Detection of Cardiac Electrical Instability Prior to Cardiac Arrest

Aneel Damaraju, Chiraag Kaushik, Andrew Pham, Kunal Rai, Tucker Reinhardt, Frank Yang

Mentors: Sebastian Acosta, PhD; Mubbasheer Ahmed, MD; Parag Jain, MD

1 Introduction

Congenital heart defects are the most common birth defects and one of the leading causes of infant death [9]. Infants affected by congenital heart defects must often spend time in intensive care units before or between palliations. While in intensive care units, these patients may experience unexpected cardiopulmonary arrests. These events are characterized by the breakdown of the heart's electrical function, leading to decreased flow of oxygenated blood and possible death [23]. In the event of cardiopulmonary arrest, prompt and decisive action by physicians is crucial for patient survival. Physiological abnormalities in patients often manifest prior to these life-threatening events and may be detected via electrocardiogram (ECG) monitoring [7, 24]. Early detection of these abnormalities offers the possibility of preventative treatment. Such proactive treatments are crucial in preventing the irreversible heart tissue damage that frequently occurs during cardiac arrest [13]. Early detection of cardiac instability is challenging because these abnormalities can often be imperceptible to the human eye, thus making them undetectable by caregivers. In this report, we develop an algorithm for early detection of electrical instability in the heart. Here, the term "electrical instability" broadly refers to changes in ECG morphology which may indicate imminent cardiac arrests.

1.1 Background

We focus on infants born with hypoplastic left heart syndrome (HLHS), a condition in which the left side of the heart is underdeveloped. Figure 1 depicts a comparison between a healthy heart and an HLHS heart. Annually, 1,025 babies are born with HLHS in the United States [17]. While HLHS accounts for only 2-3% of all congenital heart defect cases, it is extremely fatal; left untreated, it leads to death in 95% of cases within a few weeks from birth [2]. In a healthy baby, the right side of the heart pumps oxygen-poor blood to the lungs, where it is reoxygenated. The left side of the heart then pumps the re-oxygenated blood to the body. In the case of HLHS, through the Norwood procedure, the patient's major vessels are re-structured to let the right side of the heart pump partially oxygenated blood in parallel to the lungs and the rest of the body [19]. This increase in myocardial workload can result in cardiac ischemia, a condition in which cardiac tissue is not properly supplied with oxygenated blood. Cardiac ischemia results in instability of the heart's electrical conduction and eventual heart failure [18, 27]. Cardiac ischemia has been shown to create electrical instability that is manifested in the electrocardiogram (ECG) and can be observed in various components of the ECG waveform [6, 8]. Figure 2 depicts a sketch of a 1-lead ECG waveform, with various components labeled. Because the level of underdevelopment of heart tissue differs between individual HLHS patients, ECGs recorded from these patients vary greatly. As a result, detection of cardiac instability via conventional bedside monitoring is severely limited by ambiguities between healthy and ischemic states [8]. Further, any robust automated detection of cardiac instability must,

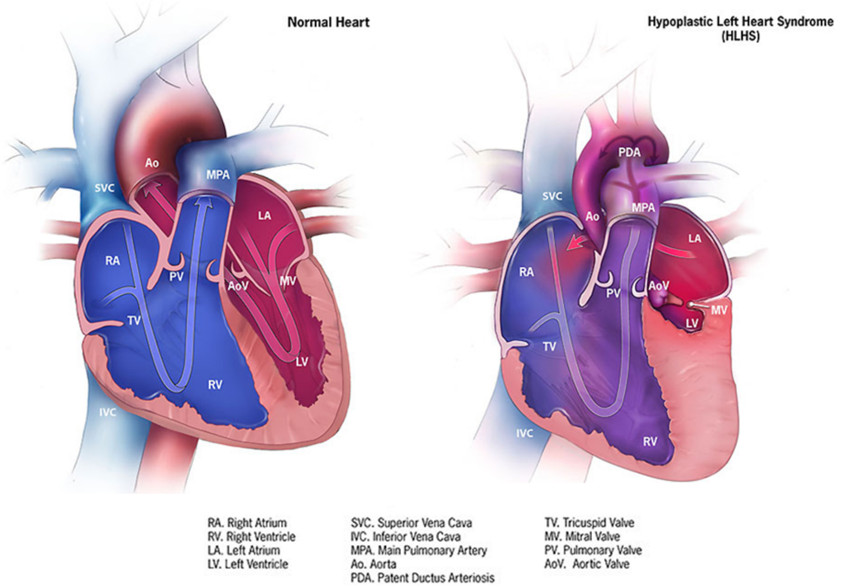


Figure 1: Comparison of normal heart (left) and heart with hypoplastic left heart syndrome (right). Infants with hypoplastic left heart syndrome need to undergo surgery and are at high risk of cardiac arrest[1].

therefore, be non-patient-specific. The challenges in detecting abnormalities in ECG signals for post-operation HLHS patients prior to cardiac arrest are summarized as follows:

I. Subtle abnormalities in ECG signals prior to cardiac arrest are often imperceptible to humans.
II. Even if ECG abnormalities can be observed, continuous bedside monitoring of ECG for several patients cannot be performed by caregivers concurrently.
III. Due to large inter-patient variability in ECG signals, development of non-patient-specific algorithms and methods are not suitable.

1.2 Prior Work

Cardiac arrhythmia detection using the ECG generally consists of 3 main steps: (1) signal pre-processing, (2) feature extraction, and (3) modeling. ECG signal pre-processing involves filtering techniques for noise reduction and R-peak detection (see Figure 1) to segment ECG signals into individual heartbeats [6]. ECG pre-processing has been explored extensively in prior works [16]. Recent works on arrhythmia detection have focused more heavily on feature extraction and modeling from single heartbeats [14]. There are two common approaches for heartbeat feature extraction: (i) extraction of morphological features and (ii) derived features. Morphological features are created through the observation of the shape of signals and require

[1] https://www.cdc.gov/ncbddd/heartdefects/hlhs.html#:~:text=Hypoplastic%20left%20heart%20syndrome%20(HLHS,type%20of%20congenital%20heart%20defect.

precise identification of the parts of a heartbeat [14]. Time intervals between Q and S waves [5], slope velocity of segments [10], and valley or peak values [26], are all examples of morphological features. Derived features are typically obtained via dimensionality reduction techniques, and include vector cardiograms [15], discrete wavelet transforms [15], and PCA components [29]. Classical approaches for heartbeat anomaly detection involve applying variability metrics to ECG signal data. In a recent work, wavelet entropy was used to predict termination of atrial fibrillation (AF) as well as electrical cardioversion in AF patients [1]. Shannon entropy has also been applied for classification of heart rate variability for patients suffering from coronary artery disease [3]. Variability metrics applied to ECG signals often focus on the heart rate to predict cardiac event onset rather than the ECG morphology. In recent years, deep learning models have become popular for feature extraction-based modeling in heartbeat anomaly detection. Stacked long short-term memory (LSTM) networks have been used in place of feature extraction to determine thresholds for irregular heartbeats [4]. These LSTM networks are able to identify different types of anomalies. LSTM networks have also been implemented for continuous classification of heartbeats in online settings due to low computation overhead on personal wearable devices [25].

Autoencoders are another neural network based approach for heartbeat anomaly detection. For example, Putra et al. [22] used autoencoder network structures for feature extraction for heartbeat classification. Autoencoder networks are often used for dimensionality reduction, as they tend to have higher information retention than classical methods. Jensen et al. [11] used sparsely-connected recurrent neural network autoencoders for time series analysis in an ensemble fashion, which was the first implementation of this ensemble network for unsupervised outlier detection. The proposed autoencoder ensembles outperformed state-of-the-art methods for outlier detection [11]. The use of convolutional denoising autoencoders (CDAE) have also been proposed for arrhythmia detection using 2-lead ECG signals [20].

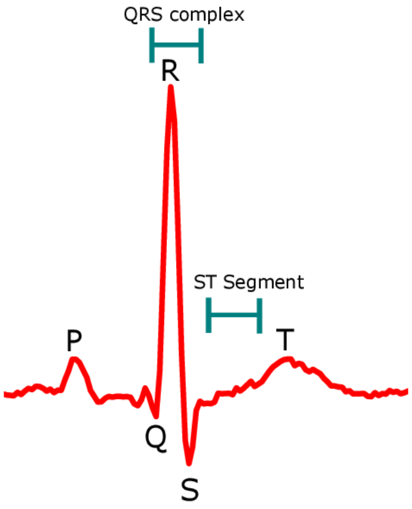


Figure 2: Sketch of 1-lead ECG waveform, with components labelled.

1.3 Our Contribution

This project builds on recent developments in anomaly detection for ECG data. Electrocardiograms from HLHS patients are more unstable than those of typical patients [8]. This makes morphological feature extraction challenging, as peaks and segments are irregular and difficult to identify. Therefore, we focus on using a series of heartbeats to quantify cardiac instability, rather than using morphological feature engineering or heart rate alone. We develop an autoencoder-based algorithm and change-point detection scheme for early detection of cardiac events.

1.4 Objectives

1. Explore and visualize HLHS patient ECG data for patient-specific and temporal trends.
2. Develop an algorithm which takes as an input a time-window of multi-lead ECG data and outputs a metric describing the instability of the heart's electrical activity.
    a. Use provided control group data to refine the above algorithm, improving robustness and reducing false positive rates.
3. Quantify and assess the effectiveness of the above algorithm in sounding an alarm prior to cardiac arrest events.
4. Create a dashboard to present the instability metric, current ECG morphology, and patient data to the attending medical provider.

2 Description of Data

The dataset used in this project was obtained from the Sickbay database at Texas Children's Hospital (TCH). Before use, all data was de-identified to minimize the risk of exposure to health information which is protected by the HIPAA. The specific physiological time series chosen for the project comes from 139 pediatric patients in the intensive care unit at TCH. 44 of these patients experience a cardiac arrest at the end of the time series, and 95 do not. Each patient was born with hypoplastic left heart syndrome and was operated on shortly after birth. The data provided for each patient are as follows:

- The **electrocardiogram (ECG)** signal measured at each of 4 different leads (electrodes placed on the patient's body). For each patient, six hours of ECG data sampled at 240Hz is provided. The provided data is not labeled, but the end time of each ECG time series (immediately after the 6 hours) corresponds to the onset of a cardiac arrest.
- The **heart rate** measured over the same six-hour period as the above ECG data, in beats per minute. The data was recorded by a heart rate monitor at a sampling rate of 2 Hz.

However, before being given to our team, the heart rate data was resampled using linear interpolation to match the 240 Hz sampling rate of the ECG signals. It is important to note that this data was not available for all of the patients; this missing data is discussed and characterized below.

- A signal of **timestamps** (in seconds) which is aligned with the 240 Hz sampling rate of the ECG signals and the heart rate.

In total, our dataset consists of over 834 hours of ECG data at each of the 4 leads (for a total of over 3,336 hours of ECG data), and heart rate information corresponding to this entire duration.

