Autoencoders and Convolutional Neural Networks for Removing Electrode Motion Noise in ECG Signals

Laurenz Tolentino*, Macario Cordel, and Romeric Pobre
De La Salle University
**Corresponding Author: laurenz_adriel_tolentino@dlsu.edu.ph*

Abstract: The electrocardiogram (ECG) has been used as a noninvasive tool for recording heart activity, and more importantly, identifying cardiovascular diseases such as arrhythmia. As an important tool for detecting cardiac arrhythmias, it is imperative to suppress noises in the signal brought about by flaws in recording tools and patients' physiology. While adaptive filters and wavelets have been shown effective in removing baseline wander (BW) noises and muscle artifacts (MA), they are inefficient or ineffective in removing electrode motion (EM) artifacts as it may resemble QRS complexes. Machine learning has been used not only for removing BW and MA but also to distinguish between EM artifacts and valid QRS complexes. This paper aims provide a comparison between two machine learning architectures called the denoising autoencoder (DAE) and the convolutional neural network (CNN) for ECG denoising. Both implementations denoise ECG signals with signal-to-noise ratio (SNR) levels of -6dB, 0dB, 6dB, 12dB, and 18dB. Experiments show that the DAE performs more efficiently in terms of model size and training duration.

Key Words: ECG; denoising; autoencoders

1. INTRODUCTION

Electrocardiograms (ECG) have been widely used as a tool for making decisions regarding cardiovascular diseases such as arrhythmias (Guaragnella et al., 2019). Physiologists and medical professionals analyze ECG recordings of a patient to identify cardiovascular diseases in order to provide the appropriate care. As such, studies related to performing computational tasks on ECG recordings have been performed with the goal of automating the detection of cardiovascular diseases or providing valuable insights or information for medical professionals to use (Nurmaini, et al., 2020; Li, 2019; Ebrahimi, 2020). Unfortunately, the tools used for recording ECG signals also capture noise contamination (Moody, Muldrow, & Mark, 1984). These noises are categorized into either baseline wander (BW), electrode motion (EM), or muscle artifact (MA). BW artifacts are low frequency signals that are caused by motions such as breathing. MA, which is also known as EMG or muscle noise, are skeletal muscle activity that overlaps the original signal. EM artifacts, which often mimic QRS complexes, are caused by electrode-skin impedance and changes in skin potential (Moody, Muldrow, & Mark, 1984). These noises can deform an ECG signal's waveforms which can result in misdiagnosis which is why removing noise is an important and necessary step in before performing any analysis or tasks on ECG signals (Chiang et al., 2019).

Filters are one of the earliest tools used to remove BW noise. Rani et al. (2011) made a comparative study between Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters for removing BW and have concluded that IIR filters are more efficient although they are prone to oscillations. A special type of filter called an adaptive filter that are capable of handling BW and MA noise but requires reference noise which makes it difficult for wider use

(Chandrakar & Kowar, 2015; Poungponsri & Yu, 2013). Wavelets, such as the work of Mithun et al. (2011), can remove both BW and MA. Compared to adaptive filters, they do not require any reference signals and do not exhibit oscillations. Unfortunately, wavelets are not efficient in removing EM artifacts as they mimic QRS complexes (Moody, Muldrow, & Mark, 1984; Moody, Muldrow, & Mark, 1984).

Deep learning (DL) models are created through machine learning methods based on artificial neural networks and representation learning. Two DL architectures, namely, Convolutional Neural Networks (CNNs) and Autoencoders (AE) have been proven to be the most effective for time-series data such as ECG signals (Arsene, Hankins, & Yin, 2019; Chiang, et al., 2019; Zhao, Lu, Chen, Liu, & Wu, 2017; Zhao, Lu, Chen, Liu, & Wu, 2017). For CNNs, convolution and pooling operations make it effective in discovering and extracting structures in a time series input (Zhao, Lu, Chen, Liu, & Wu, 2017) while for AEs, the encoder can learn the salient features of an input and can be transformed into a denoising AE (DAE) which is an AE trained to remove noise (Chiang, et al., 2019; Im, Ahn, Memisevic, & Bengio, 2017).

As DL models can be implemented in many ways, a DAE trained to remove noise from ECG signals can be combined with a time-series focused CNN architecture. This work will focus on comparing the performances of a standard denoising autoencoder and an autoencoder with a fully convolutional network.

