Deep Learning Models for Denoising ECG Signals

Corneliu T.C. Arsene, Richard Hankins, Hujun Yin
School of Electrical and Electronic Engineering
The University of Manchester
Manchester, United Kingdom
Corneliu.Arsene@manchester.ac.uk, Richard.Hankins@manchester.ac.uk, Hujun.Yin@manchester.ac.uk

***Abstract*—Effective and powerful methods for denoising electrocardiogram (ECG) signals are important for wearable sensors and devices. Deep Learning (DL) models have been used extensively in image processing and other domains with great successes but only very recently they have been used in processing ECG signals. This paper presents two DL models, together with a standard wavelet-based technique for denoising ECG signals. First, a Convolutional Neural Network (CNN) is depicted and applied to noisy ECG signals. It includes six convolutional layers, with subsequent pooling and a fully connected layer for regression. The second DL model is a Long Short-Term Memory (LSTM) model, consisting of two LSTM layers. A wavelet technique based on an empirical Bayesian method with a Cauchy prior is also applied for comparison with the DL models, which are trained and tested on two synthetic datasets and a dataset containing real ECG signals. The results demonstrate that while both DL models were capable of dealing with heavy and drifting noise, the CNN model was markedly superior to the LSTM model in terms of the Root Mean Squared (RMS) error, and the wavelet technique was suitable only for rejecting random noise.**

Keywords*—ECG signals, Deep Learning models, Convolutional Neural Networks, Long Short-Term Memory, Filtering, Denoising, Wavelets, Filtering

I. INTRODUCTION

Electrocardiography (ECG) is a widely accepted method in the medical cardiology domain for analysing of cardiac conditions of human patients [1]. However, ECG signals are often affected by noise, random or deterministic and artefacts. These errors mix with the ECG signal generated by the human heart, making it hard to extract underlying features and interpret the ECGs. The sources of errors can be due to various events such as movements of the patient, electromagnetic noise induction of the electronic hardware situated nearby, or muscular contraction artefacts.

A large number of methods to deal with noise and/or artefacts from ECG signals have been developed, such as adaptive Filters [2,3], Independent Component Analysis (ICA) [4], Empirical Mode Decomposition (EMD) [5], adaptive Fourier decomposition [6], Savitzky-Golay filter [7], threshold method for high frequency noise detection [8], Kalman filters [9], Bayesian filter framework [10], wavelet technique [11], clustering of morphological features [12], and Neural Networks [13]. Very recent attempts include arrhythmia heart classification using different Deep Learning (DL) models [14] and QRS characteristics identification using Support Vector Machines (SVM) [15]. These methods often do not take into account the problem of very high levels of noise present in the ECGs. Similarly, in [16] a first attempt to use DL based on the Long Short-Term Memory (LSTM) models for noise rejection in ECGs was proposed while in [17] auto-encoders were investigated, but they did not consider drifting noise, which can be several times higher in magnitude than the ECG signal itself. This heavy and drifting noise is common in wearable sensors. Therefore, in this paper we investigate several DL models for the removal and rejection of such noise in ECG signals. The paper is structured as follows: the DL models are presented together with the wavelet method in section II. In section III, the datasets used are described and in section IV, experimental results are presented. Finally, conclusions are drawn and directions for eventual improvements are envisaged.

The authors would like to thank to the Engineering and Physical Sciences Research Council (EPSRC) Impact Acceleration Account (IAA 220) from the University of Manchester for supporting this work.