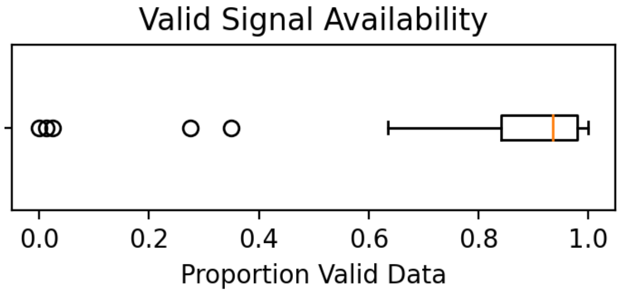


Figure 3: Proportion of original signal remaining for each cardiac arrest patient after filtering out flatline sections

There is no missing data in the provided ECG data for any patient, but there are some sections which display a flat signal, indicating that the leads were disconnected from the skin during measurement. Such flatline segments are removed in the data cleaning section of our pipeline, as described in Section 4.2 . The proportion of valid ECG data available for each cardiac arrest patient is summarized in Figure 3.

3 Data Science Pipeline

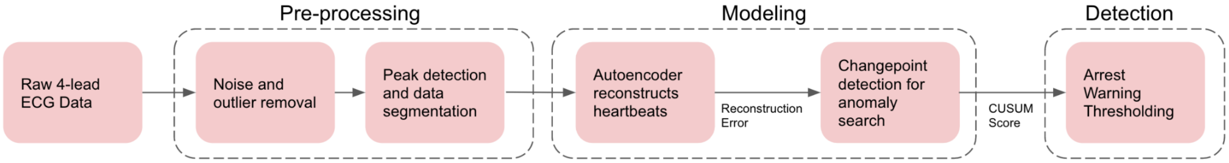

Pre-processing
Modeling
Detection
Raw 4-lead ECG Data
Noise and outlier removal
Peak detection and data segmentation
Autoencoder reconstructs heartbeats
Reconstruction Error
Changepoint detection for anomaly search
CUSUM Score
Arrest Warning Thresholding

Figure 4: Data Science Pipeline

Below, we present a brief overview of the various components of our project (which are visually summarized in Figure 4). Subsequent sections will discuss these components in more detail.

3.1 Data Wrangling

The first major section of our pipeline consists of three main parts - removing “flatline” sections of the ECG signal (which are likely due to lead disconnects), filtering out low and high frequency noise, and structuring the data into the format we input to our model. After initial data cleaning, ECG signals are separated into individual heartbeats using R-peaks as delimiters (see Figure 2). We use groups of 10 heartbeats as data for our feature vectors. The final step to format the feature vectors is to fix their dimension, which we do by linearly interpolating each 10 heartbeat section to dimension 1000.

3.2 Data Exploration

We visualize the pre-processed ECG data in order to better understand temporal, spectral, and non-linear trends and motivate subsequent steps in the pipeline. We first consider spectral characteristics of the ECG signal, looking for patterns in time-frequency response and spectral entropy. Furthermore, visualizations of the data in a low-dimensional space (2 or 3 dimensions) are used to inform and motivate the modeling phase of the pipeline.

3.3 Modeling

The modeling phase consists of developing metrics which capture the variability or complexity of the ECG over time. Early ECG data in the provided time window (i.e. >4 hours before the cardiac arrest) is treated as a nominal “healthy” baseline. We introduce a convolutional denoising autoencoder architecture which is first trained on healthy heartbeats and then evaluated on all ECG data for each patient. We then show that autoencoder reconstruction error and change-point detection methods can be used for interpretable warning metrics. Further, we utilize control patient data to improve the robustness of these techniques and reduce false positive rates. Specifically, we improve parameter selection for the above modeling scheme, incorporate time-delayed transfer learning to update our model over time, and explore the use of LSTM autoencoder networks in accounting for natural heartbeat drift over time. We also compare the use of mean-squared error and Kullback-Leibler (KL) divergence metrics in quantifying change in our high-dimensional setting.

3.4 Validation and Results

The validation phase consists of assessing the metrics devised during modeling and testing our overall algorithm on the entire patient cohort. Details on data splitting in section 6.1.We quantify the test accuracy of these models by setting an empirical threshold (to identify cardiac instability) on our change-point detection metric. The proposed architectures are compared and assessed based on the detector’s performance on patients who experienced cardiac arrest and those who did not (via visualization using ROC curves and explicit computation of the warning metric over time).

3.5 User Application

In parallel with the above data science pipeline, we create a web-based dashboard that allows medical professionals to visualize and interact with the results of the above data science pipeline. The proposed dashboard provides a cloud-based, centralized interface for viewing the pre-processed ECG waveforms and for viewing recent trends in the warning metric developed in the modeling phase.

4 Data Wrangling

4.1 Peak Detection

One of the core pre-processing tasks is separating the raw ECG signal into individual heartbeats. Every heartbeat has an R-peak, a large spike in the middle of the heartbeat complex, which is an easily detectable delimiter (Figure 2). We define a heartbeat as the data contained within a given peak-to-peak interval.

To account for multi-lead ECG data, the absolute value of each lead is taken, the leads are summed, and the resulting signal normalized between zero and one. We then detect the R-peaks on the positive sum signal based on topographic prominence (demonstrated in Figure 5), yielding a set of heartbeat delimiters common across all leads.

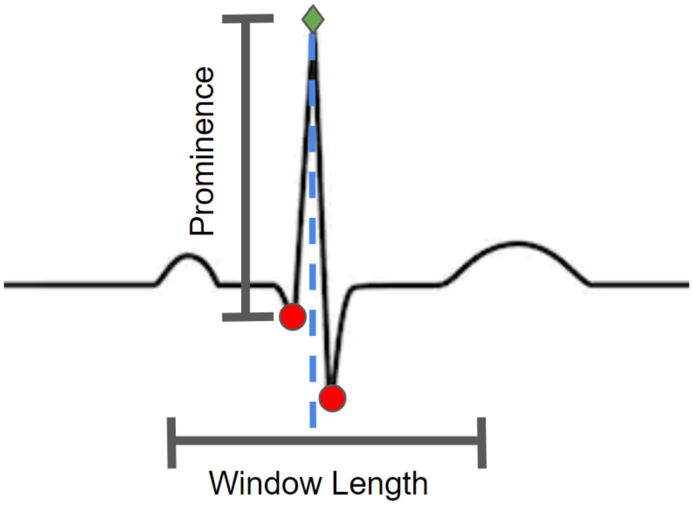


Figure 5: Candidate peaks are evaluated by finding the local minima to the left and right within a window length. For the candidate to be counted as a peak it must meet a threshold *prominence* value above the greater of the minima.

4.2 Outlier Removal and Disconnected Lead Detection

While the peak detection method described above works well for splitting raw data into individual “heartbeats”, it makes the assumption that data between two distinct R-peaks constitute a single valid heartbeat. However, this is not always the case due to lead disconnections, which result in flatlines. This is shown in Figure 6. Our algorithm detects these errors by iterating through the detected heartbeat delimiters and checking the percentage of unique values between each delimiter, removing sections with less than 5% unique values. Using this algorithm, 41 of the 44 cardiac arrest patients and 52 of the 95 control patients were considered valid based on the available data.

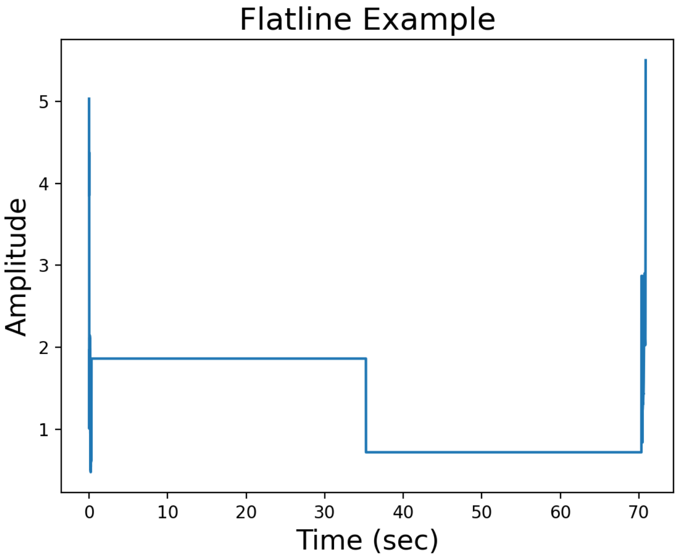


Figure 6: Detected “heartbeat” which does not contain a valid heartbeat, but rather is likely to be a lead disconnection.

4.3 Filtering

High and low frequency noise is removed from ECG signals using a zero-phase bandpass filter. For this purpose, a 6-pole Butterworth filter is applied using forward-backward filtering, using cutoff frequencies of 0.7Hz and 50Hz. Cutoff frequencies were obtained from literature (Clifford et al. 2006). The filter transfer function is shown in Figure 7A. Unfiltered and filtered ECG spectra for the first lead of a representative patient is shown in Figure 7B.

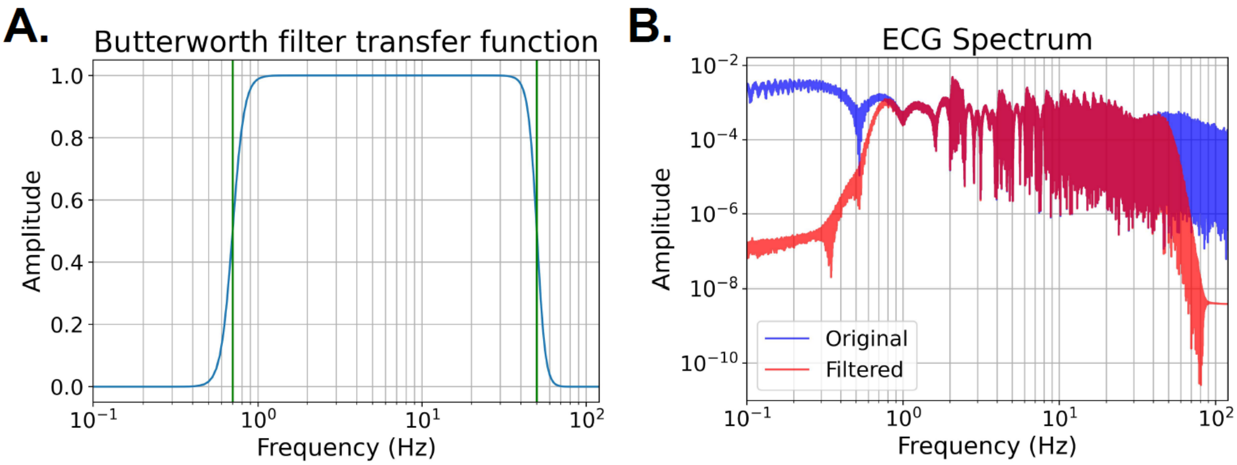


Figure 7: Noise removal (A) bandpass Butterworth filter transfer function with green vertical lines denoting cutoff frequencies, (B) unfiltered and filtered ECG spectra for a representative patient.

