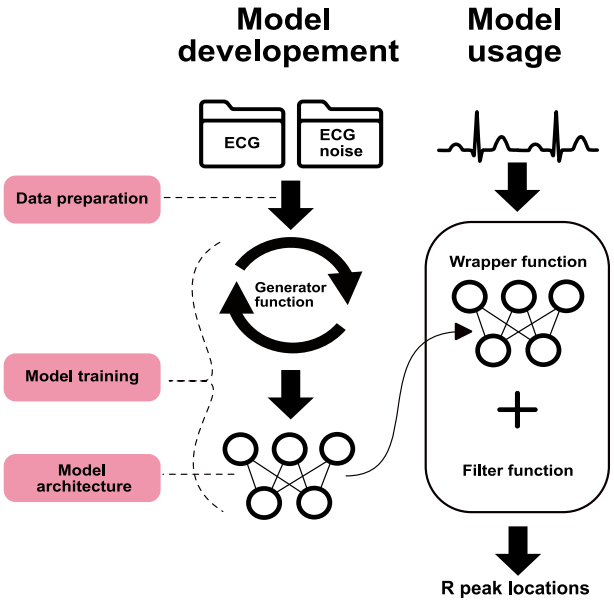

Figure 1: Overview of the proposed method.

3.2 Training data preparation

3.2.1 ECG recordings. All ECG recordings and corresponding annotations were downsampled from 360 Hz to 250 Hz. Working with lower frequency is beneficial as more temporal information can be squeezed into the same amount of sample points. This lowers the computational costs and at the same time it might help LSTM model to process temporal information. ECG annotations have different encodings for 19 different heart beat types that indicate different types of arrhythmia. From these the normal beat type is the most common, 75052 beats are classified as normal. To make problem more simple, only normal beats were selected for training. Normal beats were further filtered so that only normal beats that were located within 5 samples from the local maxima of the ECG were kept. This was done as it was noticed that in some rare cases beats that were labeled as normal occurred as local minimas (downward facing peaks) of the ECG signal.

3.2.2 Noise recordings. Downsampling from 360 Hz to 250 Hz was also done for both noise recordings. Both recordings contain two channels which differ from each other. Longer recordings from both noise sources were constructed by concatenating these channels.

3.3 Model architecture

Newest version (2.0 RC) from the TensorFlow deep learning framework with high-level Keras API was used to build and train the model. Figure 2 shows the constructed sequential model with two bidirectional LSTM layers and one dense layer. Both bidirectional layers have 64 units in each while final dense layer contains just one output unit. Hyperbolic tangent was used as activation function for all the other layers except for the final output layer which uses sigmoid as an activation function. Model contains 132 737 parameters which all are trainable. Input and output shapes are in the form (batch size, time steps, features) where the time steps and features are fixed to 1000 and 1 respectively. Model makes sequence to sequence mapping where both input and output sequences have the same length. Probability value of an time step being an R-peak is produced for every time step of the input sequence. This is illustrated in the Figure 3 where predictions produced by the model are show below the corresponding inputs.

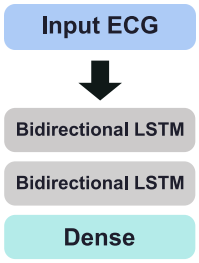

Figure 2: Schematic illustration of the network architecture.

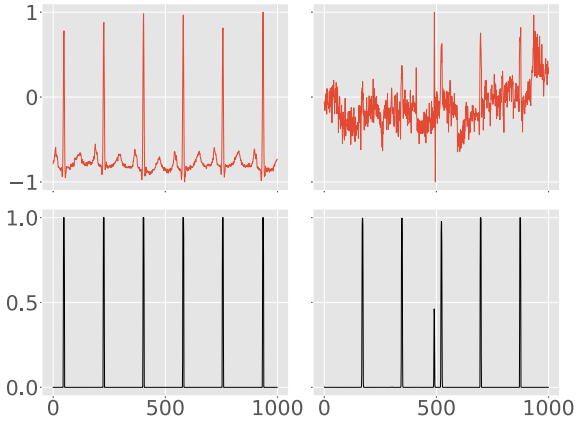

Figure 3: Upper row: Noise free and noisy ECG examples from the test dataset. Both examples have been downsampled to 250 Hz and normalized to the range [-1,1]. Corresponding predictions at the lower row. Notice the high probability (almost 0.5) produced for the noise peak.