II. DEEP LEARNING MODELS

The first DL model investigated is based on the Convolutional Neural Networks (CNNs) [18] implemented in MATLAB [19]. CNNs have been used before for noise detection in ECG [20] but not yet for ECG reconstruction. The present CNN model was obtained by experiment and it consists of six 2-Dimensional convolutional layers, each having 36 filters with kernel size of 19x1 per filter. The first layer is an input layer of size 30000x1x1 with 'zerocenter' normalization. $M = 30000$ is the number of samples per input ECG signal sequence. Each convolutional layer has neurons that connect to parts of the input feature or connect to the outputs of the previous layer. The step size (i.e. stride) for the kernels is [1 1] while the padding is introduced so that the output is the same size as the input. Each convolutional layer is followed by a batch normalization layer with 36 channels, a rectified linear unit (ReLU) layer and an average pooling layer with stride of 4 and pooling size of 2. The succession of these layers reduces the dimensionality of the input ECG signal sequences. Before the final regression output layer, the signal goes through a fully connected layer for regression. Table I details all 27 layers of the CNN model with their characteristics. Fig. 1(a) depicts the structure of the CNN model. It is possible to assume that the DL model will be able to learn suitable filters that can be used for noise reduction so to enable the recovery of the original ECG signals. An epoch goes through the entire dataset while an iteration is the calculation of the gradient and the network parameters for the mini-batch data.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-9-0827-9703-9/19/$31.00 ©2019 IEEE