4.4 Interpolation and Segmentation

After raw ECG data is partitioned into individual heartbeats, consecutive heartbeats are combined into groups of 10. Groups of heartbeats, as opposed to individual heartbeats, are selected as feature vectors. This gives our model greater robustness against errors in peak detection (i.e. missed peaks). Furthermore, heartbeats generally have different durations (i.e. the heart rate is not constant), so each 10-heartbeat signal has a different length. To obtain fixed-length feature vectors, each 10-heartbeat slice is linearly interpolated to dimension 1000, roughly corresponding to 100 data points per heartbeat. Each 10-heartbeat slice is then normalized to have mean 0 (removing DC offset) and standard deviation 1. Using four-lead data, this process yields a normalized 2-D feature vector of dimension (1000, 4).

5 Data Exploration

5.1 Spectral Characteristics

For initial data exploration, we compute the spectrogram of filtered ECG signals. Here, we test a simple classical hypothesis: the ECG spectrum should show visible changes as the signal becomes more unstable. However, computed spectrograms over patients and leads did not display this trend. A spectrogram of a representative patient is given in Figure 8A, showing no significant qualitative changes as the patient nears a cardiac arrest at time 0. The spectral entropy vs. time is plotted in Figure 8B, and similarly shows no significant qualitative increase near time 0.

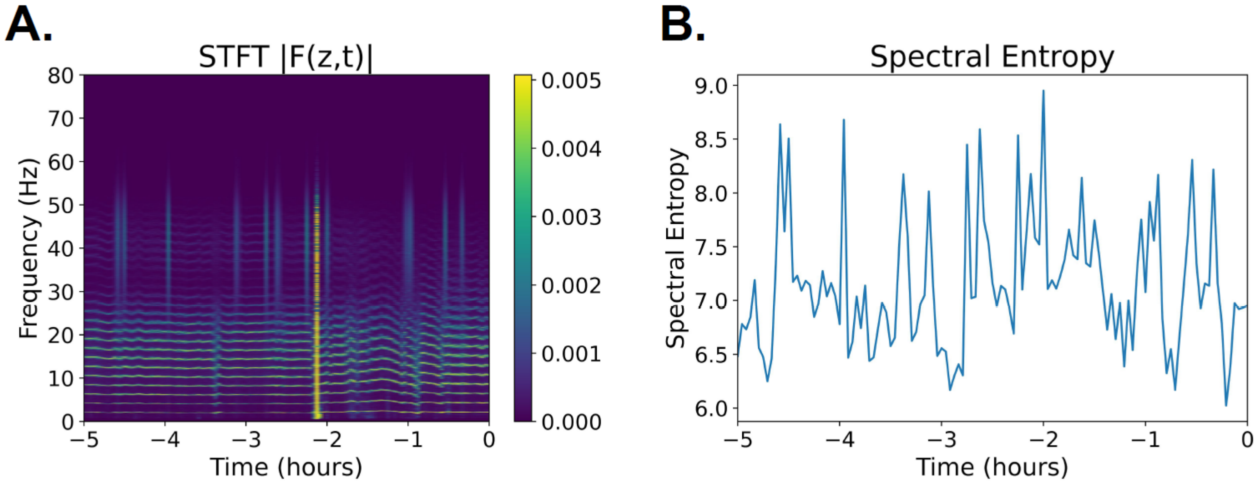


Figure 8: Spectral analysis (A) Spectrogram of a representative ECG signal, (B) spectral entropy of ECG signal spectrogram.

5.2 Dimension Reduction

We perform data visualization via dimensionality reduction. ECG data reduced to a two-dimensional space can be visualized using scatterplots, yielding some qualitative temporal information. Two techniques we use for dimensionality reduction are Principal Component Analysis (PCA), a linear technique, and Uniform Manifold Approximation and Projection (UMAP), a nonlinear technique. We compare these two methods and the trends that we can interpret from their low dimensional projections.

5.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimension reduction technique that attempts to retain most of the variance in the data when projecting to a lower dimensional space. We performed the PCA for each patient and found that the principal directions corresponding to 10% of the total number of dimensions was sufficient to account for over 95% of the healthy heartbeat variance. This partially motivates the 1:10 compression ratio selected in the modeling stage. We do not directly use PCA for modeling because data visualization suggests that linearly reduced data is not normally distributed. Additionally, we seek to circumvent the orthogonality restriction of the principle directions.

5.2.2 Manifold Learning with UMAP

Nonlinear dimension reduction often outperforms linear techniques such as PCA. PCA computes a lower dimensional hyperplane to capture variation in a dataset. However, by utilizing nonlinear relationships in the dataset, nonlinear manifolds can be explored, allowing for tighter compression and more descriptive latent representations. Figure 9 shows the projection of the data using a common manifold learning algorithm, UMAP. An interesting conclusion from UMAP is that as the samples increase in time, i.e. get closer to the cardiac arrest, they tend to move away from the cluster of points representing stable heart rates. This provides promising insight for the modeling stage of our project, as we can see that UMAP is able to find meaningful differences in the stable and unstable heartbeats.

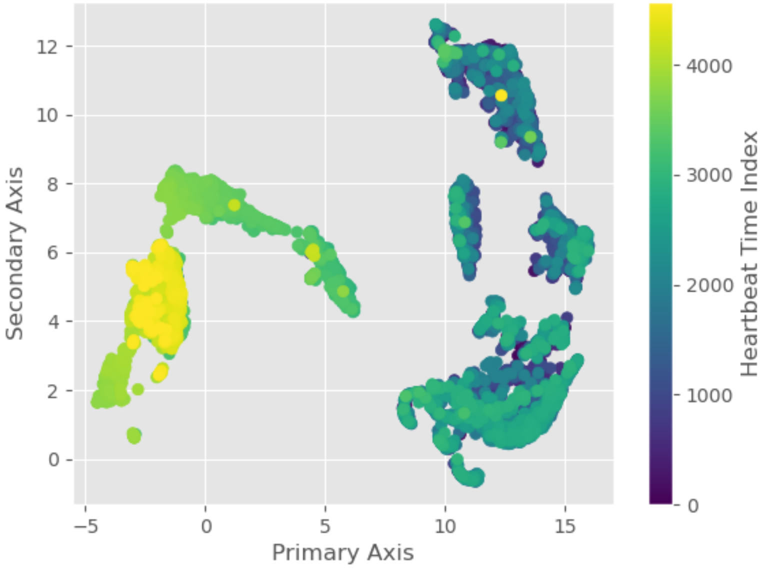


Figure 9: Two dimensional UMAP projection of heartbeats for a representative cardiac arrest test patient. Each feature vector is assigned an index, corresponding to time evolution (higher indices are closer to the cardiac arrest).

6 Modeling

6.1 Data Splitting

Since the data for this project is unlabeled, assumptions were made by the sponsor and team for how to split the data for modeling. The first third of each patient's data is assumed to be normal heartbeats while the last two-thirds are used for testing. We hypothesize that the data near the onset of cardiac arrest will be detected as abnormal. For tuning the models, the first third of the patient’s data is randomly split into a training and validation set at 85% train and 15% validation. This validation split was chosen for hyperparameter tuning of models and to verify that the models do not overfit on the training data. Once hyperparameters were chosen, all of the normal heartbeats (first two hours) were used to train the final model. This final model was then evaluated on the last two-thirds of the data. It is to be noted that this data splitting scheme is slightly modified for the transfer learning model developed in a later section of this report, as described in Section 6.2.3.

While we split the data for each patient into healthy and unhealthy heartbeats we also divide the patient set itself into three categories. Every patient is chosen to be a representative of the training set, validation set or the testing set. This designation is chosen randomly for each patient with approximately 70% of our patients in the training set, 20% in the validation set and 10% in the testing set. We run our initial models on the training set, and use these results to help narrow down which models are the most appropriate for anomaly detection in this specific application. Once we have selected our candidate models, we tune the hyperparameters of that model on the training set, and confirm the effectiveness of a parameter selection on the validation set. Finally, once we have selected both a model and a set of parameters to effectively detect cardiac instability we run this finalized model on the testing set, to ensure that we did not happen to overfit our architecture or model parameters to the training or validation patients.

6.2 Learning Normal Heartbeats with Autoencoders

6.2.1 Introduction to Autoencoders

The efficacy of nonlinear dimension reduction motivates our use of autoencoders in the modeling phase. Autoencoders [12] use neural networks to discover a low-dimensional representation of data, and are a form of unsupervised learning. The defining quality of an autoencoder is a bottleneck in a neural network architecture. The set of neural networks prior to the bottleneck are called the encoder, and those after constitute the decoder. This is shown in Figure 10.

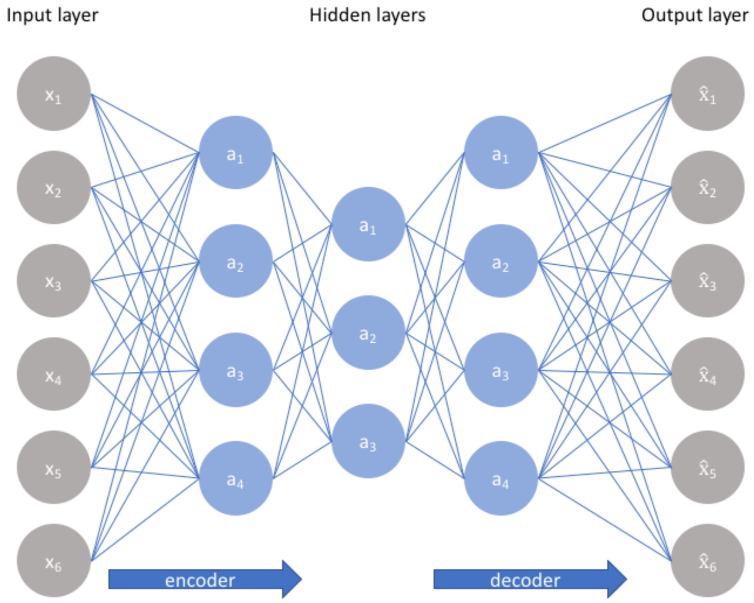


Figure 10: Autoencoder showing encoder and decoder layers. Input data is reduced through a bottleneck, forcing the model to "learn" heartbeat patterns[2].

[2] https://www.jeremyjordan.me/autoencoders/

The fundamental structure of a deterministic autoencoder is an encoder network which reduces the data to the goal latent dimension, and a decoder network which reverses these steps to try and reconstruct the original signal. By reducing input data through a low dimensional representation, or bottleneck, we force constituent neural networks to learn patterns in healthy heartbeat data. Models are trained to encode and decode healthy heartbeats and should struggle to reconstruct anomalous or irregular heartbeats.

6.2.2 Convolutional Denoising Autoencoder

Convolutional autoencoders are traditionally used for image classification using two-dimensional inputs representing image data. For this project, we implement a one-dimensional convolutional denoising autoencoder (CDAE). The mean squared error (MSE) between input data and autoencoder output data is used for the loss function. We begin by tuning a convolutional autoencoder, which uses convolution and pooling layers to learn patterns in the 10-heartbeat segments. Compared to an autoencoder using only dense layers, we found that a convolutional autoencoder was able to achieve a similar reconstruction error with more consistent convergence between the training and validation data. This implies that the convolutional autoencoder does not overfit on training data.

