Attention Based Convolutional Denoising Autoencoder for Two-lead ECG Denoising and Arrhythmia Classification

Prateek Singh*, *Student Member, IEEE*, and Ambalika Sharma*, * Electrical Engineering Department, Indian Institute of Technology, Roorkee, Uttrakhand 247667 India

***Abstract*—This paper proposes a novel attention-based convolutional denoising autoencoder model to filter and diagnose atrial fibrillation (AF) from excessively noisy wearable electrocardiograms (ECGs). First, the features are extracted from the ECG signal using convolutional layers, along with the noise removal, and then they are used to classify the AF. The model employs skip-layer connections to reduce information loss during the reconstruction of the denoised signal. In addition, an efficient channel attention module that efficiently updates the features retrieved via cross-channel interaction, allowing the network to pay more attention to the features of relevant information between the channels. The model is trained and tested using four widely available databases, and simulated additive white Gaussian noise from -20 to 20 decibels (dB) and MIT-BIH Noise stress test database (NSTDB) noises with SNRs -6 to 24 dB are added to the database for evaluation. The model achieved an average SNR improvement of 28.07 ± 1.67 and PRD of 8.1% at 0 dB SNR noise added to the databases. The model classification performance is found to be 99.25% precision, 99.50% recall, and 99.25% accuracy, respectively.**

***Index Terms*—Electrocardiogram, Atrial Fibrillation, Denoising Autoencoder, Convolutional Neural Network, Arrhythmias.**

I. Introduction

An electrocardiogram (ECG) is a non-invasive procedure that records the electrical activity of the heart and aids in the diagnosis of cardiac disease. ECG procedures are preferred over other types of heart monitoring because they are the most convenient and cost-effective. Recent advancements in remote monitoring technology have enabled ECGs to be performed in the privacy and convenience of one's own home or office rather than in a hospital setting. Remote monitoring technologies have improved healthcare for those with periodic cardiac arrhythmias because they can continually monitor heart activity. Now, with these advancements, massive amounts of ECG data are being gathered simultaneously, which needs processing and interpretation. Despite being the most often used diagnostic tool, computer-read ECGs have been found to have considerable errors [1]. This means that conventional algorithms can no longer be relied upon as a primary diagnostic tool. To make this situation worse, there is a severe shortage of cardiac specialists in low and middle-income countries [2]. For this reason, a simple, easy-to-use, reliable, and accurate ECG analysis tool is required that can use to deliver better insights on these remote monitoring devices.

Corresponding author: Prateek Singh (email: psingh@ee.iitr.ac.in).

Over the last few decades, researchers have developed a number of computer-assisted ways for automatically detecting arrhythmias. The fundamental goal of arrhythmia detection is to classify each heartbeat into one of several classes based on differences in morphology. The traditional method of detecting arrhythmias includes several steps, such as pre-processing, feature extraction, dimensionality reduction, and classification. Feature extraction is at the heart of most machine learning tasks, and domain experts in the field carry it out. The extracted features determine the classification accuracy. There are several feature extraction methods employed for the ECG classification task, such as wavelet transform [3], principal component analysis (PCA) [4], independent component analysis (ICA) [5], some higher-order statistics (HOS) [6]. These features are given to several classifiers for interpreting the results. Another difficulty with arrhythmia classification is that the ECG data is contaminated with noise during acquisition. The main noise sources are interference in the device's power supply, baseline drift, muscle noise, and noise from electrode contact with the skin. The presence of noise in the signal might induce changes in the amplitude and time intervals and can be mistakenly identified as one of the arrhythmias, leading to misdiagnosis. Therefore, filtering those signals is always necessary to avoid false positives and erroneous diagnoses. As a result, adequate pre-processing is required before analyzing the ECG signal. Denoising ECG signals can be accomplished using various methods, the majority of which rely on traditional techniques based on parameters that are particularly susceptible to noise, such as fixed filters like finite impulse response filters (FIR) and infinite impulse response filters (IIR) [7]. Fixed filters eliminate all signals in the cut-off frequencies, obliterating the ECG signal information in those frequencies as well. Therefore, adaptive filtering techniques [8] are used to denoise ECG signals. However, they often require noise reference signals as input, which are difficult to collect using the ECG signal acquisition system. ECG signals are non-stationary in nature, so they can be represented in the time-frequency domain to acquire the necessary information. Thus, time-frequency based techniques such as wavelet [9] and empirical mode decomposition (EMD) [10] have gained popularity and shown promising results in ECG denoising. These time-frequency filtering methods have two flaws: first, they only focus on low- or high-level characteristics, and second, for low frequencies, they give greater frequency resolution but poor time resolution; and for high frequencies, they provide better frequency resolution but poor time resolution [11].

Recent studies advise using deep learning (DL) algorithms to denoise ECG signals based on the above analysis. One such DL architecture used to filter ECG signals is the Denoising Autoencoder (DAE). In [12], the ECG signal is denoised using a stacked contractive denoising auto-encoder. They also demonstrated a DNN-based DAE for noise reduction in the ECG, with significant improvements in the signal to noise ratio (SNR) and root mean square error (RMSE) on noise-induced from the MIT-BIH noise stress test database (NSTDB), ranging from 0 to 5 dB. In [13] authors combined DAE and wavelet transform with soft thresholding to achieve better denoising in the range of 0 to 5dB on NSTDB. In [14] authors proposed a 13-layer fully convolutional network (FCN) based DAE for denoising ECG signals. They enhanced the SNR by 15.49 decibels using the NST database, and they investigated noise corruption levels ranging from -1 to 7 decibels. Some of the recent studies that used DL techniques to denoise ECG signals are discussed further. The authors of [15] presented a multi-kernel linear and non-linear (MKLANL) module for baseline wander (BLW) removal that is inspired by the Inception module and achieved highly promising results when compared to state-of-the-art signal processing approaches. The study's only drawback is that it focused solely on reducing baseline wander noise and ignored other noises. In [16] authors proposed a two-stage denoising approach. In the first stage, a U-net model is used to remove the noise from the ECG. Then, a DR-net is designed in the second stage for the detailed restoration of the ECG signal. The experiment demonstrates the applicability of this strategy in the presence of significant noise. Despite that, the study cannot remove noises with low SNRs. Nowadays, with wearable devices, such as smartwatches and patches, it is possible to monitor heart disease continuously. The ECGs collected from these devices are highly contaminated with noise as the individual moves and possibly runs while carrying these devices. Despite the researcher's best efforts, denoising of ECG beats remains a challenge, especially for noisy or low SNR conditions.