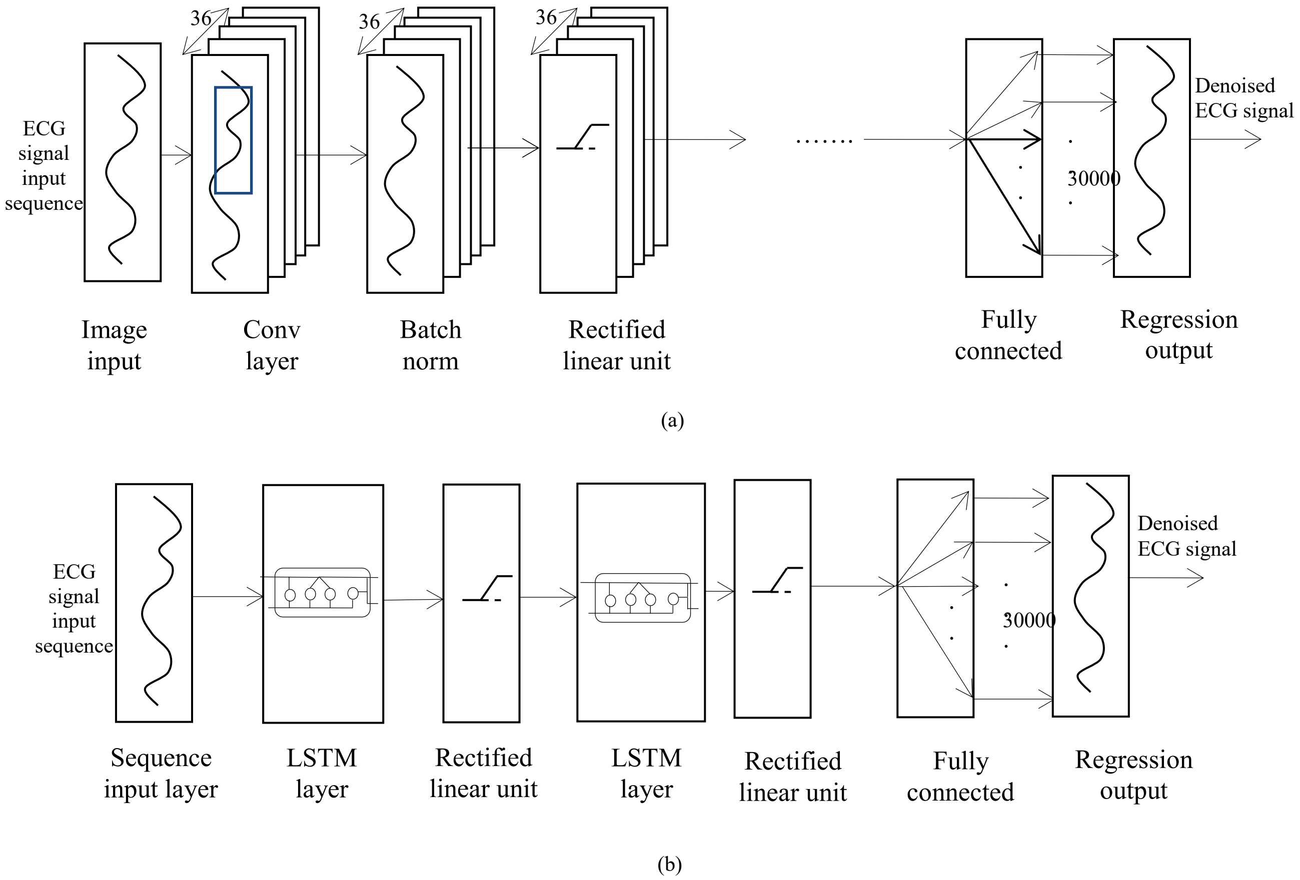

Fig. 1. Structure of the DL models: a) CNN model. b) LSTM model.

The CNN model uses the Adam optimizer and with a batch training data size of 300. It was also noticed that the obtained CNN performances did not change much with increased batch sizes. The average Root Mean Squared (RMS) error over the entire testing dataset was 0.0346, with 40 epochs of training, which took 10 minutes. For the scope of investigation, the CNN model was also left to iterate longer to 200 epochs, which took 58 minutes to reach an average RMS of 0.0299 over the same testing dataset. The CNN model was implemented in the MATLAB environment using the DL toolbox with an NVIDIA TITAN V GPU.

The second DL model is based on the Long Short-Term Memory (LSTM) layer [21, 22] and it is also implemented in MATLAB. It consists of two LSTM layers with 140 hidden nodes per layer. The first layer is a sequence input layer with the dimension similar to the ECG sequence input signal of [30000x1]. The following layer is a Long Short-Term Memory (LSTM) with 140 hidden nodes (empirically found). The LSTM layer has neurons that connect to the sequence input layer and also connect to the following layer, which is a rectified linear unit layer with 140 inputs. The second LSTM layer also has 140 hidden units and it is connected to the previous and subsequent rectified linear units. Similar to the CNN model, before the final regression output layer, the signal goes through a fully connected layer. Further LSTM layers were found not improve much the performance. Table II shows all 7 layers of the LSTM model and their properties, while Fig. 1(b) depicts the structure of the LSTM model. The model tries to learn long-term dependencies in the sequence input data, while the CNN model tries to do the same by using kernels and a deeper network structure. The LSTM model also uses the Adam optimizer for batch training and with a batch training data size of 300. A constant learning rate of 0.01 and a gradient threshold of 0.4 were used. For the same training dataset, the computational training time of the LSTM model was for 2000 epochs was about 196 minutes and with an average RMS over the entire testing dataset of 0.2321, which is significantly higher than the RMS of 0.0346 obtained with the CNN model after only 40 epochs and 10 minutes running time (or 0.0299 after 200 epochs).

For comparison, the last model used here for noise rejection is the wavelet method, which has become popular for decomposing signals in many applications including ECG noise rejection. The wavelet transformation of an input signal $x(t)$ is:

$$W_{a,b} = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \gamma^* \left(\frac{t-b}{a}\right) dt \qquad (1)$$

where t is time, $W_{a,b}$ is the wavelet transformation of $x(t)$, a is the dilation parameter, b is the location parameter, $\gamma^*(t)$ is the complex conjugate of the wavelet function which can be the Mexican-hat, Gaussian, or Daubechies wavelet function.

The wavelet method was based on an empirical Bayesian method with a Cauchy prior and implemented in MATLAB as *wdenoise* function [19], with the default 'sym4', wavelet, where 4 is the number of vanishing moments. Other options of the function included denoising method as universal threshold and noise estimate as level independent. Various parameters and options were also tried.