3.4 Model training

3.4.1 Generating training data. Generator function was used to generate the training data. It was constructed so that it also takes care of the data augmentation during the training. Data augmentation was carried out by mixing ECG signals from the MIT-BIH Arrhytmia database with the baseline wander, muscle artifact and powerline interference noise sources. Former two (Figure 4) are taken from MIT-BIH Noise Stress Test Database while powerline interference is simulated with 60 Hz sine wave.

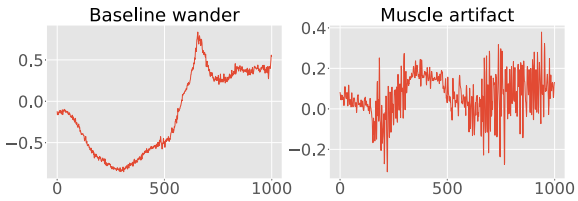

Figure 4: Examples of the used noise types from MIT-BIH Noise Stress Test Database, in 1000 sample windows.

Goal of the data augmentation phase was to generate large amount of diverse training examples that simulate real life ECG recordings. Generator function yields batches of training data and corresponding labels. Each training instance in the batch is constructed as follows (Figure 5):

(1) Randomly select one ECG record.
(2) Randomly select 1000 sample window from the ECG record.
(3) Check that all beats in the window are classified as normal.
(4) Add randomly selected noise to the window.
(5) Create numerically encoded labels for the selected window.

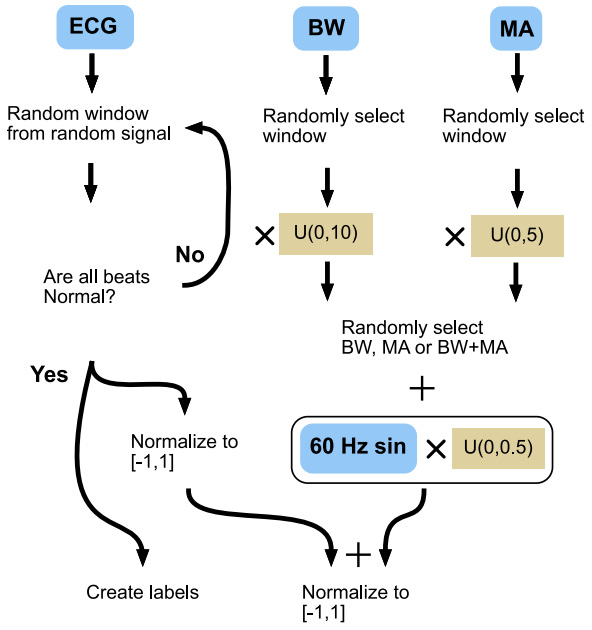

Figure 5: The construction of a single training instance in generator function. BW=Baseline wander, MA=Muscle artifact. U indicates uniform distribution where random multiplier is drawn.

From these the steps 1-3 are self explanatory but steps 4-5 warrant a more detailed description. In step 4, 1000 sample windows are first randomly selected from baseline wander and muscle artifact noise sources. Then the category of the added noise is determined randomly, it can be either baseline wander, muscle artifact or composite of these two. After category selection, selected noise source is multiplied by random number. For baseline wander random number is selected from uniform distribution (0,10) while for muscle artifact it is selected from uniform distribution (0,5). As a final step, 60 Hz sine wave multiplied by random number from uniform distribution (0,0.5) is added to the selected noise category.