A. *Contributions and Organization*

To address above mentioned issues of not being able to denoise low SNR ECG signals, this study proposes a deep convolutional denoising autoencoder network with symmetric skip-layer connections for two-channel ECG denoising. The model is inspired by the work of [17] on image restoration; however, the hyperparameters are redesigned for 1D ECG time series data. Furthermore, the efficient channel attention (ECA) structure is introduced to efficiently update the features retrieved via cross-channel interaction, allowing the network to pay more attention to the features of relevant information. As a result, the complexity of CDAE was optimized owing to the skip connections. The network is trained and tested using four widely used and readily available ECG databases. The databases are corrupted with several types of noise, including baseline wander(BW), muscles artifact (MA), electrode motion (EM), and additive white Gaussian noise (AWGN), to test the effectiveness of the proposed model. The results are compared to state-of-the-art techniques. The main contributions of the study are as follows:

- A novel neural network model named attention-based convolutional denoising autoencoder (ACDAE) is proposed with symmetric skip-layer connections for two-channel ECG denoising.
- The efficient channel attention (ECA) mechanism is integrated for the denoising and classification tasks which will help restore ECG morphology.
- Two models are used to evaluate the classification performance of the atrial fibrillation using compressed features obtained from the encoder phase of the model. Furthermore, this study investigated the classification performance of different SNRs of noise introduced.
- To the best of the author's knowledge, this is the first comprehensive attempt to investigate the different case scenarios that influence classification performance from noisy ECG signals.

The rest of this article is organized as follows. Section II describes the dataset used and pre-processing steps. In Section III, the primary methods, including DAE and ECA, are briefly reviewed. The denoising problem to be solved is formulated and describes the proposed attention-based convolutional denoising autoencoder in detail. Section IV presents three case studies to evaluate the performance of the proposed method. Section V discuss the study and compare the results with state-of-the-art methods. Finally, conclusions are provided in Section VI.

II. Dataset Used and Pre-processing

In this study, four prominent freely available databases from the Physionet are used [18]. MIT-BIH Atrial fibrillation database (AFDB) and MIT-BIH Normal Sinus Rhythm Database (NSRDB), MIT-BIH Arrhythmia (BIHA) Database and MIT-BIH Noise Stress Test Database (NSTDB) are used to train and test ACDAE denoising and AF classification models.

AFDB includes twenty-five 10 hours of two-channel ECG recordings sampled at 250Hz. Two records, "04936" and "05091", were not used out of twenty-five records due to incorrect annotations. These recordings come with rhythm annotation files that have been painstakingly generated. If one or more beats in a beat sequence show signs of AF, the beat sequence is classed as AF (usually paroxysmal), while all other beats are classified as normal in the database.

NSRDB includes 18 long-term ECG recordings. No arrhythmia beats are available except sporadic ectopy. The recordings were digitized at 128 samples/second/channel. A reference annotation file specifying the location and kind of each beat is included with each digitized recording.

The MIT-BIH Noise Stress Test Database (NSTDB) [19] contains 12 half-hour ECG recordings and three half-hour recordings of noise typical in ECG recordings. The database includes noise records such as baseline wander 'bw', electrode motion 'em', and muscles artifacts 'ma' that can be added to ECG records to create noise stress test records.

The MIT-BIH Arrhythmia (BIHA) Database has 48 two-lead ECG records of 30-minute duration. The sampling frequency of BIHA is 360Hz which was re-sampled at 250 Hz for our experimentation. The network NSTDB noise is added for denoising purposes for training and testing.

A. Preprocessing of ECG signals

Since each database has a distinct sampling frequency, thus they all are re-sampled to 250 Hz. The pre-processing is divided into the following steps: First, the two-channel ECG data from the AFDB database is segregated into atrial fibrillation and normal beats using an annotation file. Then, the signals normalized between 0 and 1. Next, the signals are segmented so that they can be given to the model for training and testing. We need at least two R peaks in a segmented window for atrial fibrillation classification. Hence, R peaks are detected using Christov segmenter algorithm from BioSPPy toolbox [20]. A Window of 1.2 sec is taken as the normal range for RR interval is 0.6-1.2 seconds. Normal ECG signals RR intervals are much larger than AF signals. Therefore, proper zero-padding is performed to ensure that each window has the same size. Similarly, NSRDB is also pre-processed. Data prepared is shuffled and split into training, testing and validating data in the ratio of 80:10:10 percentage, respectively. To remove the effect of imbalanced data, 30,000 samples for AF and non-AF is chosen randomly, a total of 60,000 samples.

B. Data preparation for experiments

Some studies suggest testing models on noises ranging from 12 dB to -6 dB, as the majority of wearable devices may produce noise in this range [21]. However, to examine the effectiveness of the proposed model, we have added the AWGN noise with levels ranging from -20dB to 20dB, with a 5dB step size.

A total of two experiments are carried out in this research, as follows:

- First experiment is used to evaluate the ECG denoising performance and classification performance of atrial fibrillation (AF). For this experiment, AFDB and NSDB pre-processed data are mixed with the AWGN. So, overall 9 experiments are done with SNR levels of -20, -15, -10, -5, 0, 5, 10, 15, 20 dB.
- Second experiment is carried out to evaluate the suggested model on more realistic noise experienced by ECG and the BIHA, and NST databases are used to train and test the model performance. The NSTDB comes with noise records such as 'bw', 'em', and 'ma', therefore these are added with varying SNR levels of -6 dB, 0 dB, 6 dB, 12 dB and 24 dB to BIHA ECG records. The experiment is also performed using AWGN with -20 dB to 20 dB SNRs. A total of 47 records are used as two records belong to the same subject. Thirty records are used for training, five records for validation, and 12 for testing. The experiment results are discussed in sections IV and V.

III. Methodology

The proposed method is shown in Fig. 1. It has two primary functions: 1) an Attention-based Convolutional Denoising Autoencoder (ACDAE) that aids in the denoising of ECG signals, and 2) two classification modules. Classification module 1 (M1) takes the encoder's learnt feature maps and passes them to the ECA module, where cross-channel weighted attention is given to the important feature maps before passing them to the fully connected (FC) layer for atrial fibrillation classification. Classification module 2 (M2) also uses the encoder's learnt feature maps but now, instead of the ECA module, it is fed to the global pooling average (GAP) and FC layer. The goal of this analysis is to see how ECA attention impacts classification. Detailed information about each module is described in the following subsections.