TABLE I.
LISTING OF THE CNN LAYERS: $M = 30000$ IS THE NUMBER OF SAMPLES PER INPUT ECG SIGNAL

nr	name and type	activations	learnable
1	Imageinput: 30000x1x1 images with 'zerocenter' normalization	30000x1x1	-
2	conv_1: 36 19x1x1 convolutions with stride [1 1] and padding 'same'	30000x1x36	Weights: 19x1x1x36 Bias: 1x1x36
3	batchnorm_1: Batch normalization with 36 channels	30000x1x36	Offset 1x1x36 Scale 1x1x36
4	relu_1: ReLu	30000x1x36	-
5	avgpool_1: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	7500x1x36	-
6	conv_2: 36 19x1x36 convolutions with stride [1 1] and padding 'same'	7500x1x36	Weights 19x1x36x36 Bias: 1x1x36
7	batchnorm_2: Batch normalization with 36 channels	7500x1x36	Offset 1x1x36 Scale 1x1x36
8	relu_2: ReLu	7500x1x36	-
9	avgpool_2: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	1875x1x36	-
10	conv_3: 36 19x1x36 convolutions with stride [1 1] and padding 'same'	1875x1x36	Weights 19x1x36x36 Bias: 1x1x36
11	batchnorm_3: Batch normalization with 36 channels	1875x1x36	Offset 1x1x36 Scale 1x1x36
12	relu_3: ReLu	1875x1x36	-
13	avgpool_3: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	469x1x36	-
14	conv_4: 36 19x1x36 convolutions with stride [1 1] and padding 'same'	469x1x36	Weights 19x1x36x36 Bias: 1x1x36
15	batchnorm_4: batch normalization with 36 channels	469x1x36	Offset 1x1x36 Scale 1x1x36
16	relu_4: ReLu	469x1x36	-
17	avgpool_4: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	117x1x36	-
18	conv_5: 36 19x1x36 convolutions with stride [1 1] and padding 'same'	117x1x36	Weights 19x1x36x36 Bias: 1x1x36
19	batchnorm_5:Batch normalization with 36 channels	117x1x36	Offset 1x1x36 Scale 1x1x36
20	relu_5: ReLu	117x1x36	-
21	avgpool_5: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	29x1x36	-
22	conv_6: 36 19x1x36 convolutions with stride [1 1] and padding 'same'	29x1x36	Weights 19x1x36x36 Bias: 1x1x36
23	batchnorm_6: Batch normalization with 36 channels	29x1x36	Offset 1x1x36 Scale 1x1x36
24	relu_6: ReLu	29x1x36	-
25	avgpool_6: 2x1 average pooling with stride [4 1] and padding [0 0 0 0]	7x1x36	-
26	fc: 30000 fully connected layer	1x1x30000	Weights 30000x288 Bias: 30000x1
27	Regressionoutput: mean-squared-error with response	-	-

TABLE II.
LISTING OF THE LSTM NEURAL NETWORK LAYERS: $M = 30000$ IS THE NUMBER OF SAMPLES PER INPUT ECG SIGNAL

nr	name and type	activation	learnable	state
1	Sequenceinput: Sequence input with 30000 dimensions	30000	-	-
2	lstm_1: LSTM with 140 hidden units	140	InputWeights 560x30000 RecurrentWeights 560x140 Bias:560x1	Hidden State 140x1
3	relu_1: ReLu	140	-	-
4	Lstm_2: LSTM with 140 hidden units	140	InputWeights 560x140 RecurrentWeights 560x140 Bias 560x1	Hidden State 140x1 Cell 140x1
5	relu_2: ReLu	140	-	-
6	Fc: 30000 fully connected layer	30000	Weights 30000x140 Bias:30000x1	-
7	Regressionoutput mean-squared error with response	-	-	-

III. DATASETS

Three datasets were used. Two comprise synthetic data generated with the software as in [23], while a third dataset is a real dataset. Each dataset was divided in a training (3/4) and a testing dataset (1/4). The first synthetic dataset has 6888 clean ECG signals (of duration 10 seconds) with 30000 samples per ECG signal (sampling rate: 3000 Hz), which was varied between 57 to 67 heart beats per minute as a test case scenario. The voltage varies between 1 mV to 3 mV. Fig. 2 shows an example of a normal synthetic ECG signal: P-wave associates the contraction or depolarization of the human heart atria, QRS-complex the contraction or depolarization of human heart ventricles, and T-wave the repolarization of human heart ventricles [1].