To improve this convolutional autoencoder, we perturb the input data with noise. Such denoising autoencoder architectures can lead to more robust feature extraction [20]. To add this denoising component to our convolutional autoencoder, we generate additional augmented training data. The training data for the CDAE consists of the original training data used in the convolutional autoencoder as well as two copies of this original data, perturbed by Gaussian noise (training loss for this model is shown in Figure 11). Amplitude of Gaussian noise is tuned empirically. We found that adding noise to our training data improved the generalizability of our model when evaluated on test data (last two-thirds of each patient). The architecture that we have implemented for the convolutional denoising autoencoder is shown in Figures 12 and 13.

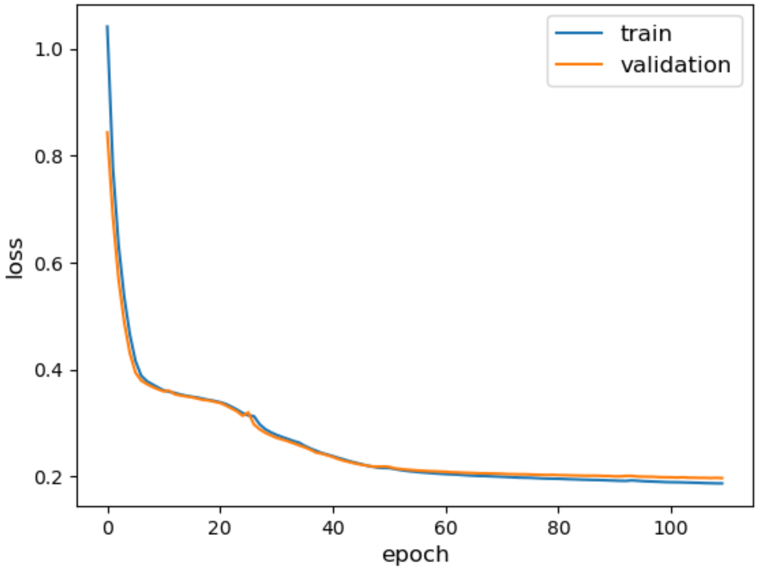


Figure 11: CDAE loss plot showing training and validation accuracy converging for a representative patient.

The input to the model is of dimension (1000,4), as we use 10 heartbeat chunks across four leads for each patient. The encoder compresses the input down to dimension (100,1). As we compress the data in the encoder, the convolution layers are extracting features from the sequences of data and creating feature maps. We have utilized these convolutional layers in conjunction with the longer heartbeat chunks to help the network learn the morphology of these heartbeats. After applying these convolutional layers across all four leads, the data is reshaped and compressed to dimension (100,1). The decoder conducts nonlinear operations that aim to reconstruct the original signal. Using dense layers, we upsample our data and then apply the deconvolution operation to achieve the reconstructed signal. Between the dense layers of the encoder and decoder architectures, we have added dropout for regularization. Further, we have also added l_2 regularization between the segments of the autoencoder model.

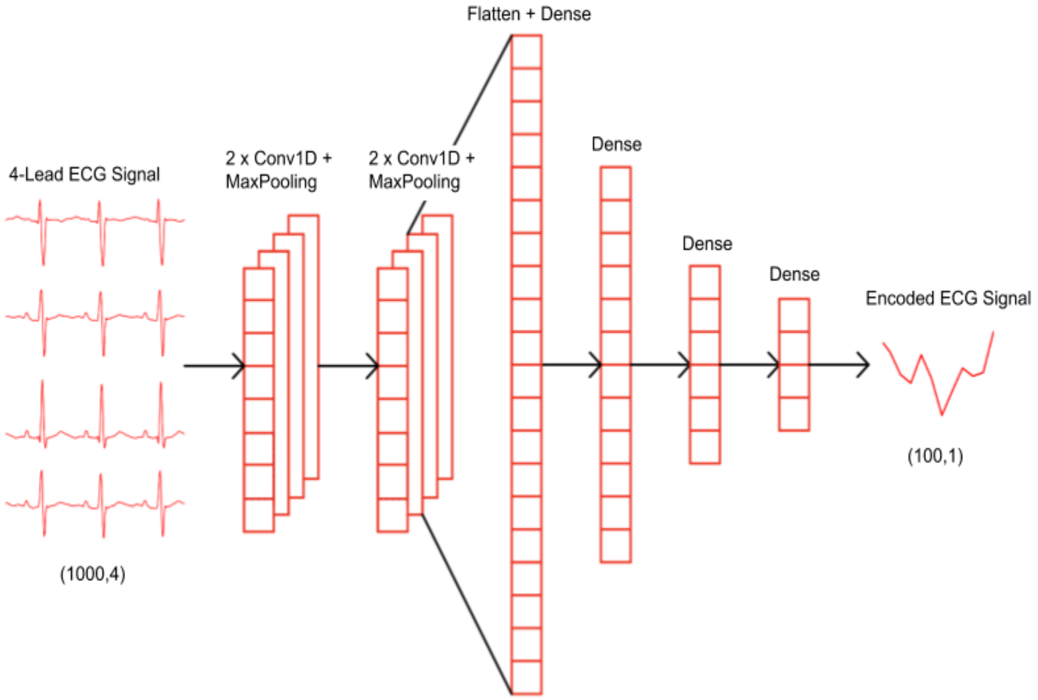


Figure 12: Encoder Network of Convolutional Denoising Autoencoder

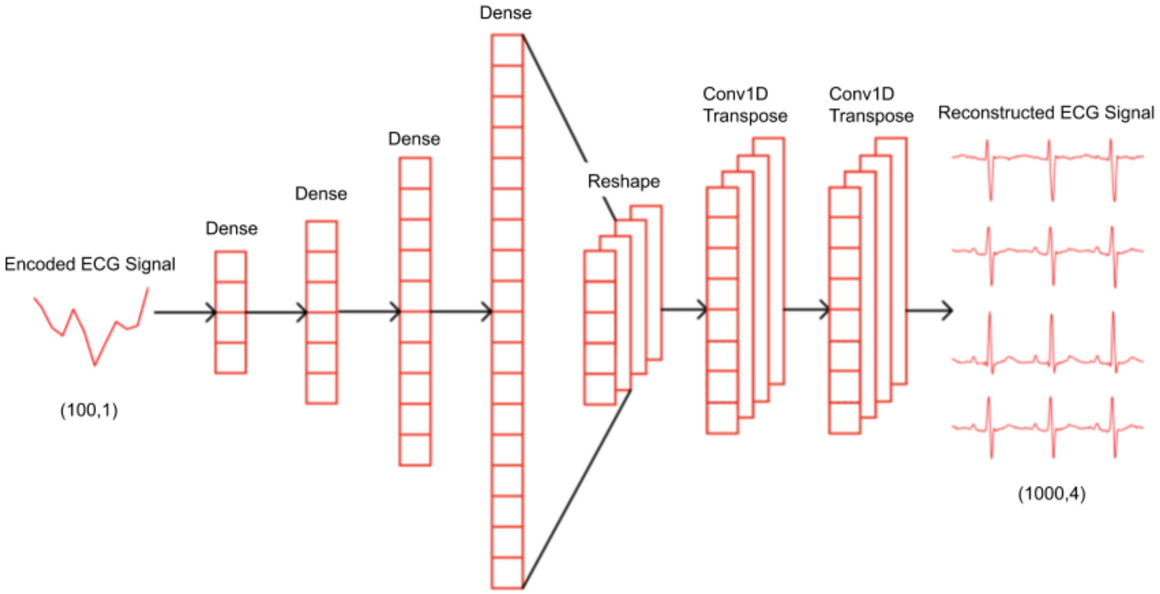


Figure 13: Decoder Network of Convolutional Denoising Autoencoder

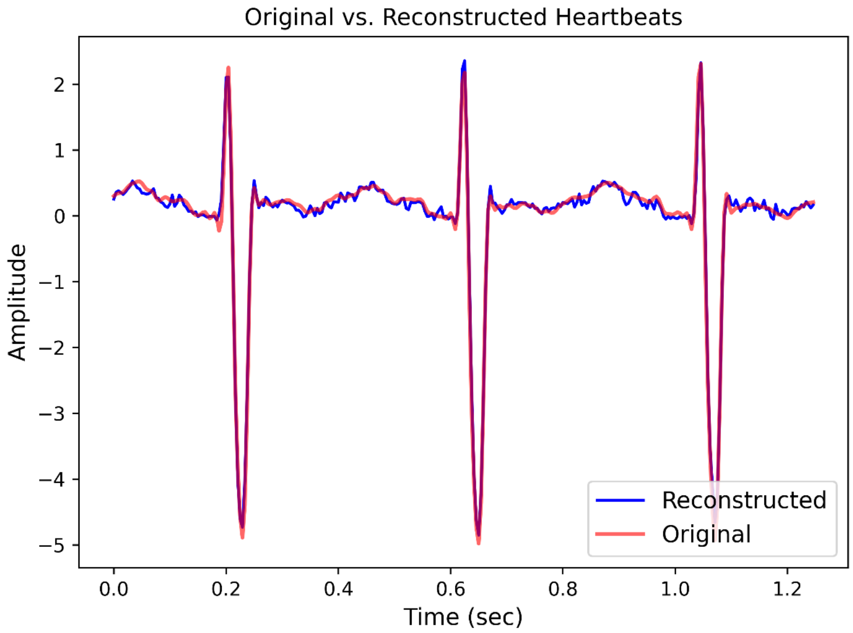


Figure 14: Original Heartbeats versus Reconstructed Heartbeats for a Single Lead

Using the CDAE on the training data, the relative MSE between input and reconstructed signals is about 0.25 when averaged over the entire patient cohort. Because noise is added to the CDAE input, we cannot make a fair comparison with the non-denoising case on the basis of relative MSE. Figure 14 shows a subset of input heartbeats versus reconstructed heartbeats sourced from the normal period of heartbeats for a representative patient from the CDAE (first third of data). This CDAE model is part of the finalized model for the fall 2020 semester, and yields a high false positive rate when applied to control patients, as shown in Figure 21.

6.2.3 Transfer Learning

Due to the false positive rate that we encountered with control patients using the finalized model for fall 2020 (Figure 21), we hypothesize that a natural drift in a HLHS patient ECG morphology occurs over time. In order to account for this drift, we have implemented a time delay transfer learning technique, as seen in Figure 15.