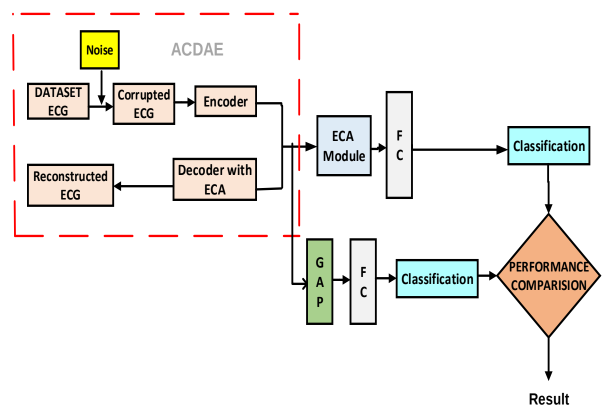


Fig. 1: Flowchart of Proposed Method

A. Convolutional Denoising Autoencoder

Autoencoder (AE) is an excellent unsupervised learning approach for extracting data feature representation. They are commonly utilized in biomedical applications for compression, feature extraction, and denoising. In [22] authors used long-short term memory (LSTM) based autoencoders for ECG arrhythmia detection. Apart from ECG signals, AE is also used in machine applications, such as the authors of [23] [24] [25] [26] used different AE networks for fault detection and to predict the health state of the machine. AE is divided into two paths: encoding, which compresses signals by learning features, and decoding, which expands the compressed signal back to its original shape. The repeated convolutional layer is followed by a non-linear activation function and a max-pooling operation for downsampling in the encoder path. On the other hand, the decoder path starts with transposed convolutional layers followed by upsampling layer and activation function and will give the denoised signal by reconstructing the input. The mathematical formulation is as follows:

The encoding path $h = c(x;\theta)$ translates a given input, $x \in [0,1]^n$, to a hidden layer, $h \in [0,1]^m$, with parameters θ and $n, m \in \mathbb{N}$. The decoding path $\hat{x} = f(h;\theta')$ converts h into a reconstruction in the input space $\hat{x} \in [0,1]^n$. The AE's training goal is to identify parameters θ, θ' that minimise the reconstruction error $L(x,\hat{x})$, i.e., the difference between x and $\hat{x}$ for all $x^i, i \in (1, ...\tau)$ samples in the training set:

$$\theta, \theta' = argmin[\frac{1}{\tau}\sum_{i=1}^{\tau} L(x_i, \hat{x}_i)] \tag{1}$$

To calculate the reconstruction error, traditional squared error $L(x, \hat{x}) = \|x - \hat{x}\|^2$ can be used or the cross entropy error function as

$$L(x, \hat{x}) = -\sum_{n=1}^{\tau}[x_n log\hat{x}_n + (1 - x_n)log(1 - \hat{x}_n)] \quad (2)$$

The hidden layer algorithm can learn input features along the primary axes of variation coordinates by minimizing reconstruction errors. They follow the same principle used in the principal component analysis (PCA). Data are projected onto the primary component that captures the most important data. Some relevant information may be lost during the compression of the original feature map. It is designed to have a small reconstruction error for test data but not for data randomly selected from input space because it compresses the training data.

As the AE model task is to reconstruct the input, they should be sensitive enough to recreate the original observation but insensitive enough to the training data, i.e. the model should not learn the input while training, which will cause overfitting. Therefore, [27] proposed a study where they corrupted the input by introducing some noise so that model now does not simply develop a mapping that memorizes the training data because now input and target output are different. Instead, the model learns a vector field for mapping the input data towards a lower-dimensional manifold.

Here, each training example x is the ECG signal. It is corrupted by a stochastic mapping $\hat{x} = q(\tilde{x}|x)$, i.e., AWGN is added to the input data, which partially destroys it according to the destruction rate (based upon SNR). The DAE then uses the encoding and decoding functions to determine the reconstruction of the corrupted input, as $\hat{x} = f(c(\tilde{x}))$. Then the parameters are updated in the direction of $\frac{\delta L(x,\hat{x})}{\delta\theta}$. As a result, the DAE tries to reconstruct x instead of $\tilde{x}$. After obtaining the reconstruction signal from DAEs, the signal-to-noise ratio (SNRs) value must be determined to assess signal quality.

The signal corruption process can be formulated as:

$$\hat{x} = \eta(x) \quad (3)$$

where $\eta : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is an arbitrary stochastic corrupting process that corrupts the input. The learning aim of the denoising job thus becomes:

$$f = argminE_y \left\| f(\hat{x}) - x \right\|^2 \quad (4)$$

This formulation goal is to identify a function f that best approximates η^{-1}. The signal denoising and restoration problems in a cohesive framework are tackled by selecting appropriate η in different contexts. The next section will discuss the ECA network and its working.

B. Efficient Channel Attention Network

According to cognitive research, people use an attention mechanism to preferentially concentrate on a subset of all information while ignoring other observable information. Convolutional neural networks (CNNs) performance has recently been demonstrated to be improved by a channel attention method. The squeeze-and-excitation network (SENet) [28] is one of these approaches, which captures channel attention for each convolutional block and achieves a noticeable performance boost for various CNN models. Wang et al. [29] realized that recording dependencies across all channels are wasteful and redundant. Therefore they avoided the dimensionality reduction step to reduce trainable parameters and presented an efficient channel attention network (ECANet) that establishes channel weights by conducting a rapid one-dimensional (1D) convolution of size k to capture cross-channel interactions quickly. ECA module working is as shown in Fig. 2 and discussed as follows:

By modelling the connection between the fused feature channels, the ECA structure is utilized to learn the weights of the features under several local channels. First, for the nth channel, a global average pooling is used to reduce the feature maps size in single values of size equal to the number of channels in the convolutional layer as follows:

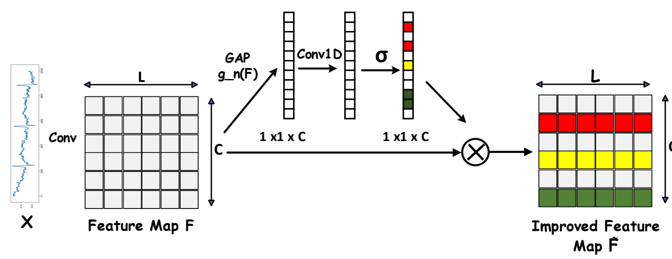


Fig. 2: Schematic of ECA module

$$g_n(F) = \frac{1}{L}\sum_{i=1}^{L} f_i \quad (5)$$

Where L is the length of the feature sequence F, following the global average pooling process. Features of $g_n(F)$ has certain periodicity and correlation which will be learned by fast 1D CNN with a kernel size of K. Fast 1D CNN is run on $g_n(F)$ to learn the feature weights under various channels, as shown below:

$$\omega = \sigma(Conv_k^{1d}(g_n(F))) \quad (6)$$

where, $Conv^{1d}$ denotes a 1D convolution, and in order to compute the kernel size K with the channel dimension C is given as:

$$k = |\frac{log2(C)}{2} + \frac{1}{2}|_{odd} \quad (7)$$

Where C is non-linearly correlated with K and the sigmoid function is denoted by σ. σ is used to compute the activation value of convolutional output to get new weights ω, which will reflect the local relationship and essential degree of the feature channel. New weights are now multiplied with each feature of the set F, implying that relevant features will be given higher weights to be improved, while less important features will be given lower weights, resulting in their suppression. Thus, an improved feature map with more relevant information is obtained. As a result, the ECA module is added after every *trans_conv* block in the decoder of our proposed network so that no extraneous data from the preceding block's feature maps are used in the reconstruction of the denoised signal. The architecture of the proposed model is described in the following section.