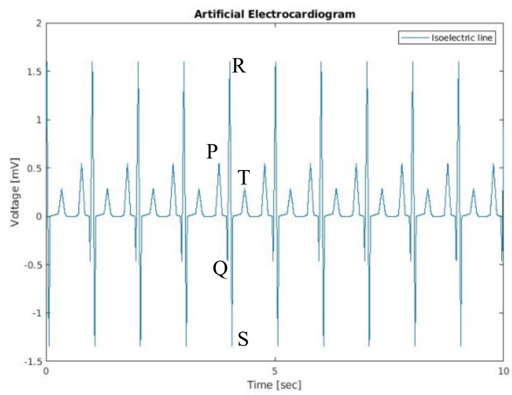

Fig.2. Synthetic normal ECG signal (10 sec) without noise comprising 30000 sample points, as the desired output of the DL models.

The first dataset also includes 6888 noisy ECG signals, which replicate the noise conditions as found in real signals [24]. Specifically, the noise can be two or three times of magnitude of the ECG signal (Fig. 3(a)) (i.e. signal-to-noise ratio (SNR) = -3dB). A further strong drift, simulated by an autoregressive process, is added to the random noise to 861 of the 6888 noisy ECG signals, as shown in Fig. 3(b) (SNR=

-7dB). The drift may correspond to various events such as movements of the patient or random limb movements.

The second synthetic dataset contains 6888 clean ECG signals and 6888 noisy ECG signals all with both random noise and various levels of drifts (SNR= -7dB). Both datasets are available upon request.

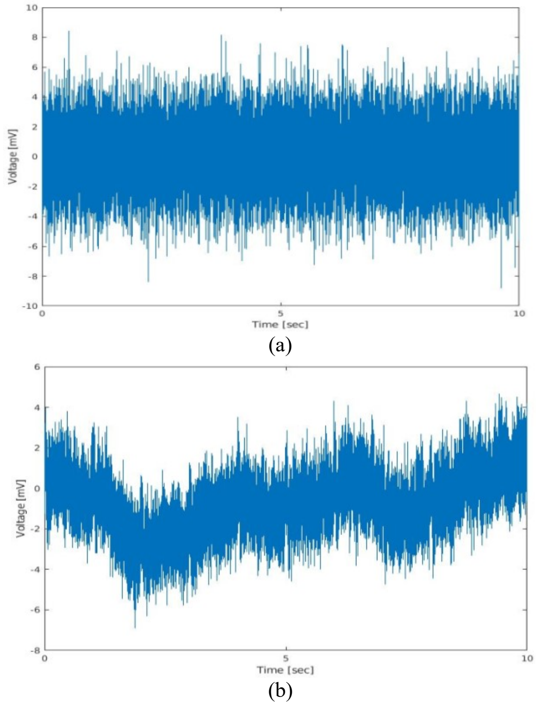

Fig. 3. ECG signal affected by noise: a) ECG signal affected by two times random noise. b) ECG signal affected by strong drifts plus random noise.

The third dataset contains real ECG signals from MIT-BIH Arrhythmia Database [25] (e.g. record 118), which can be affected by various types of noise [26] (i.e. baseline wander, muscle artifact, electrode motion artifact). Each record was obtained from 2 channels at 360 samples per second (resampled to 3000 with interpolation) with a total duration of 30 minutes. The dataset is in the Physionet WaveForm DataBase format (WFDB) [27]. The aim is to reject electrode motion artifact as the noise can be four or five times higher than the ECG signal (Fig. 4).

