Identifying Noisy ECG Signals in Large Datasets Using a Temporal Convolutional Neural Network Trained to Estimate Pseudo-SNR

Peter Doggart[1,2], Alan Kennedy[1], Daniel Guldenring[3], Raymond Bond[2], Dewar Finlay[2]

[1] PulseAI Ltd, Belfast, United Kingdom
[2] Ulster University, Belfast, United Kingdom
[3] HS Kempten, Kempten, Germany

Abstract

***Background:** Electrocardiogram (ECG) signals are often contaminated by noise. Manual review of large ECG databases to identify noisy signals is time-consuming. Traditional signal quality assessment algorithms often do not generalize well or are computationally expensive. This study developed a Temporal Convolutional Neural Network (TCNN) to estimate the signal-to-noise ratio (SNR) of ECG signals.*

***Method:** We trained a TCNN on a proprietary database of 134,019 12-lead ECGs without any machine or human-added noise labels. Assuming that this data had high SNR, we randomly selected a single lead from each ECG and added random Gaussian noise. We then scaled the signals and added noise to give a negatively skewed normal distribution of true SNR values. We trained a TCNN to regress low- and high-frequency pseudo-SNR values from the raw noisy input signals.*

***Results:** On the testing dataset, the TCNN achieved a mean error of 0.31 ± 1.80 dB and a Pearson correlation coefficient of 0.96 for low-frequency pseudo-SNR. Similarly, for high-frequency pseudo-SNR, the mean error was 0.29 ± 1.63 dB and the Pearson correlation coefficient was 0.97.*

***Conclusion:** A Temporal Convolutional Neural Network can accurately estimate the SNR of unseen ECGs.*

1. Introduction

Electrocardiogram (ECG) signals are often corrupted by noise from diverse sources, including muscle noise, motion artifact, and baseline wander. [1] These noise sources frequently overlap with the ECG frequency band of interest and can exhibit similar morphologies to the genuine ECG signal. Figure 1 depicts the normalized power spectral density of various ECG components, as well as several prevalent noise sources that overlap with the ECG spectral content. This spectral overlap makes estimating the
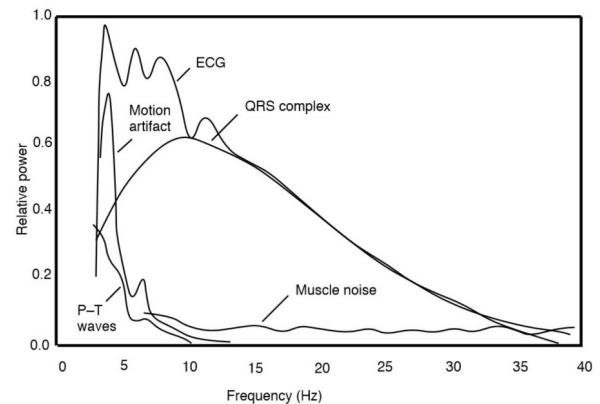

Figure 1. Relative power spectra of QRS complex, P and T waves, muscle noise and motion artifacts. [2]

signal-to-noise ratio (SNR) of captured ECG signals difficult because the pure ECG signal can never be completely separated from the inherent noise. Numerous conventional signal quality assessment algorithms have been developed, but many of them fail to generalize or have high processing requirements. The signal-to-noise ratio of ECG signals has a significant impact on the performance of all types of computerized algorithms and human interpretation, but it is rarely measured or quantified when reporting performance metrics.

The magnitude of noise-induced distortions in the ECG signal varies across the frequency spectrum. The color of the noise spectrum affects the interpretability of the ECG signal, as colored noise has a larger amplitude for a given power than white noise. This means that for signals contaminated with brown noise (such as baseline wander), the ECG signals appear much cleaner to an interpreter compared with signals with the same signal-to-noise ratio contaminated with white noise (such as muscle artifact). Figure 2 illustrates this point by showing ECG signals with three different noise colors added. It is clear from this figure that the less colored signals are more dis-
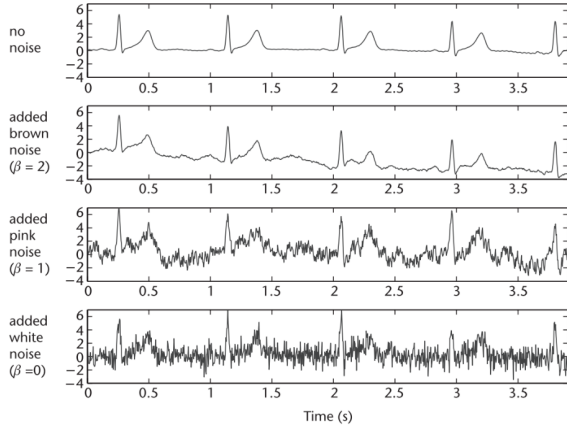
Figure 2. Zero-mean unit-variance clean ECG with additive brown, pink, and white noise (also zero-mean and unit-variance, and hence SNR = 1 in all cases). [1]