C. Proposed Model Architecture and Learning

In this study, we proposed a novel attention based convolutional denoising autoencoder (ACDAE) as shown in Fig. 3.

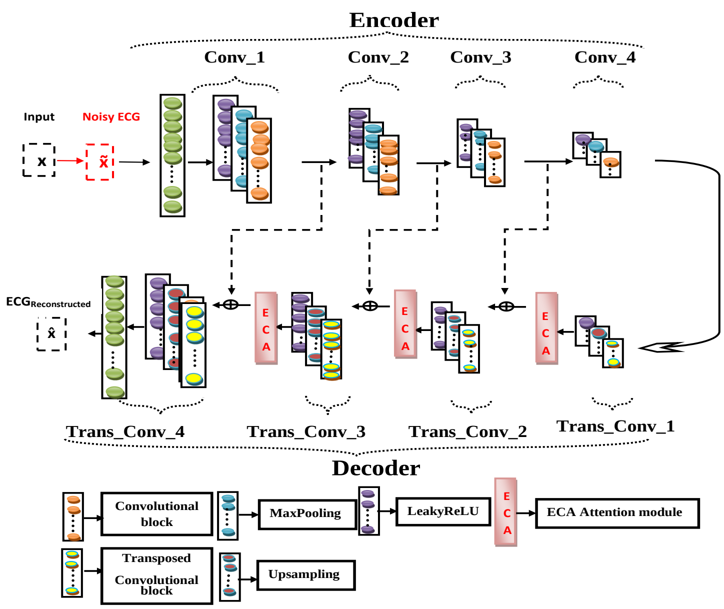


Fig. 3: Schematic of proposed Attention Based Convolutional Denoising Autoencoder

It comprises of two modules: an encoder and a decoder. The encoder module has four 1D convolutional layers, while the decoder module has four symmetric 1D transposed convolutional (*trans_conv*) layers. As input, the network receives a noisy ECG signal ($\tilde{x_i}$) noised with AWGN and outputs a denoised ECG signal($\hat{x_i}$). Initially, the feature maps are extracted using convolutional layers (Conv-layers) and downsampled using the Max-pooling layer of two. The noise gets suppressed during this encoding task while retaining the underlying structure. Next, the transposed convolutional layers *trans_conv* and upsampling layer decode the compressed ECG abstraction. The Conv-layers and the *trans_conv* layers are symmetrically connected via skip connections, as shown by dash lines. The role of the skip connections is two-fold. First, they help in back-propagating the gradients to the bottom layers and also help in recovering the ECG signal detail lost during the encoding and decoding process. ECA modules are connected after *trans_conv_1* to *trans_conv_3* as shown by the red colour box; they efficiently update the features retrieved via cross-channel interaction, allowing the network to pay more attention to the features of relevant information between the channels. The output $\hat{x_i}$ is then reconstructed through *trans_conv* and ECA module. The target of the training is to minimize the mean square value given by Eq. 11 between the $\hat{x_i}$ and the input x_i. The smaller the loss function value, the more likely the output $\hat{x_i}$ will reconstruct the input x_i. We experimented by placing the ECA module before Conv-2 and Conv-3 of the encoder, but it hardly improved the performance. We also explored the possibility of putting dropout layers. However, we found that the ECA module does not require dropouts as ECA helps get the attention from relevant channels, leading to solving the problem of overfitting. Therefore, ECA modules are only placed in the decoder path to reduce the computational time and resources.

This study is carried out on the Keras library with Google Tensorflow 2.6 and Python 3.7. NVIDIA GeForce GTX 1070 graphics card with Cuda libraries, 16GB of RAM, and an Intel Core i7-8700 3.20GHz CPU are installed on the machine and segmented noisy ECG samples from channels 1 and 2 of 300 (1.2 s), each concatenated and given input to the network. On the other hand, the proposed approach can also be used with any reasonable segmented window size (e.g., 5s–30s). The normalized mean square error given in Eq. 11 is minimized during the network's training. Keras-tuner [30] random search library is used to find the optimal hyperparameters of the neural network. The Adam algorithm, which has a learning rate of 0.0001, was chosen for optimization. A batch size of 32 is used, and epochs are set to 50 with Early-stopping callbacks, which is used to avoid overfitting. The network's details are provided in Table. I

TABLE I: Detailed Parameters of the ACDAE

Layer	Kernel Size	Filter Size	Output	Activation
Conv_1	13 x 1	16	600 x 16	LeakyReLU
Conv_2	7 x 1	32	300 x 32	LeakyReLU
Conv_3	7 x 1	64	150 x 64	LeakyReLU
Conv_4	7 x 1	128	75 x 128	LeakyReLU
trans_conv_1	7 x 1	128	75 x 128	LeakyReLU
ECA Module	3 x 1		75 x 128	Sigmoid
Addition			75 x 128	
trans_conv_1	7 x 1	64	150 x 64	LeakyReLU
ECA Module	3 x 1		150 x 64	Sigmoid
Addition			150 x 64	
trans_conv_1	7 x 1	32	300 x 32	LeakyReLU
ECA Module	3 x 1		300 x 32	Sigmoid
Addition			300 x 32	
trans_conv_1	13 x 1	16	600 x 16	LeakyReLU
Dense Layer		1	600 x 1	

IV. Experimental Results and Analysis

A. Evaluation Metrics

Denoising performance of ACDAE model for ECG signals is measured using three metrics: mean squared error (MSE), signal-to-noise ratio (SNR) and percentage-root-mean-square difference (PRD), given as follows:

$$SNR_{in} = 10 \log_{10} \frac{\sum_{i=1}^{n} x_i)^2}{\sum_{i=1}^{n} (x_i - \tilde{x_i})^2} \quad (8)$$

$$SNR_{out} = 10 \log_{10} \frac{\sum_{i=1}^{n} x_i)^2}{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \quad (9)$$

$$SNR_{imp} = SNR_{out} - SNR_{in} \quad (10)$$

$$RMSE_{out} = \sqrt{\frac{1}{N} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \quad (11)$$

$$PRD = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{\sum_{i=1}^{n} \hat{x}_i)^2}} * 100 \quad (12)$$

Where x_i is a sample of the original signal, $\hat{x_i}$ is sample the denoised signal and $\tilde{x_i}$ is the sample of noise-induced ECG. Mean squared error is used during pre-training since it serves as a loss function for weight update. On the other side, the signal to noise ratio was utilized to compare different denoising algorithms.