2. Related Work

2.1 Convolutional Neural Networks

CNNs are a specialized neural network aimed for processing data that has a grid-like topology such as time-series data which is a 1-dimensional (1D) grid by using a linear operation called convolutions. Goodfellow et al. (2016) describes CNNs typically having three stages with the first stage performing convolutions which produces a set of linear activations, the second stage is called the detector stage which makes use of nonlinear activation functions, and the last stage is the pooling function that modifies the output layer further. Arsene et al. (2016) makes use of a CNN architecture for ECG denoising with 27 layers, rectified linear unit (ReLU) as its nonlinear activation function, and a pooling stride of 4 and size 2. They added a fully connected layer just before the output for regression.

2.2 Autoencoders

An AE is trained to copy a provided input and be able to reproduce it as its output (Goodfellow, Bengio, & Courville, 2016). It starts with x as input, compressed into useful representations z via a function called the encoder $f(x)$, then undoing the compression via a function called decoder $g\big(f(x)\big)$ back into x' which is the closest reconstruction of x. Encoded representation z is produced by the encoder through minimizing a loss function which forces the AE to learn how to recreate the original input x.

Earlier uses of AEs in ECG data mainly focuses on dimensionality reduction such as data compression and feature extraction (Gudiskis & Serackis, 2016; Yildrim, Tan, & Acharya, 2018; Goodfellow, Bengio, & Courville, 2016). Dimensionality reduction is achieved in the decoder as z is constrained to have a smaller dimension than the original input x. More recently, DAEs have been used on ECG data (Chiang, et al., 2019; Fotiadou & Vullings, 2020) and have been used as a denoising phase in other tasks such as classification (Nurmaini, et al., 2020). To make an AE capable of denoising, the original input x must be corrupted with noise as $\tilde{x}$ and apply a loss function penalizing the decoder for creating a reconstruction when x' is much closer to $\tilde{x}$ rather than the uncorrupted input x. For the network architecture, Chiang et al. (2019) implemented a fully convolutional denoising autoencoder which consists of 6 convolutional layers in the encoder and 7 transposed convolutional layers in the decoder. Their encoder accepts a 1024x1 input wherein the entire record is split into fragments of 1,024 samples from a single channel. The 1024x1 input is then compressed into 32x1 via the encoder and a kernel size of 16x1 were used for the entire network alongside exponential linear units (ELU) for activation and with batch normalization for every layer. Fotiadou & Vullings' (2020) architecture for their denoising autoencoder is also similar except that there is a total of 16

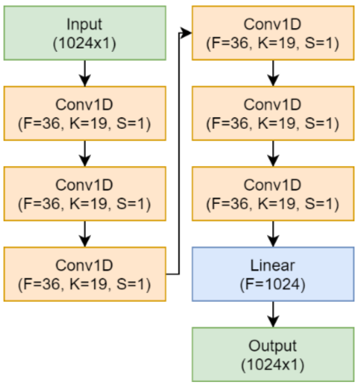

Fig. 1. CNN architecture used for experiment. Where F is filter, K is kernel size, and S is stride.

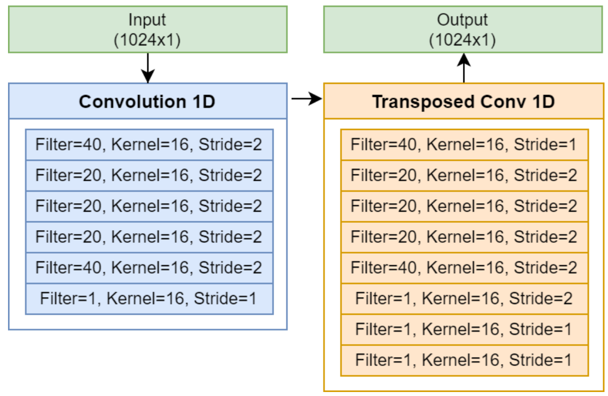

Fig 2. DAE architecture used with 6 encoder layers and 8 decoder layers.

convolutional layers, 8 of which form the encoder and the other 8 form the decoder. The network accepts a 1920x4 input wherein they have a 4 channel ECG recording and each channel were divided into fragments of 1,920 samples. Leaky rectified linear units (LeakyRelu) was used as activation after each layer. As the implementation of Fotiadou & Vullings (2020) have a deep network, skipped connections were introduced where skip connections happen every 2 convolutional layers to avoid significant loss of information due to deep and heavy subsampling.