Uniform distributions in all aforementioned cases were determined by visually examining the training examples produced by the generator function. Randomization was purposefully used in every possible step to maximize the variation in the training data. In this way, the generator function produces almost noise free training examples as well as examples that are saturated with complex noise. The composed noise is added to the ECG signal which has been normalized to the range [-1,1]. After noise addition, the noisy ECG signal is normalized again to the same range so that all training examples of the batch have similar range.

In step 5, binary labelling was used to label each time step of the window. Label "1" was used for points corresponding R-peak locations while rest of the time steps were labeled with zeroes. To make labels slightly more balanced, ones were added also two indices before and two indices after the R-peak index (Figure 6). This labeling scheme where one R-peak is marked with five ones can be also utilized to reduce the number of false positives when predictions are processed.

Figure 6: Schematic illustration of the ECG labeling scheme.

3.4.2 Training. Model was trained on a GPU runtime environment of the Google Colaboratory, which is a cloud based research tool for machine learning education and research. Binary cross entropy was selected as loss function, while Adam [16] was chosen as optimizer. Network was trained 150 epochs with steps per epoch being 40 and batch size of 256. This means that a total number of 1 536 000 (150x40x256) different training examples (Figure 7) were used for training.

3.5 Using model for R-peak detection

3.5.1 Prerequisites. Model expects its inputs to be in the form of 3D tensor where the information is arranged similarly as in the training phase. First axis of the tensor is batch dimension while the following two axes correspond to time steps and features. Output of the model is also a 3D tensor which has similar shape as input. Wrapper function (find_peaks) was developed to feed the ECG signal in the correct form to the model and to extract R-peak locations from

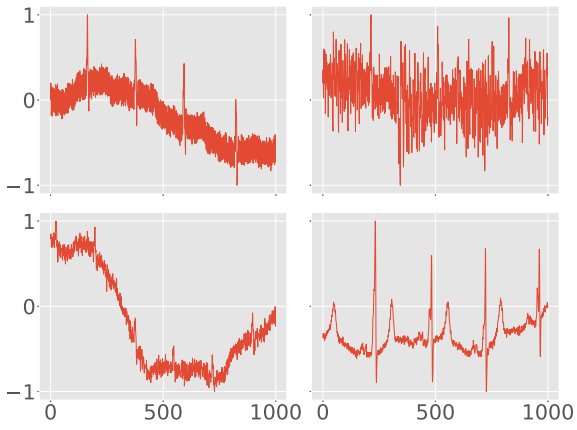

Figure 7: Examples of the training data produced by the generator function. Training example in the lower right is relatively noise free while rest of the training examples contain varying amounts of baseline wander, muscle artifacts and powerline interference.

model predictions. For many of the steps, it uses helper functions that each have their own well defined subtask. These subtasks are carried out in the following order:

(1) Resample ECG signal to 250 Hz.
(2) Split ECG signal into overlapping segments (windows).
(3) Make predictions with the model.
(4) Calculate averages for the overlapping predictions.
(5) Filter out low probability predictions.
(6) Resample R-peaks back to original sampling frequency.
(7) Correct R-peaks with respect to original signal.
(8) Remove duplicate R-peaks if present.

Steps 2 and 5-8 are described at more detail in their own subsections.

3.5.2 Wrapper function - Splitting data. ECG record is split into overlapping segments by moving the 1000 time step wide window with user defined stride. In this study, the stride of 250 was used, which corresponds to four predictions per every time step. Usage of overlapping windows adds the computational costs but at the same time it improves the R-peak detection, as time steps can be seen in a different context. Padding (median of 1000 closest time steps) is added to the both ends of the signal to allow same amount of overlap for each time step. During splitting, each 1000 time step long ECG segment is also normalized to range [-1,1]. After splitting data is reshaped into form of 3D tensor that is expected by the model.