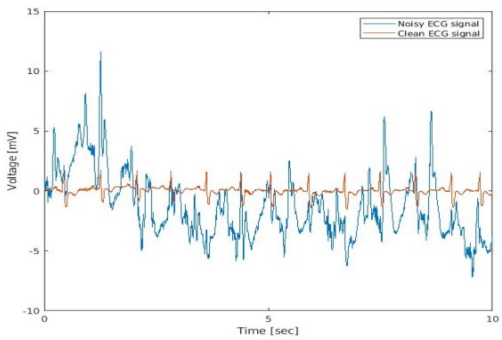

Fig.4. Real ECG signal affected by electrode motion artefact.

IV. Results

For the first synthetic dataset the predictions show that the CNN model is able to recover the original signal with the RMS value of 0.0198 for the signal shown in Fig 5(b).

For the second synthetic dataset, the LSTM predictions are less impressive with RMS value of 0.2201 for Fig. 6(b). Over the first and second testing datasets the average RMSs calculated with the CNN model were 0.0348 and 0.0299, which is 5 times lower than 0.2321 the average RMS obtained over the second testing dataset with the LSTM model.

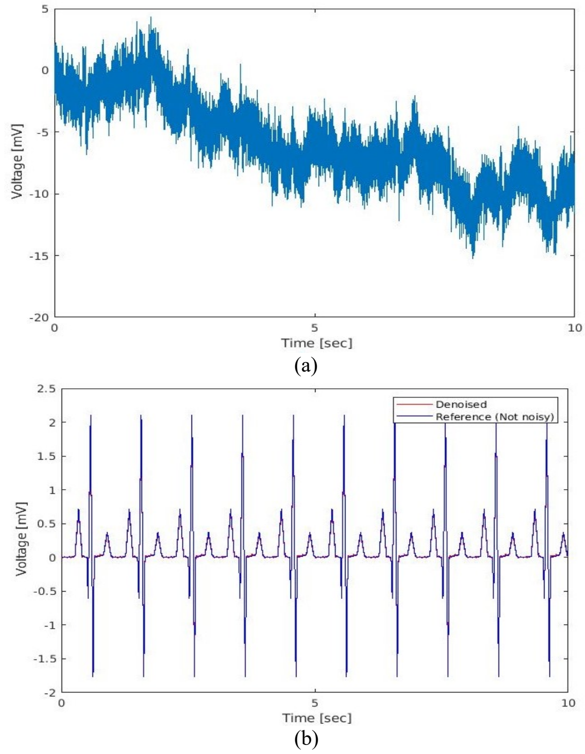

Fig. 5. CNN model: a) ECG signal affected by strong drifts and random noise, b) recovery of the original ECG signal (RMS = 0.0198).

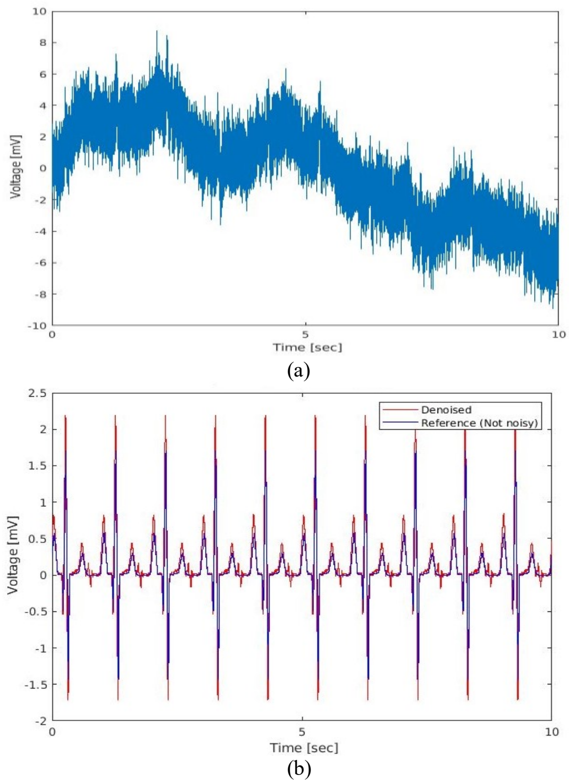

Fig. 6. LSTM model: a) ECG signal affected by strong drifts and random noise, b) recovery of the original ECG signal (RMS=0.2201).