3. METHODOLOGY

3.1 Data Preprocessing

The ECG data used for the experiment was acquired from MIT-BIH Noise Stress Test Database (NSTDB) and libraries from PhysioNet for processing (Moody, Muldrow, & Mark, 1984; Goldberger, et al., 2000). NSTDB was chosen due to its accessibility of both data and accompanying software tools. The NSTDB consists of 49 files of 30-minute ECG recordings with a sampling frequency of 360 Hz and each record has 2 channels. For the experiment, only 44 were used as the other 5 records do not have V1 lead which was the most used lead in the entire database. For the noise, NSTDB was used to introduce EM noise for all 44 records. The NSTDB introduces noise in the first 5 minutes then alternating for every 2 minutes of the record thus each record with synthetically introduced noise will have around an average of 20 minutes of corrupted signal and 10 minutes of clean signal. Each record was then divided into fragments of 1,024 samples with a total of 5,544 fragments for each SNR level used in the experiment and normalized with zero as the center.

3.3 Architecture

Two models were created for the experiment, one based on DAE and the other based on CNN. For the DAE model, the work of Chiang et al. (2019) was used as a reference while the work of Arsene et al. (2019) was used as a reference for the CNN model.

The CNN model (see figure 1) contains 7 layers with 6 are convolutional layers and the final layer as a fully connected layer and is based from the architecture used by Arsene et al (2019). After each layer, batch normalization is executed, and activation used is ReLU.

The DAE model (see figure 2) consisted of 6 layers for encoding and 8 for decoding. The encoder contains 1D convolutional layers for downsampling up to 32x1 while decoder contains transposed 1D convolutional layers to upsample the result of the encoder back to 1,024. In between each convolutional layer, batch normalization is executed, and ELU is used for activation.

3.2 Experiments

Both models were implemented on Pytorch on an Nvidia 1660 TI 6GB RAM (3 GB usable) with CUDA enabled. The DAE and CNN model both had an epoch of 10 as loss does not improve after 10 epochs on

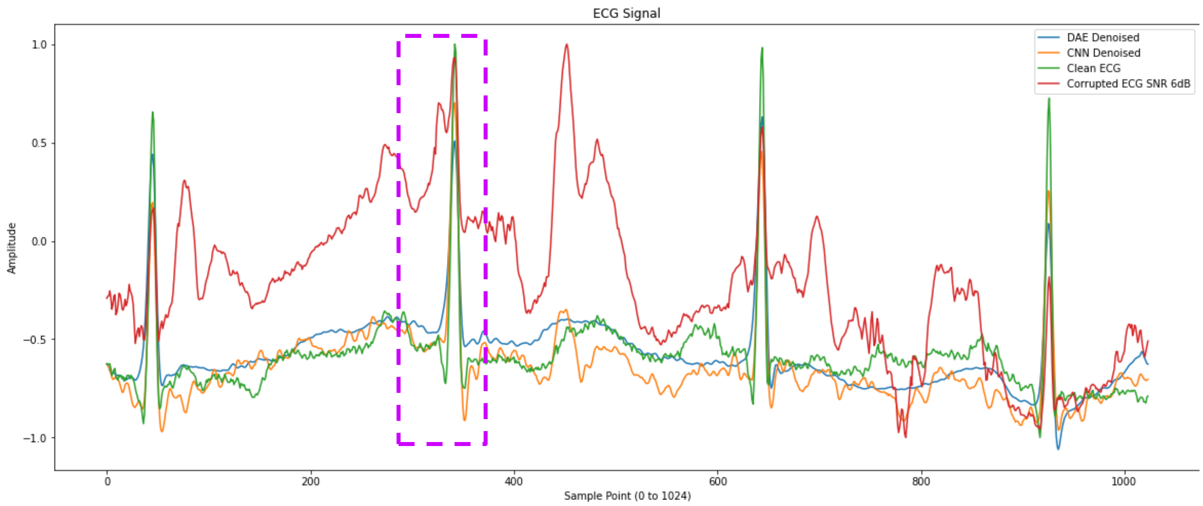

Fig. 3. Denoising result for CNN and DAE on SNR 0dB. Highlighted in purple box is improvement in recovering the P and ST segments of both approaches.

a learning rate of 1e-3. Optimizer is Adam optimization was used and both uses Mean Squared Error (MSE) for the loss function. The two models also use the same input and the same 80-20 train-test split which consists of the same 4,435 ECG fragments for training and 1,109 fragments for testing. For training, both models were trained with a record with combined SNR records of -6dB, 0dB, 6dB, 12dB, 18dB, and 24dB with 24dB is used as the ground truth or the uncorrupted signal.

3.4 Metrics

There are three evaluation metrics used for evaluation, namely: root mean square error (RMSE) (see Equation 1), percentage root mean square difference (PRD) (see Equation 2), and SNR Improvement (SNR imp) (see Equation 3) as these are the same evaluation metrics used by the studies or works mentioned in this paper.