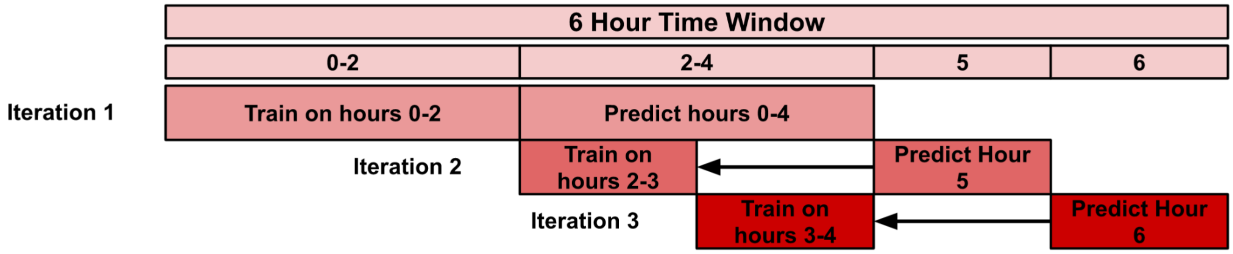
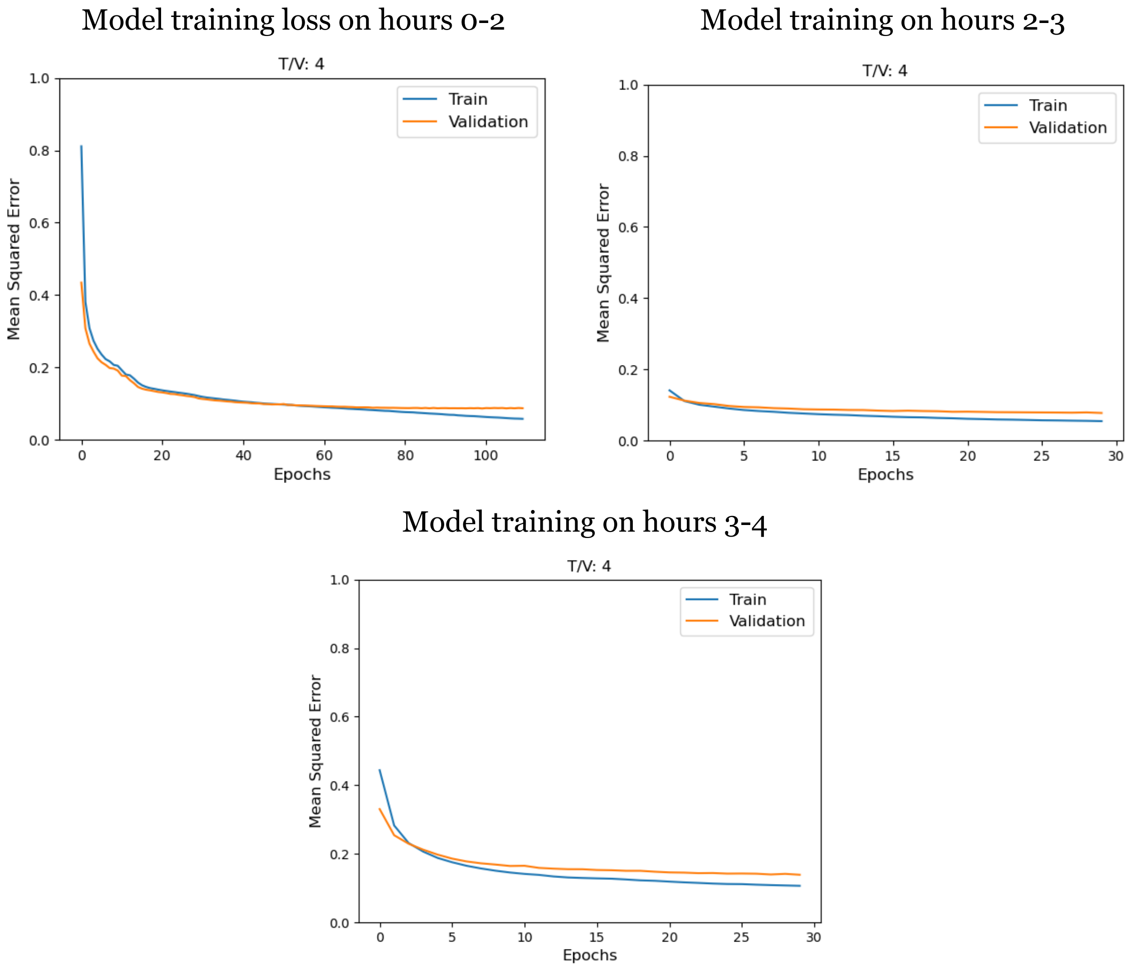


Figure 15: Transfer learning training scheme

To enable the CDAE to better account for this natural drift over time, and hopefully distinguish it from symptoms of a cardiac arrest, our scheme retrains the autoencoder after a period of time. For the first training, we use two hours of data, and subsequent re-trainings require only 1 hour of data. The first training has no time delay, and will use the first two hours of available data for a patient, while the following training requires that the data used has a delay of one hour from the time of retraining. Since the data used is time lagged, as we monitor a patient with this model, we hypothesize that it will help adjust the model over time to the subtle natural changes in the heartbeat morphology. The first model for a given patient is trained on the first two hours of data and is then used to predict the hours from the start to the 4th hour. The model weights generated from this training are then loaded into the next training session as an initial condition. In the second training session the model is trained on hours two and three and used to predict hour five. This process is repeated for hour six, with the initial condition for the model set as the weights from the second training session. By using this scheme, all following training sessions after the first are completed in a nominal time period and can be updated in real time. Loss curves for initial model training and re-training are given in Figure 16.


Model training loss on hours 0-2

Model training on hours 2-3

Model training on hours 3-4

Figure 16: Transfer learning training scheme

6.2.4 LSTM Autoencoders

Another candidate neural network autoencoder architecture employed for the task of reconstructing heartbeats is the Long-Short Term Memory Autoencoder (LSTM Autoencoder). Unlike the CDAE and transfer learning methods, the LSTM explicitly includes time dependence in the neural network architecture. Rather than just autoencoding based on 10 input heartbeats, the LSTM autoencoder inputs include previous heartbeats as well, through recurrent connections. The recurrent connections allow for a time dependent model that is able to capture longer range temporal instabilities. However, these additional connections come at the cost of increased computational complexity as additional connections are represented as additional neurons in the neural network. Aside from these architectural differences, the implementation of the LSTM autoencoder is the same as that of the CDAE. The exact architecture used in this project can be seen in Appendix B.

6.3 Computing Reconstruction Error

Once an autoencoder model is trained for a given patient, we compare the input and output feature vectors and compute a reconstruction error score. Below, we explore two possible ways to compute this reconstruction error, and the final results for each report are summarized in Section 7.

6.3.1 Relative Mean Squared Error (MSE)

Consider the input to the encoder at time point t, $e^{(t)}$, and the output of the decoder, $\hat{e}^{(t)}$. After providing the test data for a given patient as input ($e^{(t)}$) to a trained CDAE, we can obtain a reconstruction ($\hat{e}^{(t)}$) and a corresponding streaming time series of MSE values, $X^{(t)}$, which can be computed using the below equation:

$$X^{(t)} = \frac{1}{n}\frac{||e^{(t)} - \hat{e}^{(t)}||^2}{||e^{(t)}||^2}$$

where $|| \cdot ||$ represents the l_2-norm of the input signal.

6.3.2 Kullback-Leibler Divergence

In our autoencoder models, each of the input and output vectors have dimension 4000, so MSE-based analysis may not be ideal for such high-dimensional settings [31]. Hence, after exploring various alternatives, we propose the use of the Kullback-Leibler (KL) divergence to quantify the difference between the input to the encoder and the output of the decoder at each time point.

The Kullback–Leibler divergence can be used to measure the relative entropy between the original and reconstructed ECG waveforms. To do this, we first compute the element-wise absolute value of each waveform and normalize so that the elements sum 1 (and can thus be thought of as a probability distribution). This process is repeated across each of the 4 leads. As a result, at each time t, we obtain 4 probability distributions corresponding to the 4 leads of $e^{(t)}$ and 4 distributions corresponding to the 4 leads of $\hat{e}^{(t)}$. We denote these distributions as $p_i^{(t)}$and $q_i^{(t)}$, respectively, where $i \in \{1, 2, 3, 4\}$corresponds to the ECG leads. The KL-divergence for each lead i is then given by

$$K_i^{(t)} = \sum_{j=1}^{1000} q_{ij}^{(t)} \log \frac{q_{ij}^{(t)}}{p_{ij}^{(t)}}$$

We then average the score $K_i^{(t)}$across the 4 resultant leads into a composite error score, $X^{(t)} = \frac{1}{4}\sum_{i=1}^{4} K_i^{(t)}$.

6.4 Change-point Detection with CUSUM

In the previous section, we show two methods of computing an error metric, $X^{(t)}$ from the input-output pairs of a patient-specific trained autoencoder. However, it is to be noted that different patients have different baseline values of this reconstruction error, making it difficult for physicians to know if a given error score is atypical. In real-time monitoring applications, a statistic which can be interpreted in a patient-independent manner is therefore highly desirable. To this end, we apply CUSUM [21], a classical univariate change-point detection method, to the raw time series of errors.

Given a signal of reconstruction errors, denoted $X^{(t)}$, the CUSUM statistic $S^{(t)}$is specified by:

$$S^{(0)} = 0$$
$$S^{(t+1)} = max(0, S^{(t)} + L^{(t)})$$

Here, we let $L^{(t)} = \frac{X^{(t)} - c}{\sigma}$. Here, c is set to be the expected value of $X^{(t)}$ for normal ECG data and σ is set to be the standard deviation of $X^{(t)}$ for normal ECG data. In practice, these parameters can be estimated using a validation set of patient data. In our case, we use the sample mean and standard deviation of the first hour of the test data, which is the fourth hour before the cardiac arrest (assumed to be relatively normal data).

Note that, for normal heartbeat data, $L^{(t)}$ has expectation 0, forcing $S^{(t)}$ to also be near 0. However, if the reconstruction error is consistently higher than expected, $S^{(t)}$will grow steadily. The normalization by σ is added to ensure that each patient's CUSUM score grows at the same rate, allowing us to use a common threshold to detect a change (see Results section of this report).

We further note that, in addition to increasing visual interpretability, the recursive nature of CUSUM provides an inherent denoising effect, since only a persistent change in MSE is detected. This is advantageous in real-time monitoring since it is robust to temporary changes in the MSE, often caused by motion artifacts.

Remark: For a more conservative algorithm (i.e. to ensure a larger change is required to cause the CUSUM metric to grow), the value c can be increased by a small positive correction constant. This so-called "correction parameter" can be tuned to improve model performance and to address the needs of medical professionals. The effect of this parameter is explored in more detail in the following section.

7 Results

From our cohort of 44 cardiac arrest patients, 41 patients had more than 10,000 valid heartbeats, which was considered sufficient to be run through our pipeline. From the control group, 52 patients had enough valid ECG data to run through the pipeline.

7.1: Total Performance Comparison of Model/Error Metric Pairings

To summarize our pipeline, we use an autoencoder model to compress and reconstruct the heartbeats, and then we use a distance function (i.e. error metric) to measure the distance, or "error", between the original and reconstructed heartbeats. These error metrics are tracked over time via a CUSUM-based change-point detection algorithm. Over the past semester, we tried various combinations of models and error metrics to achieve the highest possible true positive rate while minimizing the false positive rate. To compare the performance of these various combinations of models and error metrics, we compute the best ROC curve for each model and overlay them in a single plot for comparison.

7.1.1: Computing Optimal ROC Curve For A Model/Error Metric Pairing

Following the modeling stage of the pipeline, there are two parameters we tune to reach a final result. The first is the correction parameter of the CUSUM change-point algorithm, which tunes the sensitivity of the algorithm to sustained perturbations off the baseline (see the Remark in Section 6.4). The second parameter is the threshold value that the CUSUM value must cross in order for the heartbeats to be flagged as a potential cardiac arrest event. In order to evaluate the performance of a particular combination of model and error metric, we find the optimal ROC curve via the following method.