The wavelet method was able to recover the original ECG signal in situations where the noise is random, i.e. no drifting (RMS=0.1560), the RMS being about ten times higher than that obtained with the CNN for the noisy ECG signal (2nd dataset), as shown in Fig. 7.

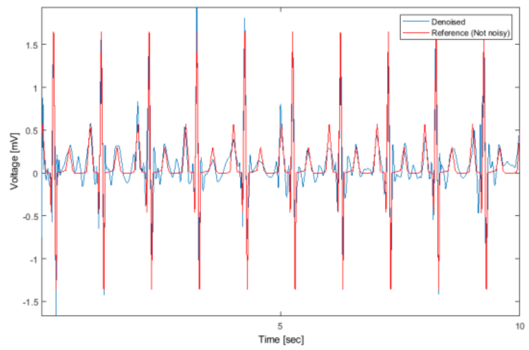

Fig. 7. Wavelet model: recovery of the original ECG signal (RMS=0.1560).

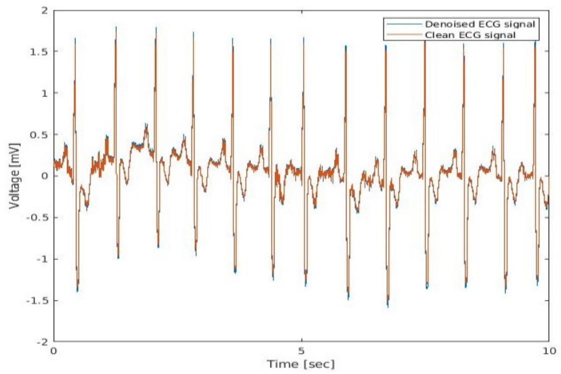

Fig. 8. CNN model: recovery of the original ECG signal (RMS = 0.0220) from noisy ECG data affected by electrode motion artefact.

The final result in Fig 8, shows that the CNN model is able to recover the original ECG signal (RMS=0.0220) by using a slightly different version of the CNN model (with kernel size: 9x1).

V. Discussion & Conclusions

A CNN model was proposed as a regression model capable of rejecting very high levels of noise in the ECG signals, a situation, which has not been addressed before. The results show that the CNN model is superior to the LSTM model in the present settings both in quality of results and computational time: the CNN model took 58 minutes and 200 epochs to achieve better results on testing dataset (i.e. average RMS =0.0299) than the LSTM model, which took 196 minutes and 2000 epochs (i.e. RMS = 0.2321). In addition, the LSTM model required four times more training samples with strong drift plus random noise as compared to the CNN model. The promising performances of the DL models, esp. CNN, were obtained on both synthetic datasets with an interval of 57 to 67 heart beats per minute and real dataset. Further work would be to test the DL models with a larger heart-beat interval such as 60-120 and variability. It will also investigate CNNs with shorter input length to reduce the latency. It is also of interest to investigate on a wide range of real datasets to further improve the performances of the CNN and LSTM models.

References

[1] R.Vecht, M.A.Gatzoulis, N.S.Peters, "ECG diagnosis in clinical practice", Springer, 2009.

[2] G.Lu, J-S.Brittain, P.Holland, J.Yianni, A.L.Green, "Removing ECG noise from surface EMG signals using adaptive filtering", Neuroscience Letters, 462, 14-19, 2009.

[3] N.V.Thakor, Y.-S.Zhu, "Applications of adaptive filtering to ECG analysis: noise cancellation and arrythmia detection", IEEE Trans. Biomed., 38(8), 785-794, 1991.

[4] J.Kuzilek, V.Kremen, F.Soucek, L. Lhotska, "Independent component analysis and decision trees for ECG holter recording de-noising", PLoS One, 9(6), 2014.