3.5.3 Wrapper function - Filtering predictions. After overlapping predictions are averaged, one probability value for each time step remains. Time steps are then filtered based on the user defined probability threshold, only time steps above the probability threshold are selected for further evaluation. Using lower values for probability threshold increases the recall but at same time some of the precision is lost. After filtering, remaining time steps are corrected to the local maxima of the ECG signal that is within five time steps. If five or more time steps are corrected to the same location, then that location is considered to be an R-peak. The idea is to utilize the labeling scheme of training phase, where each R-peak was labeled with five time steps. This has a noise reducing effect, as R-peak cannot be identified by just one positive prediction, but it needs five closely spaced positive predictions.

3.5.4 Wrapper function - Correcting R-peaks. So far, all of the work has taken place with the ECG signal that has been resampled to 250 Hz. In step 7, identified R-peak locations are mapped to the location indices that correspond R-peak locations in the original sampling frequency. Because the R-peak location can differ slightly in the two different frequencies, correction to the local maxima is done with respect to the ECG signal in original sampling frequency. In some very rare cases two R-peak locations are corrected to the same location. Step 8 is for these situations, duplicate indices are removed if they occur. After step 8 wrapper function returns R-peak indices and corresponding probability values.

3.5.5 Filtering function. Separate filtering function (Figure 8) was developed to filter out unnaturally closely occurring R-peaks. It becomes especially useful when goal is to maximize recall and user defined probability threshold has been set low. By using filtering function, precision can then be improved by removing false positives. Typical use case for the filtering function is present at the Figure 3 where the model gives almost 0.5 probability for sharp noise peak that occurs at the middle of the noisy ECG segment.

At first, R-peaks that are within threshold distance from another R-peaks are identified. Then these R-peaks are removed from the set of all R-peaks. After this, removed R-peaks are iterated over in a descending probability order. At every iteration the R-peak with the highest probability value is selected from removed R-peaks and it is checked if it can be added to the set of approved R-peaks. If its distance to the neighbouring R-peaks is greater than threshold distance, it is added, otherwise it is thrown away. This simple algorithm works surprisingly well when variations in heart rate are not too large. More advanced adaptive method will be needed for ECG signals where heart rate varies a lot.

4 PERFORMANCE EVALUATION AND COMPARISON

4.1 Evaluation methods and results

4.1.1 Evaluation metrics. The metrics used in the performance evaluation are precision, recall, and F1 score, which are calculated from true positives (TP), false negatives (FN), and false positives (FP) by:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{3}$$

Predicted R-peak is considered true positive if it falls within one tenth of the sampling rate (in this study, 50 samples at 500 Hz sampling rate) from the ground truth.

Table 1: Performance comparison between the proposed LSTM method and selected R-peak detection methods

No.	Anno. peaks	LSTM			Hamilton			Christov			Engzee			Pan-Tompkins		
		preci.	recall	F1	preci.	recall	F1	preci.	recall	F1	preci.	recall	F1	preci.	recall	F1
1	1159	**0.991**	**0.990**	**0.991**	0.979	0.984	0.982	0.988	0.985	0.987	0.977	0.983	0.980	0.952	0.979	0.966
2	1876	**1.000**	**1.000**	**1.000**	0.995	0.999	0.997	0.997	0.999	0.998	0.953	0.985	0.968	0.984	0.995	0.989
3	932	**0.996**	**0.997**	**0.996**	0.989	0.845	0.912	0.994	0.845	0.914	0.290	0.550	0.380	0.961	0.974	0.968
4	956	**0.998**	**0.999**	**0.998**	0.973	0.998	0.986	0.986	0.987	0.987	0.965	0.992	0.978	0.955	0.991	0.972
5	1720	**0.995**	**0.987**	**0.991**	0.987	0.910	0.947	0.953	0.224	0.363	0.983	0.970	0.977	0.982	0.949	0.965
6	843	**0.999**	**0.999**	**0.999**	0.989	**0.999**	0.994	0.995	**0.999**	0.997	0.979	0.993	0.986	0.990	0.987	0.989
7	1054	**0.995**	**0.995**	**0.995**	0.984	0.994	0.989	0.993	0.657	0.790	0.983	0.990	0.986	0.977	0.973	0.975

