Arrhythmia Classification in Multi-Channel ECG Signals Using Deep Neural Networks

Kyungna Kim

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-80
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-80.html

May 19, 2018

Arrhythmia Classification in Multi-Channel ECG Signals Using Deep Neural Networks

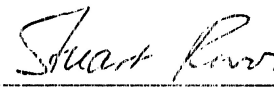
by Kyungna Kim

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

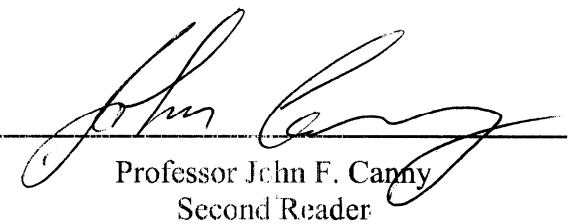
Committee:



Professor Stuart J. Russell
Research Advisor

May 6, 2018

(Date)



Professor John F. Canny
Second Reader

5/15/18

(Date)

Arrhythmia Classification in Multi-Channel ECG Signals Using Deep Neural Networks

Abstract

An important diagnostic tool in identifying heart rhythm irregularities, known as arrhythmias, is the electrocardiogram (ECG). Accurate identification of arrhythmias is critical to patient well-being in clinical settings, as both acute and chronic heart conditions are typically reflected in these readings. This is known to be a difficult problem even for human experts, due to variability between individuals and inevitable noise. We explore the use of deep neural networks for the task of classifying ECG recordings using recurrent and residual architectures. Using a dataset of 106 patient readings, we train several deep networks to categorize slices of ECG data into one of six classes, including normal sinus rhythm, artifact/noise, and four arrhythmias of varying levels of severity. We investigate the usefulness of multi-channel ECG data without additional feature extraction in this problem, especially when common frequency domain transformation feature representation may not be suitable due to low periodicity of data.

Contents

List of Figures

List of Tables

Acknowledgments

I thank my advisor, Professor Russell, for his expertise and guidance during my years at UC Berkeley. I also thank Professor Canny for being second reader.

This would not have been possible without the help of my mentors; many thanks to Paria Rashidinejad, my closest collaborator, for her advice and support over the duration of my graduate career. I also thank Yusuf Bugra Erol for initially introducing me to medical applications of machine learning when I was an undergraduate, and the subsequent projects that led to this study.

Lastly, I thank my friends and family for their unending encouragement and support.

Chapter 1

Introduction

An important diagnostic tool in identifying chronic and acute heart rhythm irregularities (cardiac arrhythmias) is the electrocardiogram (ECG), which measures electrical heart activity via electrodes placed on a patient's skin. ECGs are ubiquitous in intensive care units (ICUs), where clinicians must be able to make critical care decisions quickly and accurately. The ability to correctly distinguish various arrhythmias from each other is crucial for patient well-being; in many cases, the wave morphologies of benign and lethal arrhythmias can be difficult to distinguish.

Existing monitoring systems for ECGs record a myriad of vital signs and also utilize algorithms to determine changes in cardiac rhythm. However, accurate identification of arrhythmias is known to be challenging even for medical professionals, and requires considerable medical expertise. A study investigating diagnostic accuracy for licensed general practitioners showed a specificity of 92% and sensitivity of only 80% in distinguishing atrial fibrillation from healthy sinus rhythms [1]. Waveforms often show variation given an individual's unique biological characteristics, even for arrhythmias whose identifying patterns are known and well-documented.

ECG recordings also suffer from several potential sources of considerable noise, including device power interference (as the measurements themselves are voltages), baseline drift, contact noise between the skin and the electrode, and motion artifacts. These motion artifacts, in particular, can be caused by any muscular activity from the patient; even innocuous movement can be mistakenly registered as arrhythmia. Many of our data points are classified as artifacts due to lead failure, excessive measurement noise, or even unclassifiable arrhythmia.