To evaluate the performance, performance measures like precision, recall, F1-score and accuracy is calculated. Formulae for calculating performance measures are given as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (13)$$

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

Where TP is True Positive, TN is True Negative, FP is False Positive, FN is False Negative.

B. Denoising performance of the proposed model

Two experiments are performed to evaluate the denoising performance of the proposed model. In experiment one, the segmented window of AFDB and NSRDB are used as described in section II. These databases are added with AWGN noise of different SNRs. In experiment two BIHA database is noised with AWGN noise and NST database noise. The details are described in the following subsections.

1) Denoising performance using additive white Gaussian noise: In this experiment, $1.2s$ segmented two-channel ECG signals from the AFDB and NSRDB databases are used for the experiment. Each window contains $250 \times 1.2 = 300$ samples, where 250 Hz is the signal's sampling frequency, two channels are concatenated and fed to the model for training. The AWGN noise is added, as discussed in section II. The AWGN stipulates the same amount of energy across all spectral bands. Denoising based on AWGN demonstrates the model's robustness when facing random perturbations. Denoising potential diminishes as the SNR of a system drops. ECG signals are corrupted using AWGN of varying standard deviations. We observed that the signal almost lost all morphological information with SNR of -15 dB and below as shown in Fig. 4 and Fig. 5. A random window from the test set is chosen to visualize the results of the denoised signal from 0 dB to -15 dB. Here, the top figure is the original signal chosen randomly, in the figure blue color shows the channel 1 signal and red color shows the channel 2 signal. Subsequent plots

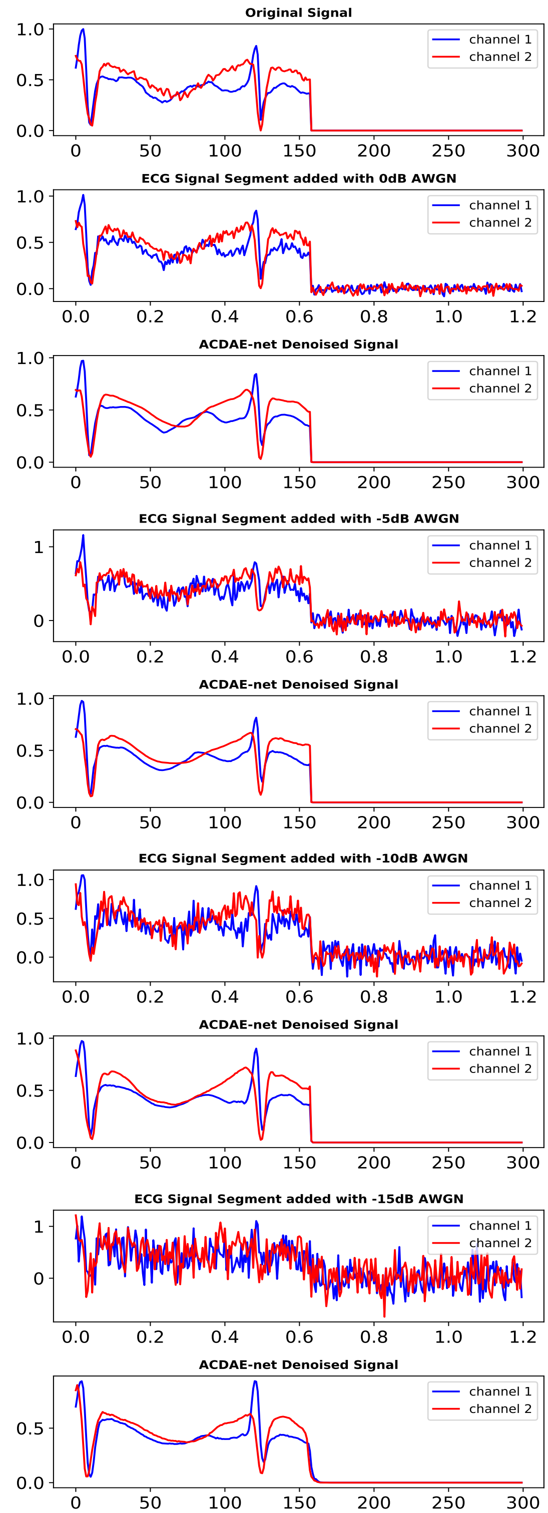


Fig. 4: Denoising Performance of ACDAE with different AWGN added to ECG signal

show the noised signal and their denoised output from the model ranging from 0 dB to -15 dB. As depicted, the model achieves a good denoising result till -15 dB, as it was able to reconstruct the morphology of the signal from the significant noise-induced. However, for -20 dB, the model is not able to reconstruct the morphology of the signal from channel 1 as indicated by blue color, whereas the morphology of the signal from channel 2 is reconstructed accurately, as shown in Fig. 5. It indicates that there is a need for increased training epochs, but as model training is using early stopping callback with a patience value of 5 therefore, it could be concluded that the proposed model's threshold had reached. As a result, additional tuning is necessary to reconstruct signals at and beyond -15 dB AWGN SNR. As previously stated, the largest range of noise that can disturb ECG is -6 dB on a wearable device, for which our model performs well. Therefore, we chose the range for our proposed model in which it will give great results are -15 dB to 20 dB SNR. Our assertion is also supported by a quantitative comparison in terms of SNR improvement and PRD, as shown in Table. II.

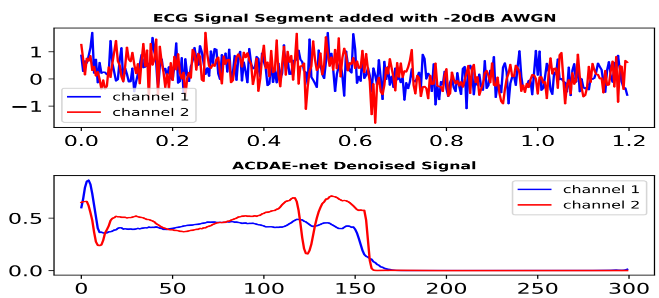
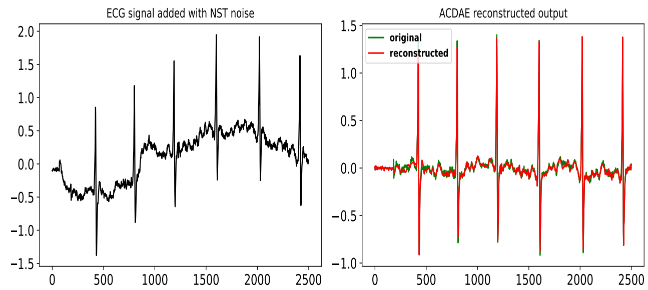