[5] J.Lee, D.D. McManus, S. Merchant, K.H. Chon, "Automatic motion and noise artifact detection in holter ECG data using empirical mode decomposition and statistical approaches", IEEE Trans. Biomed. Eng., 59(6), 1499-1506, 2012.

[6] Z. Wang, C.M. Wong, J. Nuno, F. Wan, "Muscle and electrode motion artifacts reduction in ECG using adaptive Fourier decomposition", IEEE Interm. Conf. on Syst., Man and Cyber., 2014.

[7] M.Chakraborty, S.Das, "Determination of signal to noise ration of electrocardiograms filtered by band pass and Savitzky-Golay filters", Procedia Technology, 4, 830–833, 2012.

[8] K.Le, S.Orn, O.Kleiven, "High frequency noise detection and handling in ECG signals", 26th European Signal Processing Conference (EUSIPCO), 46-50, 2018.

[9] D.Panigrahy, P.K.Sahu, "Extended Kalman smoother with differential evolution technique for denoising of ECG signal", Australas. Phys. Eng. Sci. Med., 39(3), 783-795, 2016.

[10] R.Sameni, M.Shamsollahi, C.Jutten, G.Clifford, "A nonlinear Bayesian filtering framework for ECG denoising", IEEE Trans. Biomed. Eng. 54(12), 2172-2185, 2007.

[11] M.Alfaouri, K.Daqrouq, "ECG signal denoising by wavelet transform thresholding", Amer. J. Appl. Sci., vol.5(3), 2008.

[12] J. Rodrigues, D. Belo, H. Gamboa, "Noise detection on ECG based on agglomerative clustering of morphological features", Computers in Biology and Medicine, 87, 322-334, 2017.

[13] R.Rodrigues, P.Couto, "A neural network approach to ECG denoising", arXiv:1212.5217v1, 2012.

[14] P. Rajpurkar, A.Y. Hannun, M. Haghpanahi, A.Y. Ng, "Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks", arXiv:1707.01836v1,2017.

[15] P.Xiong, H. Wang, M. Liu, S. Zhou, Z. Hou, X. Liu, "ECG signal enhancement based on improved denoising auto-encoder", Engineering Applications of Artificial Intelligence, 52, 194-202,2016.

[16] S.S.Mehta, N.S Lingayat, "Detection of QRS complexes in electrocardiogram using support vector machine", J. Med. Eng. Technol., 32(3), 206-15, 2008.

[17] K.Antczak, "Deep recurrent neural networks for ECG signal denoising", arXiv:1807.11551, 2018.

[18] K.He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", arXiv:1512.03385v1, 2015.

[19] "MATLAB System Requirements – Release 13", MathWorks, 2015.

[20] J.Jennifer, G. Conner, V. Alex,"Deep Convolutional Neural Networks for Noise Detection in ECGs", arXiv:1810.04122v1, 2018.

[21] S.Hochreiter, J.Schmidhuber, "Long short-term memory", Neural Computation, 9(8), 1735-1780, 1997.

[22] C.Hu, Q. Wu, H. Li, S. Jian, N. Li, Z. Lou, "Deep learning with a long short-term memory networks approach for rainfall-runoff simulation", Water, 10, 2018.

[23] G.A. Kumar, S. Vegi,"Analyzing of an ECG signal mathematically by generating synthetic-ECG data", IJES, 4(4), 39-44, 2015.

[24] D.Alvydas, R.Lukocius, M.Vaitkunas, G.Nedzinskaite, "Sensors and signal processing methods for a wearable physiological parameters monitoring system", Electronika Elektrotechnika, vol. 23(5), 2017.

[25] G.B. Moody, RG. Mark, "The impact of the MIT-BIH Arrhythmia Database", IEEE Eng in Med and Biol, 20(3), 45-50, 2001.

[26] G.B. Moody, WE Muldrow, "A noise stress test for arrhythmia detectors", Computers in Cardiology, 11, 381-384, 1984.

[27] AL. Goldberger, Amaral LAN, Glass L, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource of Complex Physiologic Signals", Circulation, 101(23), 2000.