Finding the Most Optimal ROC Curve for a Particular Model/Error Metric Pairing

First, we select a model that we want to use in a particular pairing. We train this model and compute reconstructions over the test and control set.

We then select an error metric for the CUSUM algorithm to use for comparing distances between original and reconstructed heartbeats. With a predetermined correction value, we run this CUSUM algorithm over all patients to generate the CUSUM signal for each patient. With these signals, we sweep over a range of thresholds, and for each threshold, we compute the true and false positive rate by measuring what percentage of CUSUM signals in the test and control set cross the threshold. This generates the true and false positive rates that we can plot to create a single ROC curve.

With this method of computing ROC curves, we then sweep over a range of *n* correction parameters to generate *n* ROC curves with the above method. For example, for the transfer learning convolutional autoencoder with a KL-Divergence error metric, we swept over a range of correction values from 0 to 1 (with an interval of 0.01) and plotted the corresponding AUC score for each value as shown in Figure 17. Each point on the curve corresponds to the AUC score for a single ROC curve as a function of the correction parameter.

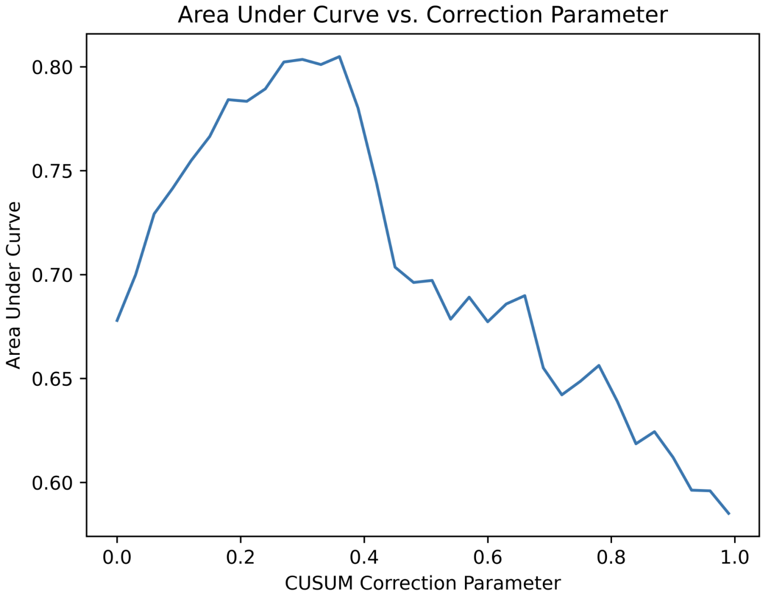


Figure 17: Example of finding the optimal ROC curve by sweeping over correction parameters

From the graph in this example, we see that the correction value *c=0.36* leads to the ROC curve with the highest AUC score. Thus, we select this ROC curve as the best representation of that model's performance.

Over this semester, we tried three autoencoder models: a convolutional denoising autoencoder, a convolutional denoising autoencoder with transfer learning, and an LSTM autoencoder. Furthermore, we tried multiple error metrics, but we will only report the results for two error metrics for this report: mean squared error and Kullback–Leibler divergence. Then, we computed the ROC curve for every possible model and error metric pair (6 total) at the optimal correction parameter. These results are plotted in Figure 18.

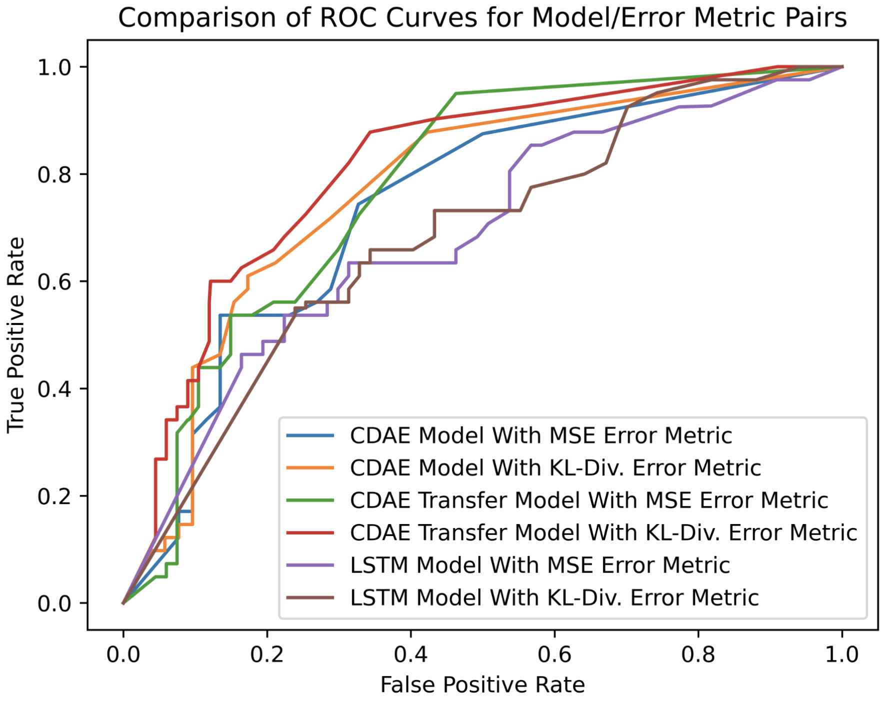


Figure 18: Overlay of Best ROC Curves from Each Model/Error Metric Pairing

The AUC score of the best ROC curves are tabulated in the table below.

	CDAE	**CDAE Transfer**	**LSTM**
MSE	0.739	0.775	0.684
KL Div.	0.772	0.811	0.682

Table 1: AUC Scores of Optimal ROC Curves from each Model/Error Metric Pairing

From the table, we see that the transfer learning convolutional autoencoder paired with the KL-Divergence error metric was our most successful model. By observing the values of the ROC curve, choosing a threshold value of 15 achieves a 88% true positive rate with a 34% false positive rate. Depending on the needs of the medical professional, we can adjust the threshold to vary the true/false positive rates.

7.2: Comparing Best Model Performance From Fall 2020 Semester

Over the course of this semester, we have made significant improvements over our best model that we achieved in the fall semester of 2020. Our primary goal of this semester was to decrease the false positive rate. By using transfer learning to continuously update our model, we better account for concept drift (i.e. change in the underlying statistical properties of the heartbeats) over time. This modification significantly contributed to reducing the false positive rate.

Furthermore, using KL-divergence instead of the relative mean squared error as a distance metric between the original and reconstructed heartbeats significantly contributed to our increased true positive rate. Substituting KL-Divergence for mean squared error allowed the model to more effectively compare distances between heartbeats signals which lie in a high dimensional space. We compare the ROC curves from our best models from the Fall 2020 and Spring 2021 semesters in Figure 19.

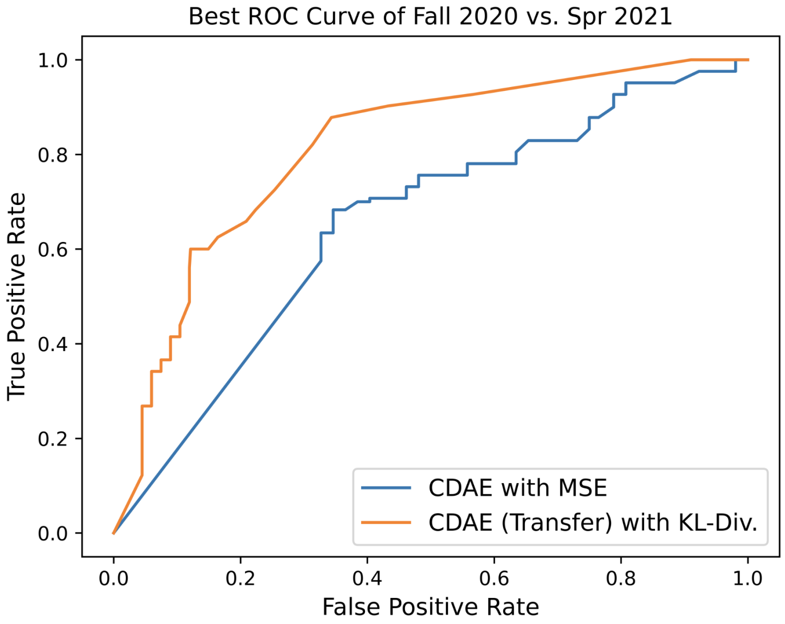


Figure 19: Comparison of ROC Curves from Fall 2020 and Spring 2021 Models

The ROC curve of the Fall 2020 model achieved an area under curve (AUC) score of 0.652. For comparison, the Spring 2021 model achieved an AUC score of 0.811. Furthermore, when we compare the optimal threshold and the resulting true/false positive rates from those curves, we see that we obtain a significant increase in the true positive rate and a significant reduction in the false positive rate.

	Fall 2020 Model (CDAE with MSE)	Spring 2021 Model (CDAE Transfer Learning with KL-Divergence)
True Positive Rate	73%	88%
False Positive Rate	46%	34%

Table 2: Comparison of Optimal True/False Positive Rates of Fall 2020 and Spring 2021 Models

7.3 Optimal Model Results: Transfer Learning and KL-Divergence

As described in the previous section, the optimal model was a CDAE model trained with a transfer learning scheme and paired with the KL-divergence error metric. In this case, when we use a CUSUM threshold of 15, 36/41 test patients cross the threshold, while 17/52 control patients cross the threshold. Furthermore, on the test set, our model detects cardiac arrest an average of **1.32 hours** hours beforehand with a 95% confidence interval of 0.27 hours. Figure 20 gives the comparison between CUSUM metric development over time for the test and control patients. **Note the difference in y-axes scaling between the test and patient cohorts.**

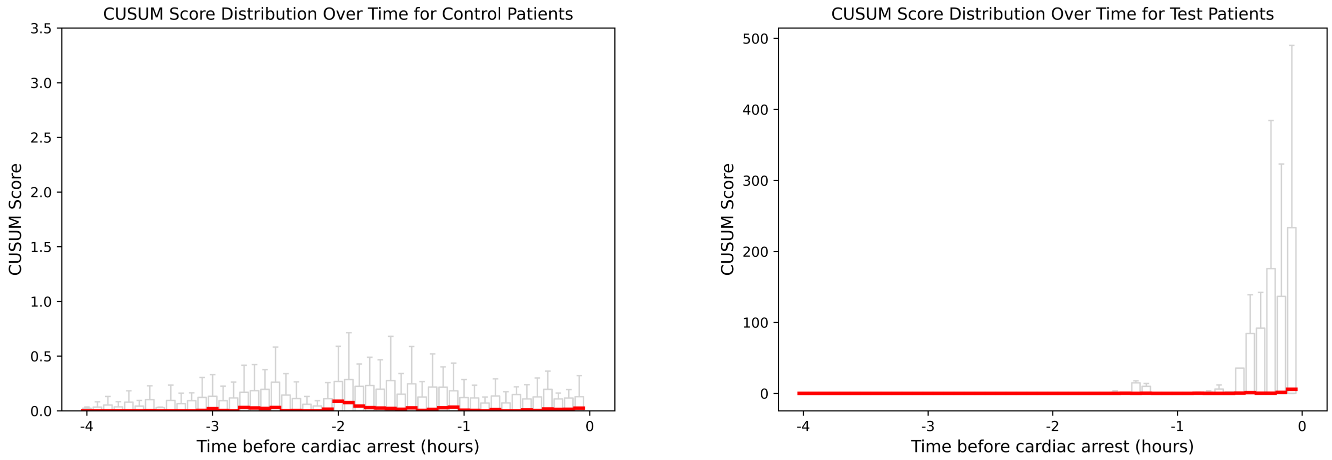


Figure 20: (left) CUSUM score distribution for 52 patient control group, (right) CUSUM score distribution for 41 patient test group

From the boxplot on the test patients, we can see a brief rise in the CUSUM scores that corresponds to the 1.32 hours before cardiac arrest. This large increase in CUSUM score is absent in the control patient cohort.