Fig. 5: Denoising Performance of ACDAE at -20dB AWGN

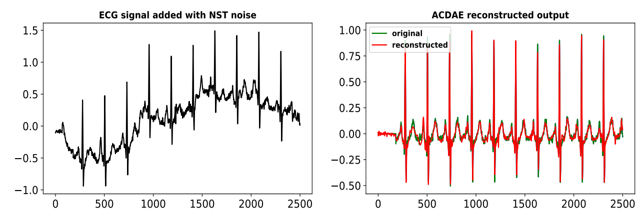
2) Denoising performance on NST database: This experiment is performed using the BIHA and NST databases, as explained in section II. For this experiment, the signal window of 10 sec is taken, and the signal is re-sampled at 250 Hz before giving it to the model. A sample size of $10 \times 250 = 2500\ samples$ are fed to the model as input. The prepared data is then added with 'em', 'bw', 'ma' noises in the same proportion for training and testing. The record number used for test set are 103, 107, 116, 117, 119, 124, 205, 207, 212, 220, 228, 233. The model performance for a random window with -6 dB and 0 dB SNR can be seen in Fig. 6. Here, the first window represents noised signal added with NSTDB noise, and the second window represents the reconstructed signal in red and the original signal in green. Individual SNR_{imp} s defined in Eq. 10 for the test records is shown in Fig. 9. Moreover, the BIHA database is also tested with AWGN noise as presented in the previous section and results obtained are shown quantitatively in Table. II. Individual SNR_{imp} for the test records is shown in Fig. 8. The plots are between records (on the X-axis) and SNR_{imp} (on Y-axis).

C. Classification performance of Proposed Model

This study uses two classification modules for atrial fibrillation (AF) classification from the normal signal. As explained in section II, the experiments are performed on two-channel



(a) Filtered Signal noised with NSTDB with SNR 0 dB



(b) [Filtered Signal noised with NSTDB with SNR -6 dB

Fig. 6: ACDAE reconstruction performance on NST Database

ECG segment windows. A total of 60,000 samples are used with 30,000 AF beats and 30,000 non-AF beats. SNR values ranging from -20 dB to 20 dB with a step size of 5dB are used to train and test each module similar to that of the denoising task. Classification performance was evaluated using Eq. 13, 14, 15. For simplicity, classification module 1 is referred to as M1 and module 2 as M2 throughout the paper.

M1 uses the feature map that was got from the encoder of the model. These feature maps are a compressed representation of ECG signals. This experiment aims to see the classification performance of the compressed feature maps. The advantage of using compressed feature maps or signals is that they take less power while transferring wireless and take less memory. Both the advantages are very much required for modern cloud-based wearable devices. Therefore, ECA attention modules are used for the classification task. They can help get relevant information from the compressed feature maps as it has the GAP layer that will average each feature map and reduce its size. These will be passed through the 1D CNN layer with adaptive kernel size for local channel attention, suppressing the weak channel features and giving attention to only important features as explained in section III(B).

M2 uses FC layer followed by GAP layer as shown in Fig 1. This experiment aims to see the difference using the ECA module as the ECA module comes with some extra trainable parameters. Therefore, M2 employs the GAP and FC layers instead of the ECA module. M1 performed better than M2, showing the usefulness of the ECA module. M1 achieved precision, recall and accuracy of 99.25%, 99.50%, and 99.25%, respectively, followed by M2 with 97.50%, 97.50%, and 97.75%, respectively. Table III shows the classification performance at 10 dB added noise. We used a 10 dB added signal as it hardly adds any noise and can be compared with other published work. The experiment is performed for 9 SNR levels as explained in section II, and precision, accuracy

TABLE II: Comparison of ACDAE performance with AWGN on different Databases

Database	-10 dB			-5 dB			0 dB		
	SNR_{imp}	RMSE	PRD(%)	SNR_{imp}	RMSE	PRD(%)	SNR_{imp}	RMSE	PRD(%)
AFDB database	31.87	0.0587	21.1	27.4	0.0315	11	19.31	0.0134	6.90
BIHA Database	22.06	0.0612	31.72	18.57	0.0432	24.1	14.63	0.0231	9.8

and recall for M1 and M2 are shown in Fig. 7 where the top figure shows the results for AF class and the lower figure shows the result of normal class. Here, we can observe that the performance of classification models till -10 dB is almost the same as shown in Table III and for -15 dB, overall accuracy for M1 is reduced to 95.84% from 99.25 %, and similarly precision and recall also decreased to 92.51% and 97.62 %, respectively, and a similar trend can be seen for M2. At -20 dB, the performance for M1 and M2 drops drastically, as also can be seen from the figure.

TABLE III: Performance of two modules for Atrial Fibrillation classification

Models	A			N		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy
Module 1(M1)	99.50	99.50	99.0	99.00	99.50	99.50
Module 2(M2)	97.50	96.50	97.50	97.50	98.50	98.00

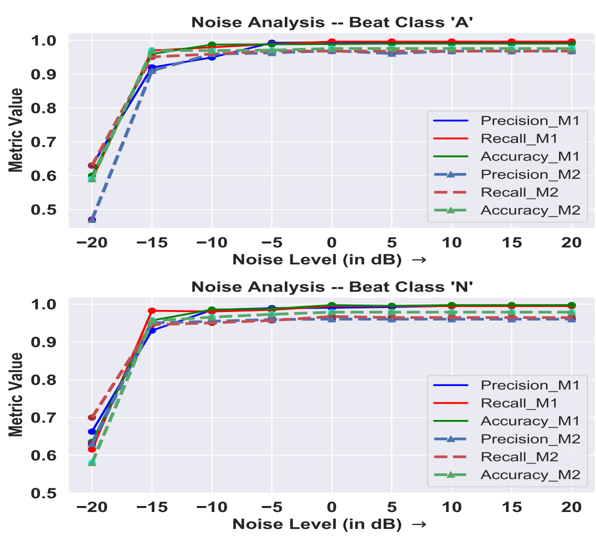


Fig. 7: Classification Performance of M1 and M2 with different AWGN added to ECG signal

V. Comparison of Results

A. Comparison of Denoising Performance

The experimental results have found that the proposed ACDAE model performed fairly well within the range of -15 dB to 20 dB. The performance is tested using Eq. 10, Eq. 11 and Eq. 12, with RMSE being used to quantify the variance between the ACDAE model output and the original signal. PRD calculates the total distortion in the denoised signal, and $SNRimp$ calculates the improvement in SNR between the denoised and input signal. A comprehensive review paper [31] is referred to for comparing the performance of the proposed model. Our model is solely compared to the best performance model reported in [31] and shown in Table IV. Additionally, our model performance on the record 103 is compared to a recent study [32]. The proposed work outperformed nearly all the work in the review paper, and some of the tests used a different database or different noises, which are outside of the focus of this paper. However, a recent work [16], in which the authors developed a two-stage denoising technique, is noteworthy since, after filtering ECG signals, they employed a morphology reconstruction network with five convolutional layers, making the model bulky and computationally expensive. The proposed strategy may assist in minimizing the size of the model while maintaining the same performance as proven in this work by using a very shallow model attention module with only one convolutional layer.