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{N} \times \sum_{n=1}^{N} (x_i - x_i')^2} \qquad \text{(Eq. 1)}$$

$$\boldsymbol{PRD} = \sqrt{\frac{\sum_{n=1}^{N} (x_i - x_i')^2}{\sum_{n=1}^{N} x_i^2} \times 100} \qquad \text{(Eq. 2)}$$

$$\begin{gathered} SNR_{imp} = SNR_{out} - SNR_{in} \\ SNR_{out} = 10 \cdot log_{10}\left(\frac{\sum_{n=1}^{N} x_i^2}{\sum_{n=1}^{N} (x_i' - x_i)^2}\right) \\ SNR_{in} = 10 \cdot log_{10}\left(\frac{\sum_{n=1}^{N} x_i^2}{\sum_{n=1}^{N} (\tilde{x}_i - x_i)}\right) \end{gathered} \qquad \text{(Eq. 3)}$$

RMSE is used for determining the variance between output of the model and ground truth thus a lower RMSE value is better. PRD shows the recovery quality of compressed signal by measuring error between original and reconstruction thus, like RMSE, lower PRD value is better. SNR-imp measures the difference of SNR between the SNR of corrupted-to-original and reconstruction-to-original. For all three equations, x represents the original sample in the signal, $\tilde{x}$ represents corrupted version of x, x' represents the denoised reconstructed version of x.

4. RESULTS AND DISCUSSION

Through visual inspection provided in figure 3, significant noise was removed and that the QRS complex is now visible (as highlighted by the purple box). Spikes that appear in between points 400 to 600 have been brought down to manageable levels and that they cannot anymore be confused as a false QRS segment. The DAE result also tends to follow the

Table 1. Evaluation metrics of DAE and CNN models trained with all SNR levels

SNR (dB)		-6	0	6	12	18
SNR Imp	DAE	0.53	**1.52**	**0.54**	**-1.27**	**-8.04**
	CNN	**1.51**	0.39	-1.94	-6.35	-14.67
RMSE	DAE	**0.48**	0.39	0.34	0.26	0.21
	CNN	0.43	**0.44**	**0.45**	**0.46**	**0.46**
PRD	DAE	**8.80**	7.09	6.22	4.65	3.90
	CNN	7.86	**8.08**	8.26	8.34	8.36

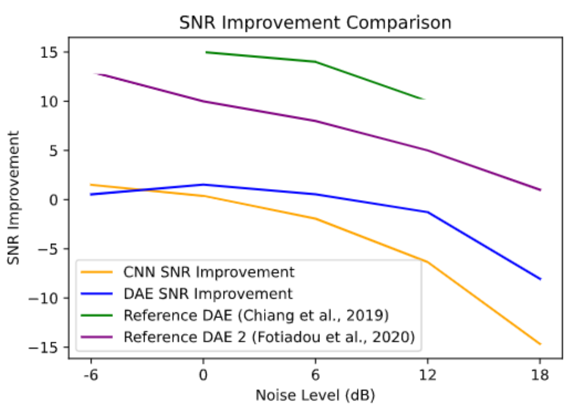

Fig. 4. SNR Improvement chart of both DAE and CNN model. Reference results are interpolated.

amplitude of the clean signal when compared to the result of the CNN output.

The quantitative results of the experiment show that DAE performs better than CNN in SNRs 0dB, 6dB, 12dB, and 18dB when both models were trained with all available SNRs as seen in table 1. In figure 4, starting at 0dB, CNNs performance drop is doubled especially in 18 dB.

Chart featured in figure 4 shows that the experiment follows the trend presented in the works of Chiang et al. (2019) and Fotiadou & Vullings (2020). The lowest SNR levels, such as -6 dB and 0 dB, show the most significant SNR improvement across all implementations. The works of Chiang et al. (2019) and Fotiadou & Vullings (2020) showed higher SNR improvements as those works were also trained to remove BW and MA noise, exposing their networks to more types of noise and much more longer training when compared to EM only even though this paper only adds EM noise to the experiment data. Those papers also were trained in different SNR levels such as Chiang et al. (2019) who only trained their DAE with -1 dB, 5 dB, and 7 dB while Fotiadou & Vullings (2020) trained their DAE from -20 dB and 20 dB in increments of 5, though it is important to take note that Fotiadou & Vullings (2020) work used a different data set from this paper's experiment as well as Chiang et al (2019).