torted, although these signals have the same SNR. Due to this colorization effect, we have split the power spectrum into two distinct frequency bands and therefore report both high-frequency ($\geq$0.67 Hz) and low-frequency ($<$0.67Hz) SNRs, where 0.67 Hz corresponds to a standard ambulatory monitor high-pass filter for removing baseline wander and respiratory effects. [3]

In this paper, we propose a Temporal Convolutional Neural Network (TCNN) [4] based framework for estimating the high- and low-frequency pseudo-SNRs of unseen ECG signals. Pseudo-SNRs are estimates of the true SNR, which is never directly measurable.

2. Materials and Methods

2.1. Datasets

To train our Temporal Convolutional Neural Network (TCNN), we extracted 134,019 ECGs from the proprietary PulseAI worldwide ECG database. This database contains over 1 million labelled ECGs from patients in 7 countries. The ECGs were labelled as part of standard clinical care by a cardiologist or emergency medicine physician. For this study, we selected only ECGs that did not have any machine- or human-added noise labels and that had machine-generated median beats available. To calculate SNRs, we used the same methodology for calculating signal and noise powers as the NST application [5] in the PhysioNet WFDB software package [6, 7].

To simplify the creation of ECGs with known SNRs, we assumed that the selected ECGs represent signals with zero noise power and therefore infinite SNR. Although this assumption is never strictly true, it simplifies the creation of the signals and has a limited impact on model performance because, although such signals do contain some

noise power, it is typically very small.

To create the additive noise, we used a Gaussian noise spectrum generator to create random noise with colorization exponents in the range [0, 2]. This noise was then zero-phase low-pass filtered at 150 Hz to remove extremely high frequencies, corresponding to the cutoff frequency of most 12-lead ECG machines. [3] The noise signal was then zero-phase high-pass filtered at 0.67 Hz to remove low-frequency components, and the noise power of the remaining high-frequency noise was calculated. The residual of the 0.67 Hz filter represents the low-frequency noise power, which was also measured.

To generate a spread of target SNRs that replicates real-world data, we created target SNR distributions based on negatively skewed normal distributions with parameters $\alpha = -4$, $\mu = 24$ and $\sigma = 15$ for both high and low-frequency SNRs. We then scaled the measured ECG and noise signals to replicate these distributions, giving us our complete dataset of known SNR signals. We split this dataset into training (75%) and held-out test (25%) sets. To check for overfitting, we also tested the model on records from the Physionet Noise Stress Test Database (NSTDB) [6, 8].

2.2. Temporal Convolutional Neural Network

In this work, we trained a TCNN to predict the pseudo-SNR of previously unseen ECG signals. A TCNN architecture is well-suited for this task because dilated causal convolutions allow the network to have extremely large receptive fields, which is advantageous given the relatively high sampling frequency of ECG signals. Figure 3 shows a visualization of a stack of dilated causal convolutional layers, with the black arrows indicating the receptive field of the network on the input signal. This structure allows TCNNs to exhibit longer memory than recurrent architectures of the same size, and also allows parallelism, which reduces training and inference time.

Our TCNN was trained using Keras [10] and keras-tcn [11]. The model takes 10 seconds of single lead ECG data sampled at 360 Hz as input and regresses two numbers: the low-frequency psuedo-SNR and the high-
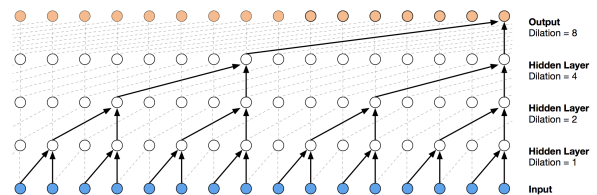

Figure 3. Simplified visualization of stacked dilated causal convolutional layers utilized in a TCNN. [9]

frequency psuedo-SNR. The model contained 8 layers, with the dilation factor increasing by 2 upon each new layer, in the same way as Figure 3. The kernel size for all convolutions was 9, and there were 128 filters per layer. The model also utilized residual blocks, dropout, and batch normalization. The model was trained using the Adam optimizer with a learning rate of 3e-4 and mean squared error loss. It consisted of 2.2 million parameters in total.