The combination of inter-patient variability and noise makes this a challenging algorithmic classification problem, and a variety of methods have been proposed to increase diagnostic accuracy. Standard regression and feed-forward neural network models have been explored in the past, and more recent approaches also utilize deep CNN or RNN structures. Feature engineering and spectral analysis are also popular methods for adding or replacing features, though this often limits the scope of the classification.

To exploit the inherently time-dependent nature of ECG readings, we investigate the use of LSTM networks, commonly used to classify or generate sequential data, to distinguish

normal sinus rhythm from artifacts and several arrhythmias of varying levels of severity. In addition, we explore a combined architecture that includes residual connections, which have been have shown to improve classification performance without the significant increase in model complexity typically seen in deep learning architectures.

As we have access to multi-channel data, we incorporate this increased dimensionality into our algorithm, in contrast to the single-channel input format commonly used in ECG classification. Individual channels record cardiac electrical activity from various spatial angles, and the use of multiple channels is likely to give deeper insight into any underlying patterns of arrhythmia that can be interpreted by our model.

We train two baseline models: a single-layer, unidirectional LSTM, and a convolutional neural network without residual connections. We then train models from three categories (LSTM only, residual networks, and LSTM-CNN combined networks), for a total of six models not including our baseline models.

All models exceed baseline performance on training, validation, and test accuracy. A 2-layer, bidirectional LSTM achieves the best performance overall; though networks with residual structures achieve higher training accuracy, their validation and test accuracies are lower than those of the LSTM only networks. The overall F1 score of 0.803 achieved by this network exceeds reported cardiologist F1 scores reported in [2].

Chapter 2

Background

2.1 Cardiac Arrhythmias

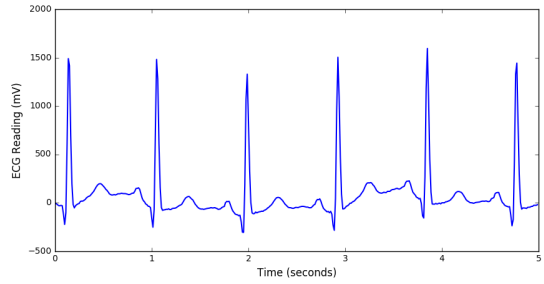
The rhythm of a human heart is regulated by electrical signals produced by two nodes within the heart and conducted through a series of specialized cardiac cells. During healthy, normal operation, this occurs at regular intervals and the electrical signal, which causes the heart muscles to contract, propagates via the cardiac electrical conduction system along the correct path through the atria and ventricles.

Cardiac arrhythmia occurs when the heartbeat is too fast (tachycardia), too slow (bradycardia), or altogether abnormal. Both atrial and ventricular arrhythmias can have any number of causes, including scar tissue from previous trauma (such as myocardial infarction) and coronary disease, and can even occur in healthy hearts. Though many arrhythmias are asymptomatic, those that are not can cause symptoms as mild as occasional palpitations or as severe as stroke and sudden cardiac death. As arrhythmias are caused by disorders of the electrical conduction system, they are reflected in ECG readings as abnormal waveforms.

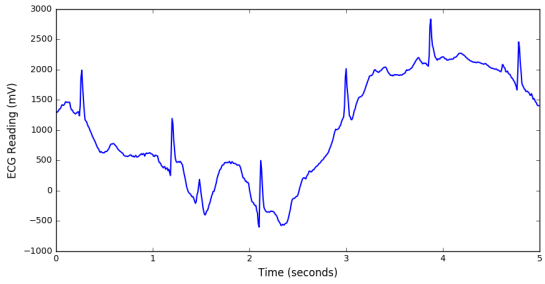
The complexity of this system often necessitates that clinicians use anywhere from six to twelve ECG leads to capture electrical activity across multiple spatial planes [3]. A single lead only provides a 'projection' of this activity across one specific plane and may not provide enough information to make accurate diagnoses of underlying pathologies.