**The best performance with precision, recall, or F1 score within the same ECG record (in each row) are highlighted in bold font.*

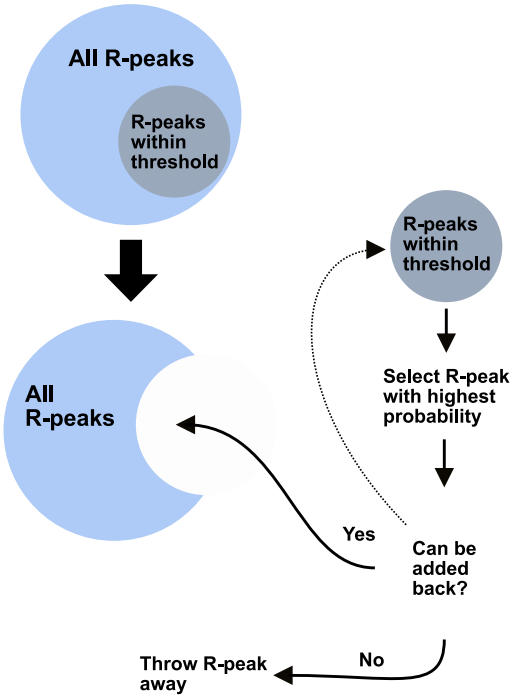

Figure 8: Working principle of the filtering function.

4.1.2 Test with annotated test dataset. Our method was first tested with the seven raw ECG recordings from the test dataset. For our method parameters, We selected 0.05 as a user defined probability threshold and stride value of 250. Then filtering function was run by using the threshold distance of 350 ms. These parameters were kept the same for all of the tested ECG recordings. To compare the performance of the proposed method with the existing methods, four classic methods were tested as well with the same ECG recordings. The other tested methods were Hamilton [12], Christov [4], Engzee [6, 19], and Pan-Tompkins [25]. In Hamilton and Engzee, the raw ECG signals were band-pass (3 - 45 Hz) filtered before detection. The inputs to Christov were raw ECG signals. For the first three methods, we used the versions that were implemented at the BioSPPy library [2]. These methods were run with their default parameters. For the Pan-Tompkins method, we used a ready-made Matlab function [29] and the input were raw ECG recordings segmented with a sliding 4-second window.

Table 1 presents the performance of the tested methods. It is clear that the proposed LSTM based method has higher performance with the whole test dataset than the classical methods. Moreover, the performance of the proposed method is more consistent as it is not so sensitive to the variations in the record quality as are the classical methods. For example, the signal quality of record No. 2 and No. 6 are generally good throughout the record, therefore the corresponding performance of Hamilton and Christov are high as well (F1 score $\geq$ 0.994). By contrast, in the record No. 3 where the signal quality is very low in some parts (Figure 3 is an extreme example in record No. 3), the F1 score of the proposed method is 0.996 while the F1 scores of Hamilton and Christov are lower than 0.92. For these two methods, the recall values drop to 0.845 indicating that many R peaks are not detected.