3. Results

3.1. Low Frequency Signal-to-Noise Ratio

On the held-out test set, the model achieved excellent performance in predicting low-frequency pseudo-SNR. Figure 4 shows a scatter plot of the results, with a Pearson correlation coefficient of 0.96 for the linear least-squares regression fit. Figure 5 shows a Bland-Altman plot of the true versus predicted low-frequency SNR values, with a mean difference of 0.32 dB and a standard deviation of 1.80 dB.
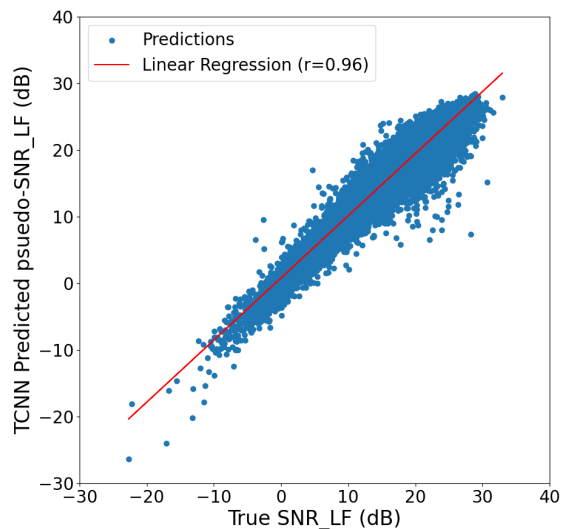

Figure 4. Scatter plot with linear regression fit showing the true low-frequency SNR versus the TCNN-predicted pseudo-SNR.

3.2. High Frequency Signal-to-Noise Ratio

The model also achieved excellent accuracy in predicting high-frequency pseudo-SNR, similar to its performance on low-frequency pseudo-SNR. Figure 6 shows a scatter plot of the high-frequency pseudo-SNR results, with a Pearson correlation coefficient of 0.97 for the linear least-squares regression fit. Figure 7 shows a Bland-Altman plot of the true versus predicted high-frequency
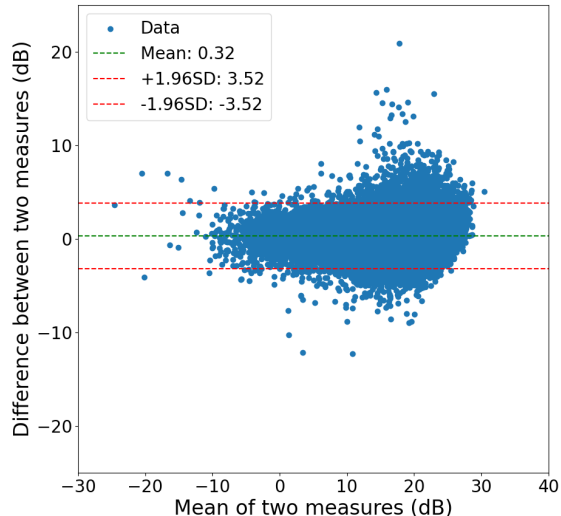

Figure 5. Bland-Altman plot of the true versus predicted low-frequency SNR values. The green line represents the mean difference; the red lines represent values for the mean ±1.96 standard deviations.

SNR values, with a mean difference of 0.29 dB and a standard deviation of 1.63 dB.

3.3. Noise Stress Test Database

To ensure that the model did not simply memorize the artificial noise generation process, we also evaluated it on records from the Physionet Noise Stress Test Database (NSTDB) [6, 8]. Figure 8 shows the time-varying pseudo-SNR values for record 118 (SNR -6 dB). For brevity, we do not present the full results for this dataset here, but it is clear that the pseudo-SNR values rise and fall corresponding to the times when noise was added.

4. Conclusion

A Temporal Convolutional Neural Network (TCNN) can accurately estimate the signal-to-noise ratio (SNR) of previously unseen ECGs. TCNNs can be parallelized and accelerated on GPUs, allowing this solution to automatically quantify SNR characteristics at scale.

References

[1] Clifford GD, Azuaje F, McSharry P, et al. Advanced methods and tools for ECG data analysis, volume 10. Artech house Boston, 2006; 71.

[2] Tompkins WJ. Biomedical digital signal processing. Prentice Hall, 1993; 237.