The rhythms we aim to classify are normal (sinus) rhythm, noise/artifact (or otherwise unidentifiable arrhythmia), ventricular tachycardia (VT), atrial fibrillation (AF), bigeminy, and premature ventricular contraction (PVC). An example of each class is shown in figure 2.1; only one channel is plotted for clarity.

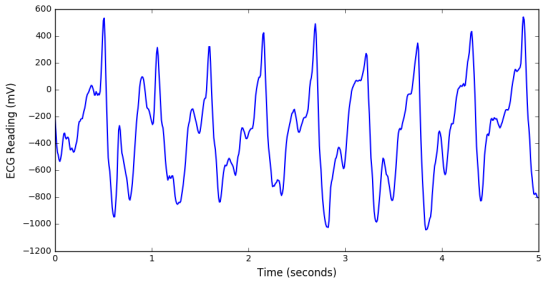
Figure 2.2 presents a case where using multiple channels is key to correct classification. While the single channel recording in 2.2a implies a normal (albeit slightly noisy) sinus rhythm, the full multi-channel recording of the same data point in 2.2b shows a large, abnormal wave about 3 seconds in. This is labeled as an artifact in our dataset; had we considered only the data shown in 2.2a, this would have likely been classified as 'sinus'.



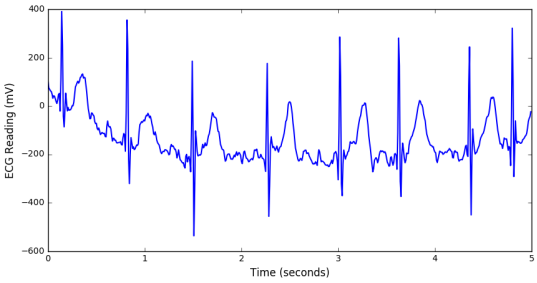
(a) Sinus/normal



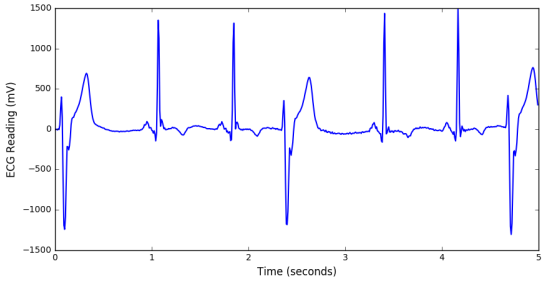
(b) Artifact



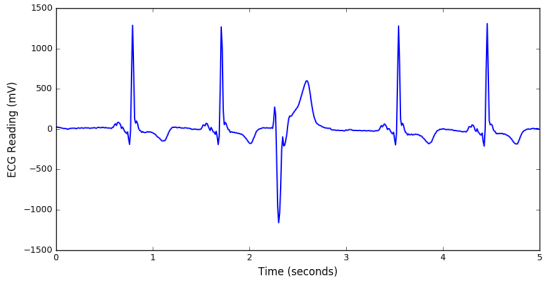
(c) Ventricular tachycardia



(d) Atrial fibrillation

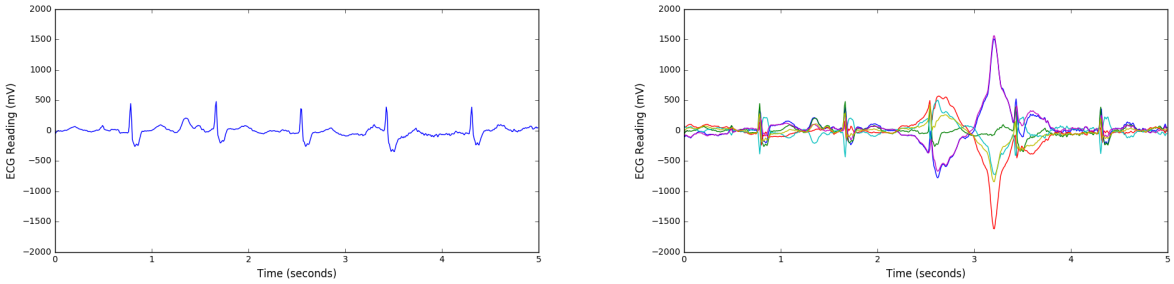


(e) Ventricular bigeminy



(f) Premature ventricular contraction

Figure 2.1: Examples of rhythm types



(a) Single-channel recording (b) Multi-channel recording

Figure 2.2: Single vs multi-channel data

Clinical Significance

Differentiation between various types of arrhythmias is critical. For example, ventricular bigeminy, a premature beat every two beats, can occur for short periods without symptoms in healthy people. They are a general type of premature ventricular contraction (though a label of PVC typically indicates a single premature beat), which, when occurring rarely enough, is clinically inactionable [4]. In contrast, sustained ventricular tachycardia or complete heart block can lead to cardiac arrest within minutes and thus require immediate intervention.

While high sensitivity of arrhythmia classification is important, its specificity is also significant in practice. Repeated false positive alarms from monitoring devices, which can be caused by innocuous motion artifacts or lead failures, lead to a documented phenomenon among clinicians known as alarm fatigue. This can lead to false positive rates as high as 89% [4], which can cause desensitization or annoyance, and can be a significant safety hazard for patients.

2.2 Related Work

Finding efficient, accurate ECG classification algorithms is not a new problem; currently available ICU monitoring devices are able to perform live diagnosis based on observed recordings. However, the accuracy of these devices—specifically in differentiating between morphologically similar cardiac rhythm types and safely reducing false positive rates—remains an issue.

As far back as 1990, simple feed-forward networks have been proposed for classifying underlying cardiac conditions from ECGs [5]. Hidden Markov models have also been used in conjunction with QRS waveform detection for classifying specific arrhythmias, though these have been limited to distinguishing between very few possible classes, such as supraventricular tachycardia [6] or premature ventricular contraction versus beat fusion [7].