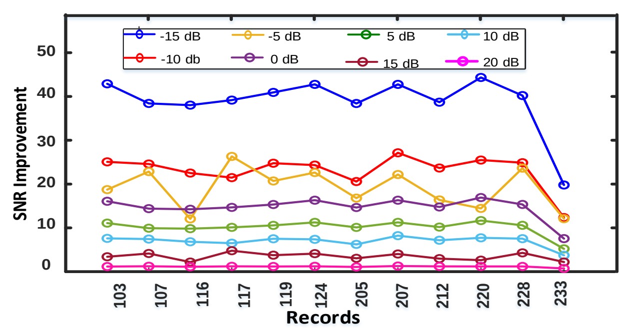


Fig. 8: SNR improvement with different AWGN added to BIHA Test records

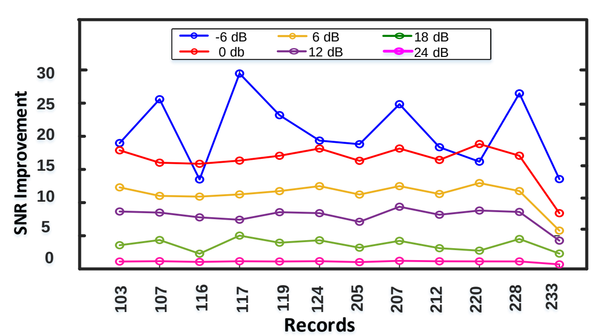


Fig. 9: SNR improvement with different NST noise added to BIHA Test records

TABLE IV: Comparison of ACDAE performance with different studies on BIHA Databases in presence of AWGN

Database	-5dB			0dB			5dB		
	SNR_{imp}	RMSE	PRD(%)	SNR_{imp}	RMSE	PRD(%)	SNR_{imp}	RMSE	PRD(%)
Record				103-BIHA					
Proposed ACDAE	23.73	0.0433	21.1	19.07	0.0235	11	13.08	0.0092	6.90
Successive local Filtering [32]	12	NR	NR	11.7	NR	NR	10.6	NR	NR
IHP-ST EMD [33]	7.84	NR	72.11	6.14	NR	49.32	5.89	NR	28.54
Record				Average of all records					
Proposed ACDAE	19.57	0.012	23.72	16.63	0.0332	14.1	14.10	0.0231	9.8
New MP-EKF [34]	15.00	NR	32.00	11.10	NR	28.00	8.00	NR	22.00

B. Comparison of Classification Performance

The performance of the proposed model achieved better results against many top-cited works. Our study looked at two classification approaches M1, which uses the attention mechanism (ECA layer) just before the fully connected (FC) layer for classification, and M2, which uses the most commonly used GAP layer before the FC layer, with the results indicating that the ECA layer helps to reduce false positives (FP). M1 receives a 1.75 % boost in precision, as well as a 2 % improvement in recall and 1.75 % improvement in accuracy. This work also utilized employed Grad-CAM [35], an explainable artificial intelligence technique to compare the two classification modules. This will help us to look at how the attention module is working, and we discovered that the ECA module significantly increased attention, as shown in Fig 10, the top heatmap shows M1 Grad-CAM, and the bottom heatmap shows M2 Grad-CAM for the same window. Here, red is indicting the highest attention and blue means least attention. In M1, we can observe that more attention is paid to the third QRS peak, and attention is increased at the first peak, and also it got slightly improved for the second peak. Therefore, we can confidently conclude that the proposed ECA module will help in classifying and denoising tasks in a neural network.

We also compared our findings to the top referenced recent papers using AFDB, which are listed in Table V. The authors of [36] reported 97.1 % precision and 97 % recall using stationary wavelet transform (SWT), feature extraction approach, and support vector machine (SVM) to classify AF. The authors of [37] transformed the ECG signal into a 2D image using SWT and employed a three-layer 2D-convolutional layer to classify AF with an accuracy of 98.63 %. The authors of [38] employed a combination of CNN and LSTM with focal loss to achieve 99.29 % precision, whereas [39] another work that also uses a combination of CNN and LSTM layer got 97.80 % of precision. With all of the research listed, our proposed model M1 achieves better results, demonstrating its resilience.

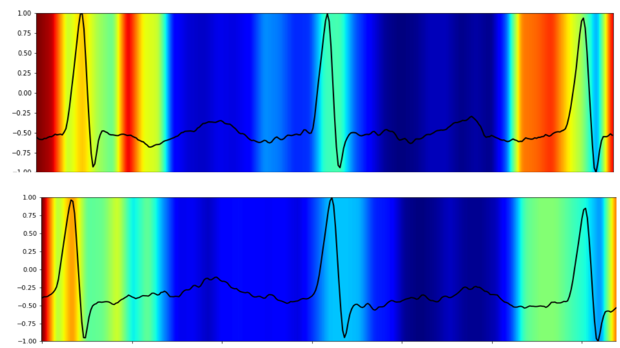


Fig. 10: Grad-CAM Heatmap plot for a AF window

TABLE V: Comparison of the performances of AF detection algorithms based on their viability on MIT-BIH AFDB database.

Algorithm	Precision	Recall	Accuracy
Wavelet Transform with SVM [36]	97.10	97.00	-
SWT+ 2D CNN [37]	97.87	98.79	98.63
CNN + LSTM [39]	97.80	98.98	97.80
CNN + LSTM [38]	99.29	97.87	-
Proposed Model	99.25	99.50	99.25

VI. Conclusion and Future Work

The proposed attention-based convolutional denoising autoencoder effectively denoises the low SNR ECG signal. It outperformed the atrial fibrillation classification results compared to the most cited recent works. The study is comprehensively evaluated using four publicly available databases. The study uses two classification modules to test the effectiveness of an efficient channel attention module that efficiently update the features retrieved via cross-channel interaction, allowing the network to pay more attention to the features of relevant information between the channels. The model with the ECA module is found better compared to State-of-the-art results. This study uses NST database noises and AWGN; however, real-time noise, particularly motion artifacts, may be different. As a result, the recommended strategy may not yield the same outcomes in such situations. In the future, we will refine our algorithm by running it through a slew of real-world training scenarios.

References

[1] J. Wu, C. P. Gale, M. Hall, T. B. Dondo, E. Metcalfe, G. Oliver, P. D. Batin, H. Hemingway, A. Timmis, and R. M. West, “Editor’s choice-impact of initial hospital diagnosis on mortality for acute myocardial infarction: A national cohort study,” *Eur. Heart J. Acute Cardiovasc. Care*, vol. 7, no. 2, pp. 139–148, 2018.