[3] Kligfield P Gettes LS BJea. Recommendations for the standardization and interpretation of the electrocardiogram. Circulation 2007;115(10):1306–1324.
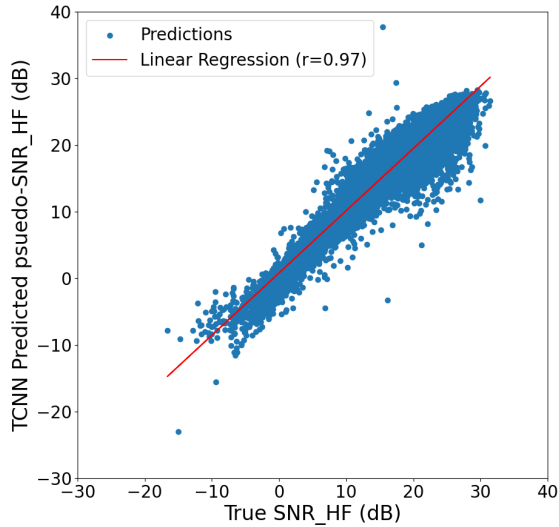
Figure 6. Scatter plot with linear regression fit showing the true high-frequency SNR versus the TCNN-predicted pseudo-SNR.
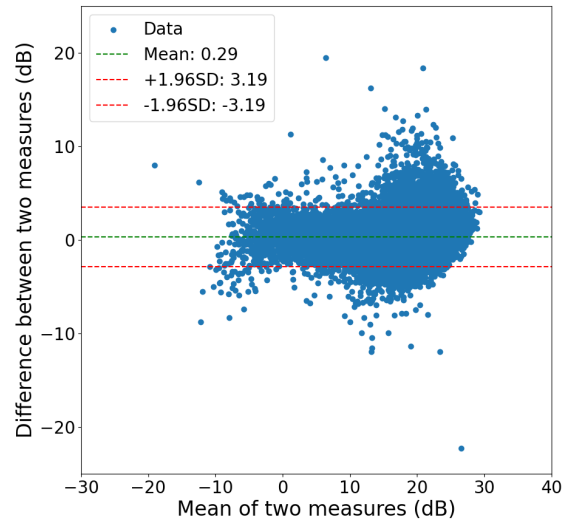
[4] Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv180301271 2018;.

[5] Moody GB. Wfdb applications guide. Harvard MIT Division of Health Sciences and Technology 2010;10:51–53.

[6] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. circulation 2000;101(23):e215–e220.

[7] Moody G, Pollard T, Moody B. Wfdb software package. https://physionet.org/content/wfdb/, 2022.

[8] Moody GB, Muldrow W, Mark RG. A noise stress test for arrhythmia detectors. Computers in cardiology 1984; 11(3):381–384.

[9] Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. Wavenet: A generative model for raw audio. arXiv preprint arXiv160903499 2016;.

[10] Chollet F, et al. Keras. https://keras.io, 2015.

[11] Remy P. Temporal convolutional networks for keras. https://github.com/philipperemy/keras-tcn, 2020.

Address for correspondence:

Peter Doggart MEng CEng MIET
PulseAI Ltd, 58 Howard Street, Belfast, BT1 6PL, UK
peter.doggart@pulseai.io

Figure 7. Bland-Altman plot of the true versus predicted high-frequency SNR values. The green line represents the mean difference; the red lines represent values for the mean ± 1.96 standard deviations.
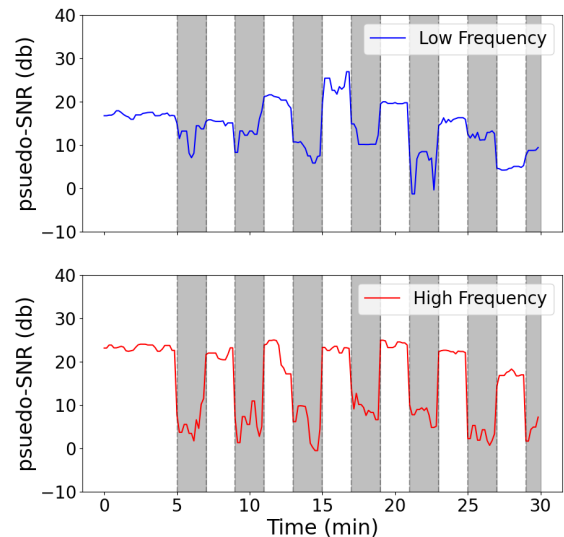

Figure 8. Time varying pseudo-SNRs for NSTDB record 118 (SNR -6 dB). The shaded gray areas indicate times when noise was added to the signal.