4.1.3 Test with SNR controlled ECG samples. To further assess the performance of our method with the noisy ECG signals, test examples with different degrees of Gaussian noise were generated. Test was carried out by using exactly similar settings for the different methods as in the previous test. Test examples were generated from ECG record No.1 by adding Gaussian noise with controlled linear SNR = P_{Signal}/P_{Noise}. The noise was added in sliding 1-second windows. The detrended raw ECG samples inside the window were the signal base and their total power was P_{Signal}. All of methods were then tested with the generated noisy ECG samples with an SNR values of 20, 10, 5, 1, 0.5, 0.4, 0.3, 0.2, and 0.1. Figure 9 shows a segment of ECG signal with different levels of additional noise. In Figure 10, F1 scores at different SNR levels are plotted for all of the methods. It is evident that our method has the highest performance in all noise levels. Performance decay is also slowest from the tested methods. This is especially true at the highest noise levels where the performances of the classical methods plummet. Even at the highest noise level (SNR 0.1), our method achieves the precision (0.939), recall (0.938) and F1 (0.939) scores that are well above 0.9.

4.2 Error analysis

Our method is the most error prone when sharp and intense base line changes associated to electrode motion or loss of contact are present in the ECG signal. Good example from this kind of situation

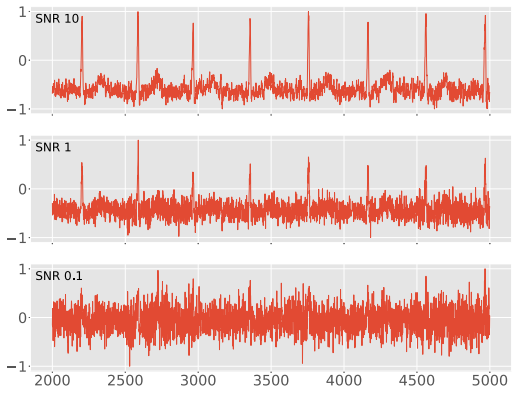

Figure 9: Examples of ECG with three different SNR levels. All signals have been normalized to range [-1,1].

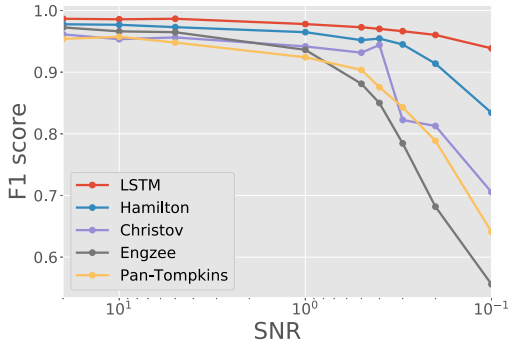

Figure 10: F1 scores at different noise levels.

can be seen on the Figure 11 where otherwise good quality ECG signal has sharp and intense noise peak at the middle. Detection failures in these situations are mostly due the fact that these types of artifacts were not simulated during the training phase. In the presence of intense base line shift, ECG values associated to R-peaks are at completely different range than they were during the training. Therefore model has no clue about the correct R-peak positions. Trying to include this noise type to training phase would be the logical next step and basis for future improvement.

5 DISCUSSIONS AND FUTURE WORK

In this paper we presented a new LSTM-based approach for R-peak detection from ECG signals. Compared to traditional methods, our method gives more robust R peak detection both with the real life ECG data and ECG data with added Gaussian noise. Our method was developed on the basis that it will be used in a offline manner. However, it would be straightforward to convert it to real-time applications.

Our initial testing gives encouraging results of the method performance. The current results may be limited to the size of test dataset. In our future work, we plan to further validate the proposed method with a publicly available ECG database recently published by Porr and Howell [27]. This database contains noisy ECG signals where

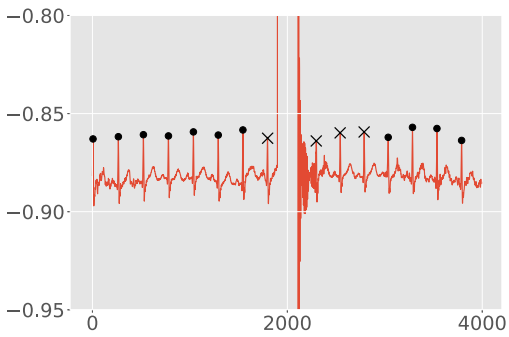

Figure 11: Detection failures at the immediate vicinity of strong electrode contact artifact. Circles are true positives while crosses represent false negatives. ECG signal has been normalized to the range [-1,1].

noise originates from the real life activities like walking or jogging. This kind of database would give more comprehensive evaluation of the method performance.