[2] S. Mendis, D. Bettcher, F. Branca *et al.*, “World health organization global status report on noncommunicable diseases. 2014,” 2014.

[3] M. Abdelazez, S. Rajan, and A. D. C. Chan, “Detection of atrial fibrillation in compressively sensed electrocardiogram measurements,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[4] P. Bera, R. Gupta, and J. Saha, “Preserving abnormal beat morphology in long-term ecg recording: An efficient hybrid compression approach,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2084–2092, 2020.

[5] R. J. Martis, U. R. Acharya, and L. C. Min, “Ecg beat classification using pca, lda, ica and discrete wavelet transform,” *Biomed. Signal Process. Control*, vol. 8, no. 5, pp. 437–448, 2013.

[6] S. Osowski and T. H. Linh, “Ecg beat recognition using fuzzy hybrid neural network,” *IEEE Trans. Biomed. Eng.*, vol. 48, no. 11, pp. 1265–1271, 2001.

[7] M. S. Chavan, R. Agarwala, and M. Uplane, “Suppression of baseline wander and power line interference in ecg using digital iir filter,” *International journal of circuits, systems and signal processing*, vol. 2, no. 2, pp. 356–365, 2008.

[8] P. Singh, K. Bhole, and A. Sharma, “Adaptive filtration techniques for impulsive noise removal from ecg,” in *2017 14th IEEE India Council International Conference (INDICON)*, 2017, pp. 1–4.

[9] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, “A wavelet-based ecg delineator: evaluation on standard databases,” *IEEE Trans. biomed. eng.*, vol. 51, no. 4, pp. 570–581, 2004.

[10] M. A. Kabir and C. Shahnaz, “Denoising of ecg signals based on noise reduction algorithms in emd and wavelet domains,” *Biomedical Signal Processing and Control*, vol. 7, no. 5, pp. 481–489, 2012.

[11] Z. Golrizkhatami and A. Acan, “Ecg classification using three-level fusion of different feature descriptors,” *Expert Syst. Appl.*, vol. 114, pp. 54–64, 2018.

[12] P. Xiong, H. Wang, M. Liu, F. Lin, Z. Hou, and X. Liu, “A stacked contractive denoising auto-encoder for ECG signal denoising,” *Physiological Measurement*, vol. 37, no. 12, pp. 2214–2230, nov 2016. [Online]. Available: https://doi.org/10.1088/0967-3334/37/12/2214

[13] P. Xiong, H. Wang, M. Liu, S. Zhou, Z. Hou, and X. Liu, “Ecg signal enhancement based on improved denoising auto-encoder,” *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 194–202, 2016.

[14] H.-T. Chiang, Y.-Y. Hsieh, S.-W. Fu, K.-H. Hung, Y. Tsao, and S.-Y. Chien, “Noise reduction in ecg signals using fully convolutional denoising autoencoders,” *IEEE Access*, vol. 7, pp. 60 806–60 813, 2019.

[15] F. P. Romero, D. C. Piñol, and C. R. Vázquez-Seisdedos, “Deepfilter: an ecg baseline wander removal filter using deep learning techniques,” *Biomedical Signal Processing and Control*, vol. 70, p. 102992, 2021.

[16] L. Qiu, W. Cai, M. Zhang, W. Zhu, and L. Wang, “Two-stage ecg signal denoising based on deep convolutional network,” *Physiological Measurement*, vol. 42, no. 11, p. 115002, 2021.

[17] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” *Adv. Neural Inf. Process. Syst*, vol. 29, 2016.

[18] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals,” *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[19] G. B. Moody, W. Muldrow, and R. G. Mark, “A noise stress test for arrhythmia detectors,” *Computers in cardiology*, vol. 11, no. 3, pp. 381–384, 1984.

[20] C. Carreiras, “BioSPPy: Biosignal processing in Python,” 2015–. [Online]. Available: https://github.com/PIA-Group/BioSPPy

[21] B. Yuen, X. Dong, and T. Lu, “Detecting noisy ecg qrs complexes using waveletcnn autoencoder and convlstm,” *IEEE Access*, vol. 8, pp. 143 802–143 817, 2020.

[22] B. Hou, J. Yang, P. Wang, and R. Yan, “Lstm-based auto-encoder model for ecg arrhythmias classification,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1232–1240, 2020.

[23] W. Mao, J. He, and M. J. Zuo, “Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1594–1608, 2020.

[24] B. Zhao, C. Cheng, Z. Peng, Q. He, and G. Meng, “Hybrid pre-training strategy for deep denoising neural networks and its application in machine fault diagnosis,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[25] W. Mao, J. Chen, X. Liang, and X. Zhang, “A new online detection approach for rolling bearing incipient fault via self-adaptive deep feature matching,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 443–456, 2020.

[26] J. Yu, X. Liu, and L. Ye, “Convolutional long short-term memory autoencoder-based feature learning for fault detection in industrial processes,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[28] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[29] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “Eca-net: Efficient channel attention for deep convolutional neural networks,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.01155

[30] T. O’Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, “Kerastuner,” https://github.com/keras-team/keras-tuner, 2019.

[31] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, “Review of noise removal techniques in ecg signals,” *IET Signal Processing*, vol. 14, no. 9, pp. 569–590, 2020.

[32] N. Mourad, “Ecg denoising based on successive local filtering,” *Biomedical Signal Processing and Control*, vol. 73, p. 103431, 2022.

[33] S. Samadi and M. B. Shamsollahi, “Ecg noise reduction using empirical mode decomposition based on combination of instantaneous half period and soft-thresholding,” in *2nd Middle East Conference on Biomedical Engineering*. IEEE, 2014, pp. 244–248.

[34] H. D. Hesar and M. Mohebbi, “An adaptive particle weighting strategy for ecg denoising using marginalized particle extended kalman filter: An evaluation in arrhythmia contexts,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1581–1592, 2017.

[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[36] S. Asgari, A. Mehrnia, and M. Moussavi, “Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine,” *Comput. in biol. Med.*, vol. 60, pp. 132–142, 2015.

[37] Y. Xia, N. Wulan, K. Wang, and H. Zhang, “Detecting atrial fibrillation by deep convolutional neural networks,” *Comput. in biol. Med.*, vol. 93, pp. 84–92, 2018.

[38] G. Petmezas, K. Haris, L. Stefanopoulos, V. Kilintzis, A. Tzavelis, J. A. Rogers, A. K. Katsaggelos, and N. Maglaveras, “Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets,” *Biomedical Signal Processing and Control*, vol. 63, p. 102194, 2021.

[39] R. S. Andersen, A. Peimankar, and S. Puthusserypady, “A deep learning approach for real-time detection of atrial fibrillation,” *Expert Syst. Appl.*, vol. 115, pp. 465–473, 2019.