8 User Application

Currently patients with HLHS require around-the-clock monitoring during the critical time period directly following surgery. Thus, the end goal of this project is to have a user application which runs our algorithm to monitor a patient’s ECG signal in real-time, and alerts the user when cardiac arrest is predicted.

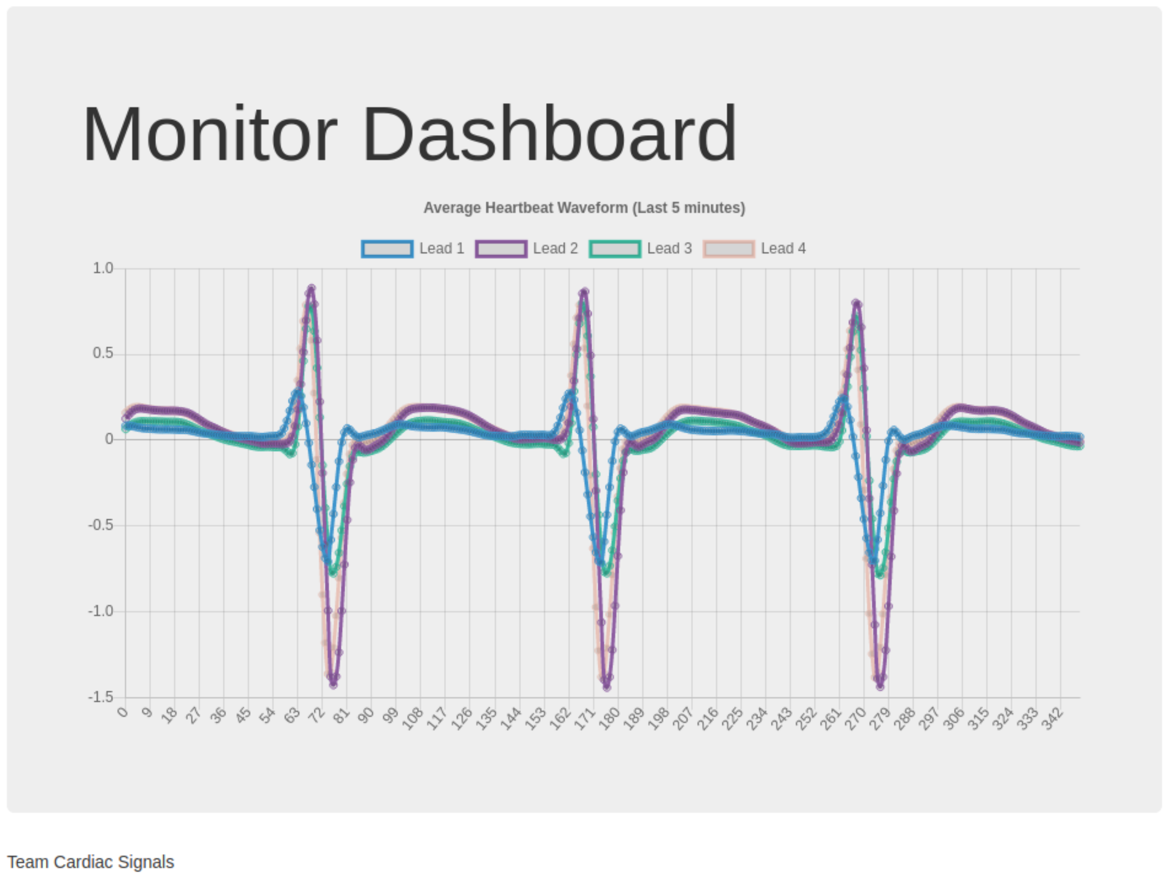


Figure 21: Current Dashboard Prototype

8.1 Web Dashboard

We have created an interface accomplishing both these tasks via a web-hosted dashboard. This medium allows doctors (or other caretakers) to view an array of metrics (including our instability metric) in real-time from anywhere on any device. We have built our prototype using Flask, a lightweight Python framework for web hosting.

8.2 Dashboard Features

The primary feature displayed on the dashboard is our instability metric. If the metric is above our prediction threshold, the dashboard will indicate that cardiac arrest is imminent. Additionally we display graphs showing the history of the metric, giving the user an indication of the trend. Finally we display a chart with all four leads of three complete heartbeats waveforms which are the average waveforms over a recent time period. This chart is interactive, allowing the user to choose which leads are displayed.

8.3 Dynamic Update

The monitoring system evaluates 5-minute chunks of data using our algorithm, thus updating our evaluation metric in real-time five minute intervals. Our dashboard has POST request handlers which refreshes the graphs when the monitor machine sends the corresponding POST request with new data to upload. We do not currently use a database; rather, we store the raw data on the web server itself for simplicity. This is acceptable for our current dataset (consisting of a few hundred patients), but would need to be changed to bring the system up to enterprise scale, or possibly for privacy concerns.

9 Reproducibility

To reproduce the figures and the results in this report, visit the *README.md* file at https://github.com/RiceD2KLab/TCH_CardiacSignals_F20 for more information.

10 Conclusions and Future Work

In this project, we proposed a method for the early detection of cardiac arrest using a convolutional denoising autoencoder in conjunction with a novel change point detection algorithm (CUSUM). Our model is robust to noise and inter-patient variability in ECG signals, and does not require specific morphological feature extraction. These results provide a promising avenue for implementing real-time monitoring applications for patients in a clinical setting for not just ECG but also other time-varying signals. Our model achieves a 88% true positive rate and 34% false positive rate in test and control patients, respectively. Detection of cardiac arrest in test patients occurs on average 1.32 hours before carrest events. Additionally, we have built a web-hosted application for displaying relevant electrocardiogram instability metrics, allowing for convenient patient monitoring. Our work is also likely to assist caregivers and physicians in monitoring HLHS infants, thereby improving patient outcomes.

Citations

1. Raúl Alcaraz and José J. Rieta. 2012. Application of Wavelet Entropy to Predict Atrial Fibrillation Progression from the Surface ECG. Comput. Math. Methods Med. 2012, 7 (Sep 2012), 245213. https://doi.org/10.1155/2012/245213
2. David J Barron, Mark D Kilby, Ben Davies, John G CWright, Timothy J Jones, and William J Brawn. 2009. Hypoplastic left heart syndrome. Lancet (London, England) 374, 9689 (August 2009), 551—564. https://doi.org/10.1016/s0140-6736(09)60563-8
3. Andrea Bravi, André Longtin, and Andrew J. E. Seely. 2011. Review and classification of variability analysis techniques with clinical applications. Biomed. Eng. Online 10, 1 (Dec 2011), 1–27. https://doi.org/10.1186/1475-925X-10-90
4. S. Chauhan and L. Vig. 2015. Anomaly detection in ECG time signals via deep long short-term memory networks. In 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). 1–7. https://doi.org/10.1109/DSAA.2015. 7344872
5. Ivaylo Christov, Gèrman Gómez-Herrero, Vessela Krasteva, Irena Jekova, Atanas Gotchev, and Karen Egiazarian. 2006. Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification. Med. Eng. Phys. 28, 9 (Nov 2006), 876–887. https://doi.org/10.1016/j.medengphy.2005.12.010
6. Gari D. Clifford, Francisco Azuaje, and Patrick McSharry. 2006. Advanced Methods And Tools for ECG Data Analysis. Artech House, Inc., USA.
7. Barbara J. Drew, Robert M. Califf, Marjorie Funk, Elizabeth S. Kaufman, Mitchell W. Krucoff, Michael M. Laks, Peter W. Macfarlane, Claire Sommargren, Steven Swiryn, and George F. Van Hare. 2004. Practice Standards for Electrocardiographic Monitoring in Hospital Settings. Circulation (Oct 2004). https://www.ahajournals.org/doi/10.1161/01.CIR.0000145144.56673.59
8. Vu E. L., Rusin C. G., Penny D. J., Kibler K. K., Easley R. B., B. Smith, D. Andropoulos, and K. Brady. 2017. A Novel Electrocardiogram Algorithm Utilizing ST-Segment Instability for Detection of Cardiopulmonary Arrest in Single Ventricle Physiology: A Retrospective Study. Pediatr. Crit. Care Med. 18, 1 (Jan 2017), 44–53. https://doi.org/10.1097/pcc.0000000000000980
9. Danielle M. Ely and Anne K. Driscoll. 2020. Infant Mortality in the United States, 2018: Data From the Period Linked Birth/Infant Death File. Natl. Vital Stat. Rep. 69, 7 (Jul 2020), 1–18. arXiv:32730740 https://pubmed.ncbi.nlm.nih.gov/32730740
10. I. Jekova, G. Bortolan, and I. Christov. 2008. Assessment and comparison of different methods for heartbeat classification. Med. Eng. Phys. 30, 2 (Mar 2008), 248–257. https://doi.org/10.1016/j.medengphy.2007.02.003
11. Christian S. Jensen, Tung Kieu, Bin Yang, and Chenjuan Guo. 2019. Outlier Detection for Time Series with Recurrent Autoencoder Ensembles. , 2725–2732 pages. https://www.ijcai.org/Proceedings/2019/378
12. Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. arXiv (Dec 2013). arXiv:1312.6114 https://arxiv.org/abs/1312.6114v10
13. Y. Krokhaleva and M. Vaseghi. 2018. Update on prevention and treatment of sudden cardiac arrest. Trends Cardiovasc. Med. 29, 7 (Nov 2018), 394–400. https://doi.org/10.1016/j.tcm.2018.11.002
14. Hongzu Li and Pierre Boulanger. 2020. A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG). Sensors 20, 5 (2020). https://doi.org/10.3390/s20051461
15. Mariano Llamedo and Juan Pablo Martínez. 2010. Heartbeat Classification Using Feature