Our ultimate goal is to develop fully automated end-to-end solution to R-peak detection. This would remove the need for separate filtering function altogether. Good starting point for the network improvement can be found from the data augmentation step as more noise sources (e.g., electrode motion artifacts) should be incorporated into training data. There is also a great deal of parameters (e.g. window size, network architecture) that could be tuned to increase the performance.

ACKNOWLEDGMENTS

This research was supported by Academy of Finland project, Personalized Pain Assessment System based on IoT 313488. It was also supported partially by the US National Science Foundation (NSF) WiFiUS grant CNS-1702950 and Academy of Finland grants 311764, 313448, and 313449.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[2] Carlos Carreiras, Ana Priscila Alves, André Lourenço, Filipe Canento, Hugo Silva, Ana Fred, et al. 2015. BioSPPy: Biosignal Processing in Python. https://github.com/PIA-Group/BioSPPy/. [Online; accessed 2019-12-02].

[3] François Chollet et al. 2015. Keras. https://keras.io. [Online; accessed 2019-12-02].

[4] Ivaylo I Christov. 2004. Real time electrocardiogram QRS detection using combined adaptive threshold. *Biomedical Engineering Online* 3, 1 (2004), 28. https://doi.org/10.1186/1475-925X-3-28

[5] Mohamed Elgendi, Björn Eskofier, Socrates Dokos, and Derek Abbott. 2014. Revisiting QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems. *PloS One* 9, 1 (2014), e84018. https://doi.org/10.1371/journal.pone.0084018

[6] WAH Engelse and C Zeelenberg. 1979. A single scan algorithm for QRS-detection and feature extraction. *Computers in Cardiology* 6, 1979 (1979), 37–42.

[7] Gary M Friesen, Thomas C Jannett, Manal Afify Jadallah, Stanford L Yates, Stephen R Quint, and H Troy Nagle. 1990. A comparison of the noise sensitivity of nine QRS detection algorithms. *IEEE Transactions on Biomedical Engineering* 37, 1 (1990), 85–98. https://doi.org/10.1109/10.43620

[8] Fay C.M. Geisler, Nadja Vennewald, Thomas Kubiak, and Hannelore Weber. 2010. The impact of heart rate variability on subjective well-being is mediated by emotion regulation. *Personality and Individual Differences* 49, 7 (2010), 723–728. https://doi.org/10.1016/j.paid.2010.06.015

[9] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. 2000 (June 13). PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000 (June 13)), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215

[10] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 273–278. https://doi.org/10.1109/ASRU.2013.6707742

[11] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In *Neural Networks*, Vol. 18. 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[12] P Hamilton. 2002. Open Source ECG Analysis. *Computers in Cardiology* 29 (2002), 101–104. https://doi.org/10.1109/CIC.2002.1166717

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (nov 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14] John D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science and Engineering* 9, 3 (may 2007), 99–104. https://doi.org/10.1109/MCSE.2007.55

[15] Vignesh Kalidas and Lakshman Tamil. 2017. Real-time QRS detector using stationary wavelet transform for automated ECG analysis. In *IEEE 17th International Conference on Bioinformatics and Bioengineering*, Vol. 2018-Janua. 457–461. https://doi.org/10.1109/BIBE.2017.00-12

[16] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. (dec 2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[17] B-U Kohler, Carsten Hennig, and Reinhold Orglmeister. 2002. The principles of software QRS detection. *IEEE Engineering in Medicine and Biology Magazine* 21, 1 (2002), 42–57. https://doi.org/10.1109/51.993193

[18] Chengyu Liu, Xiangyu Zhang, Lina Zhao, Feifei Liu, Xingwen Chen, Yingjia Yao, and Jianqing Li. 2019. Signal quality assessment and lightweight QRS detection for wearable ECG smartvest system. *IEEE Internet of Things Journal* 6, 2 (2019), 1363–1374. https://doi.org/10.1109/JIOT.2018.2844090

[19] André Lourenço, Hugo Silva, Paulo Leite, Renato Lourenço, and Ana Fred. 2012. Real time electrocardiogram segmentation for finger based ECG biometrics. In *Proceedings of the International Conference on Bio-Inspired Systems and Signal Processing*. 49–54.