5. CONCLUSION

This paper shows that DAE outperforms CNN in terms of quantitative metrics such as SNR-imp, PRD, and RMSE. In terms of training duration and size, the DAE took around 20 minutes of training while the CNN averaged around 47 minutes. For future works, QRS identification can be used to count detected QRS complexes as a form of qualitative measurement as being able to reveal the QRS complexes from high amounts of noise can already helpful. Future works will also include other types of noise as this paper only focuses on electrode motion as filters are already capable of removing BW and MA noise types. Future works should also discuss what SNR noise levels that would reflect real world scenarios (or specific scenarios) rather than relying on the provided SNR levels by MIT-BIH.

6. REFERENCES

Arsene, C., Hankins, R., & Yin, H. (2019). Deep learning models for denoising ECG signals. *27th European Signal Processing Conference.*

Bagheri, F., Ghafarnia, N., & Bahrami, F. (2013). Electrocardiogram (ECG) signal modeling and noise reduction using Hopfield Neural Networks. *Engineering, Technology & Applied Science Research.*

Bai, Y.-W., Chu, W.-Y., Chen, C.-Y., Lee, Y.-T., Tsai, Y.-C., & Tsai, C.-H. (2004). Adjustable 60Hz Noise Reduction by a Notch Filter for ECG Signals. *IEEE Instrumentation and Measurement Technology Conference.*

Chandrakar, C., & Kowar, M. (2015). Denoising ECG signals using adaptive filter algorithm. *International Journal of Soft Computing and Engineering*.

Chiang, H.-T., Hsieh, Y.-Y., Fu, S.-W., Hung, K.-H., Tsao, Y., & Chien, S.-Y. (2019). Noise reduction in ECG signals using fully convolutional denoising autoencoders. *IEEE Access, 7*, 60806-60813.

Ebrahimi, Z. L. (2020). A review on deep learning methods for ECG arrhythmia classification. *Journal of Expert Systems with Applications*.

El B'Charri, E., Rachid, L., Elmansouri, K., Abenaou, A., & Jenkal, W. (2017). ECG signal performance de-noising assessment based on threshold tuning of dual-tree wavelet transform. *Biomedical Engineering Online, 16*(1).

Fotiadou, E., & Vullings, R. (2020). Multi-Channel Fetal ECG Denoising With Deep Convolutional Neural Networks. *Frontiers in Pediatrics*.

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P., Mark, R., . . . Stanley, H. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Guaragnella, C., Rizzi, M., & Giorgio, A. (2019). Marginal component analysis of ECG signals for beat-to-beat detection of ventricular late potentials. *Electronics, 8*(1000).

Gudiskis, A., & Serackis, A. (2016). Auto-Encoder Based ECG Signal Feature Extraction for Real-Time Detection of Cardiac Arrhythmias. *Biomedical Engineering*.

Im, D., Ahn, S., Memisevic, R., & Bengio, Y. (2017). Denoising criterion for variational auto-encoding framework. *31st AAAI Conference on Artificial Intelligence*, 2059-2065.

Li, F. L. (2019). Feature extraction and classification of heart sound using 1D convolutional neural networks. *Eurasip Journal on Advances in Signal Processing*.

Mithun, P., Pandey, P., Sebastian, T., Mishra, P., & Pandey, V. (2011). A Wavelet Based Technique for Suppression of EMG Noise and Motion Artifact in Ambulatory ECG. *33rd Annual International Conference of the IEEE EMBS*.

Moody, G., Muldrow, W., & Mark, R. (1984). A noise stress test for arrhythmia detectors. *Computers in Cardiology*. IEEE.

Nurmaini, S., Darmawahyuni, A., Mukti, A., Rachmatullah, M., Firdaus, F., & Tutuko, B. (2020). Deep learning-based stacked denoising autoencoder for ECG heartbeat classification. *Electronics*, 1-17.

Poungponsri, S., & Yu, X.-H. (2013). An adaptive filtering approach for electrocardiogram signal noise reduction using neural networks. *Neurocomputing, 117*.

Rani, S., Kaur, A., & Ubhi, J. (2011). Comparative study of FIR and IIR filters for the removal of Baseline noises from ECG signal. *International Journal of Computer Science and Information Technologies, 2*(3), 1105-1108.

Rodrigues, J., Belo, D., & Gamboa, H. (2017). Noise detection on ECG based on agglomerative clustering of morphological features. *Computers in Biology and Medicine*.

Yildrim, O., Tan, R., & Acharya, R. (2018). An efficient compression of ECG signals using deep. *Cognitive Systems Research*.

Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 162-169.