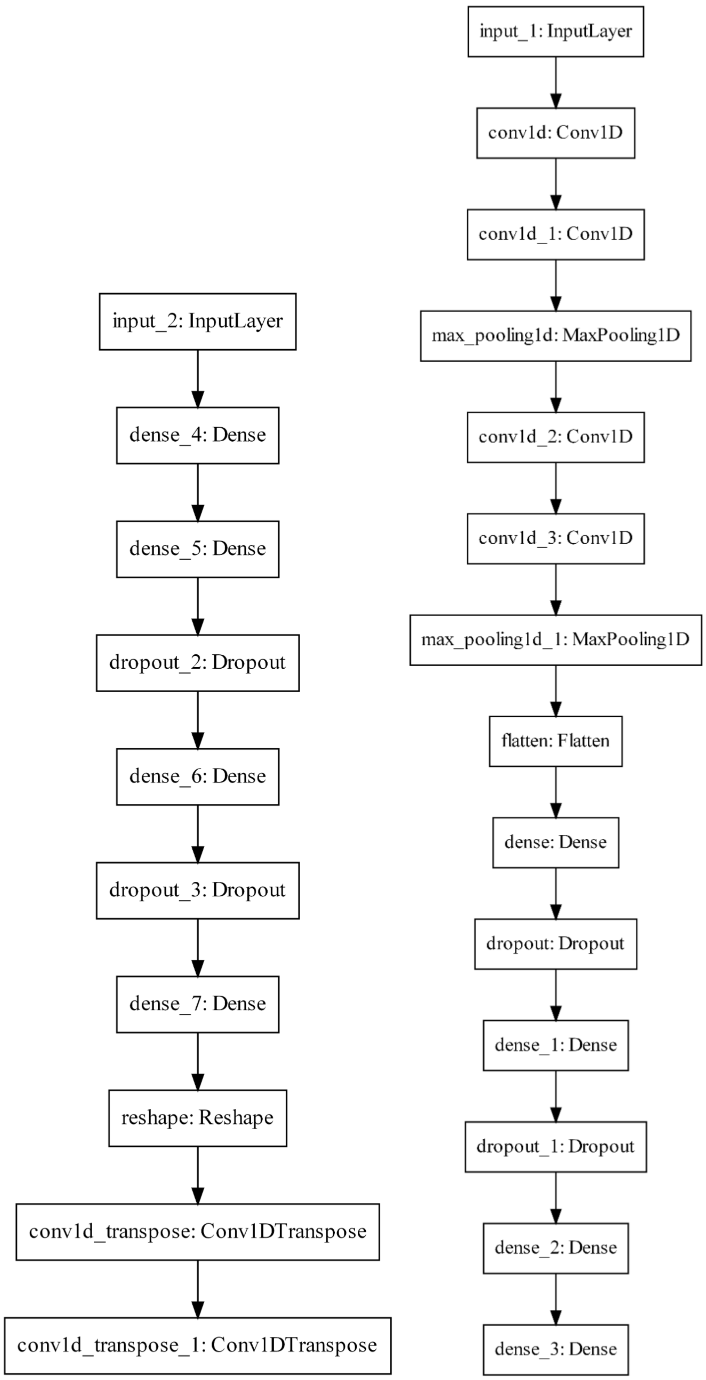
Selection Driven by Database Generalization Criteria. IEEE Trans. Biomed. Eng. 58, 3 (Aug 2010), 616–625. https://doi.org/10.1109/TBME.2010.2068048

16. Eduardo José da S. Luz, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. 2016. ECG-based heartbeat classification for arrhythmia detection: A survey. Comput. Methods Programs Biomed. 127 (Apr 2016), 144–164. https://doi.org/10.1016/j.cmpb.2015.12.008
17. Cara T. Mai, Jennifer L. Isenburg, Mark A. Canfield, Robert E. Meyer, Adolfo Correa, Clinton J. Alverson, Philip J. Lupo, Tiffany Riehle-Colarusso, Sook Ja Cho, Deepa Aggarwal, and Russell S. Kirby. 2019. National population-based estimates for major birth defects, 2010–2014. Birth Defects Research 111, 18 (Nov 2019), 1420–1435. https://doi.org/10.1002/bdr2.1589
18. L. P. McCallister, A. J. Liedtke, and H. C. Hughes. 1979. Ischemic injury to the conducting system of the heart. Involvement of myocardial lysosomes. J. Thorac. Cardiovasc. Surg. 77, 5 (May 1979), 647–661. arXiv:431098 https://pubmed.ncbi.nlm.nih.gov/431098
19. William I. Norwood, Peter Lang, and Dolly D. Hansen. 1983. Physiologic Repair of Aortic Atresia–Hypoplastic Left Heart Syndrome. N. Engl. J. Med. (Jan 1983), 23–26. https://doi.org/10.1056/NEJM198301063080106
20. Keiichi Ochiai and S. Takahashi. 2018. Arrhythmia Detection from 2-lead ECG using Convolutional Denoising Autoencoders. In ACM SIGKDD Deep Learning Day. https://www.semanticscholar.org/paper/Arrhythmia-Detection-from-2-lead-ECG-using-Ochiai-Takahashi/7954bee1180a55acdff127281f25a244b92a3c2e
21. E. S. Page. 1954. CONTINUOUS INSPECTION SCHEMES. Biometrika 41, 1-2 (Jun 1954), 100–115. https://doi.org/10.1093/biomet/41.1-2.100
22. Bayu Wijaya Putra, Muhammad Fachrurrozi, M. Rudi Sanjaya, Firdaus, Anita Muliawati, Akhmad Noviar Satria Mukti, and Siti Nurmaini. 2019. Abnormality Heartbeat Classification of ECG Signal Using Deep Neural Network and Autoencoder. 2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). https://doi.org/10.1109/ICIMCIS48181.2019.8985206
23. John F. Rhodes, Andrew D. Blaufox, Howard S. Seiden, Jeremy D. Asnes, Ronda P. Gross, Julianna P. Rhodes, Randall B. Griepp, and Anthony F. Rossi. 1999. Cardiac Arrest in Infants After Congenital Heart Surgery. Circulation (Nov 1999). https://www.ahajournals.org/doi/full/10.1161/circ.100.suppl_2.Ii-194
24. Craig G. Rusin, Sebastian I. Acosta, Lara S. Shekerdemian, Eric L. Vu, Aarti C. Bavare, Risa B. Myers, Lance W. Patterson, Ken M. Brady, and Daniel J. Penny. 2016. Prediction of imminent, severe deterioration of children with parallel circulations using real-time processing of physiologic data. J. Thorac. Cardiovasc. Surg. 152, 1 (Jul 2016), 171–177. https://doi.org/10.1016/j.jtcvs.2016.03.083
25. Saeed Saadatnejad, Mohammadhosein Oveisi, and Matin Hashemi. 2019. LSTM Based ECG Classification for Continuous Monitoring on Personal Wearable Devices. IEEE J. Biomed. Health Inf. 24, 2 (Apr 2019), 515–523. https://doi.org/10.1109/JBHI.2019.2911367
26. Santanu Sahoo, Bhupen Kanungo, Suresh Behera, and Sukanta Sabut. 2017. Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities. Measurement 108 (Oct 2017), 55–66. https://doi.org/10.1016/j.measurement.2017.05.022
27. R. M. Shaw and Y. Rudy. 1997. Electrophysiologic effects of acute myocardial ischemia. A mechanistic investigation of action potential conduction and conduction failure. Circ. Res. 80, 1 (Jan 1997), 124–138. https://doi.org/10.1161/01.res.80.1.124 arXiv:8978331
28. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning (Helsinki, Finland) (ICML '08). Association for

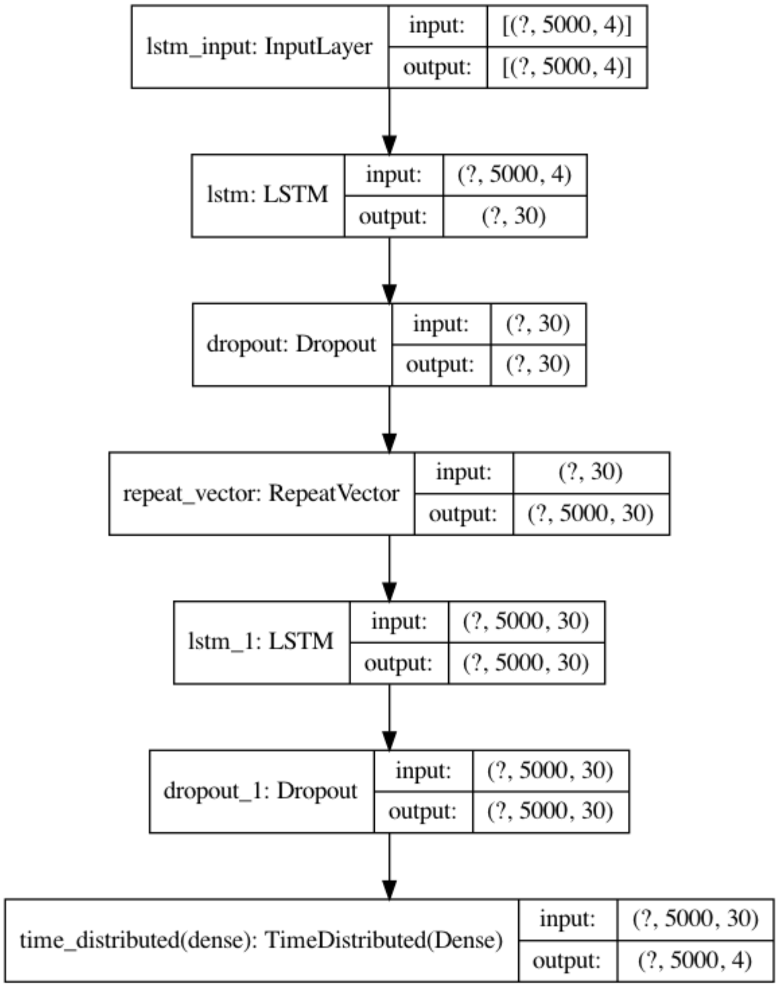
Computing Machinery, New York, NY, USA, 1096–1103. https://doi.org/10.1145/1390156.1390294

29. M. Wess, P. D. Sai Manoj, and A. Jantsch. 2017. Neural network based ECG anomaly detection on FPGA and trade-off analysis. In 2017 IEEE International Symposium on Circuits and Systems (ISCAS). 1–4. https://doi.org/10.1109/ISCAS.2017.8050805
30. Özal Yıldırım, Paweł Pławiak, Ru-San Tan, and U. Rajendra Acharya. 2018. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. Comput. Biol. Med. 102 (Nov 2018), 411–420. https://doi.org/10.1016/j.compbiomed.2018.09.009
31. Thudumu, S., Branch, P., Jin, J. et al. A comprehensive survey of anomaly detection techniques for high dimensional big data. J Big Data 7, 42 (2020). https://doi.org/10.1186/s40537-020-00320-x

Appendix A: Encoder and Decoder Architecture for CDAE



Appendix B: LSTM Autoencoder Architecture


lstm_input: InputLayer
input: [(?, 5000, 4)]
output: [(?, 5000, 4)]
lstm: LSTM
input: (?, 5000, 4)
output: (?, 30)
dropout: Dropout
input: (?, 30)
output: (?, 30)
repeat_vector: RepeatVector
input: (?, 30)
output: (?, 5000, 30)
lstm_1: LSTM
input: (?, 5000, 30)
output: (?, 5000, 30)
dropout_1: Dropout
input: (?, 5000, 30)
output: (?, 5000, 30)
time_distributed(dense): TimeDistributed(Dense)
input: (?, 5000, 30)
output: (?, 5000, 4)