[20] Wes McKinney. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, Vol. 445. Austin, TX, 51–56.

[21] George B Moody and Roger G Mark. 2001. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50. https://doi.org/10.1109/51.932724

[22] Moody GB, Muldrow WE, and Mark RG. 1984. The MIT-BIH Noise Stress Test Database. In *Computers in Cardiology*. 381–384. https://doi.org/10.13026/C2HS3T

[23] Travis E. Oliphant. 2015. *Guide to NumPy* (2nd ed.). CreateSpace Independent Publishing Platform, USA.

[24] Christina Orphanidou and Ivana Drobnjak. 2017. Quality assessment of ambulatory ECG using wavelet entropy of the HRV signal. *IEEE Journal of Biomedical and Health Informatics* 21, 5 (2017), 1216–1223. https://doi.org/10.1109/JBHI.2016.2615316

[25] Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *IEEE Transactions Biomedical Engineering* 32, 3 (1985), 230–236.

[26] Sean Parsons and Jan Huizinga. 2019. Robust and fast heart rate variability analysis of long and noisy electrocardiograms using neural networks and images. (2019). arXiv:arXiv:1902.06151

[27] Bernd Porr and Luis Howell. 2019. R-peak detector stress test with a new noisy ECG database reveals significant performance differences amongst popular detectors. *bioRxiv* (2019). https://doi.org/10.1101/722397 arXiv:https://www.biorxiv.org/content/early/2019/08/08/722397.full.pdf

[28] Victor Kathan Sarker, Mingzhe Jiang, Tuan Nguyen Gia, Arman Anzanpour, Amir M. Rahmani, and Pasi Liljeberg. 2017. Portable multipurpose bio-signal acquisition and wireless streaming device for wearables. In *Proceedings of IEEE Sensors Applications Symposium*. https://doi.org/10.1109/SAS.2017.7894053

[29] Hooman Sedghamiz. 2014. Complete Pan-Tompkins implementation ECG QRS detector. https://se.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector.

[30] Fred Shaffer and J. P. Ginsberg. 2017. An overview of heart rate variability metrics and norms. *Frontiers in Public Health* 5, September (2017), 1–17. https://doi.org/10.3389/fpubh.2017.00258

[31] Stéfan Van Der Walt, S. Chris Colbert, and Gaël Varoquaux. 2011. The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering* 13, 2 (mar 2011), 22–30. https://doi.org/10.1109/MCSE.2011.37

[32] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. 2019. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. , arXiv:1907.10121 pages. arXiv:cs.MS/1907.10121

[33] Yande Xiang, Zhitao Lin, and Jianyi Meng. 2018. Automatic QRS complex detection using two-level convolutional neural network. *BioMedical Engineering Online* 17, 1 (2018), 1–17. https://doi.org/10.1186/s12938-018-0441-4

[34] Chen Xie and Julien Dubiel. 2016. wfdb-python. https://github.com/MIT-LCP/wfdb-python. [Online; accessed 2019-12-02].

[35] Sean Shensheng Xu, Man Wai Mak, and Chi Chung Cheung. 2019. Towards end-to-end ECG classification with raw signal extraction and deep neural networks. *IEEE Journal of Biomedical and Health Informatics* 23, 4 (2019), 1574–1584. https://doi.org/10.1109/JBHI.2018.2871510