The periodic nature of most ECG signals has also been an area of interest, and spectral analysis has been used specifically to identify sustained ventricular tachycardia [8]. Alterna-

tive time-frequency distributions to Fourier transform have been explored in the identification of various ventricular arrhythmias [9].

In general, much work in this area, until recently, has relied on extracting hand-crafted features and comprehensive prior knowledge of specific arrhythmias and their waveform patterns. This limits the number of classes that can be reliably differentiated with a given model, as the morphology of different waveforms is usually highly specific to a given arrhythmia. More recent developments in arrhythmia classification research have utilized well-known deep learning algorithms such as deep CNNs and incorporation of skip/residual layers to improve classification accuracy [2].

Leveraging multi-channel time series data in clinical settings for various classification tasks is an active area of research [10]. In recent work, this approach has been used to classify human activities based on motion sensor data [11]. Existing techniques for ECG classification in particular almost exclusively use single-channel data [5–7], however, likely due to lack of data. The ECG recordings provided by the PhysioNet database of physiological signals [12], used for testing and training many ECG classification algorithms, are primarily dual or single-channel.

Chapter 3

Methods

3.1 Data

The data used to train the models is a set of ECG recordings, obtained from UCSF, for 106 patients for varying lengths of time. Seven channels are provided, though one is removed during extraction as it is simply a linear combination of three of the other channel readings. These include manually labeled alarms of specified length—this is distinct from many of the datasets found in the MIT-BIH database [12] that is typically used for ECG classification tasks, as the MIT-BIH data typically does not provide duration information for any of the alarms.

The window for each sample is either truncated or padded to 5 seconds as many of the viable 'alarm segments'—namely, alarms not caused by internal error—are close to this length. As this data was passively collected, the distribution of classes is extremely skewed, so more common classes (specifically, sinus rhythm and noise) were downsampled to achieve a more even class distribution.

We extract over 150,000 segments classified as one of six classes, as detailed in section 2.1. From this dataset we sample nearly 12,000 data points due to the aforementioned class imbalance.

We split this dataset into training, validation, and test sets, at 68%, 17%, and 15% respectively (obtained from an initial 85/15 split into training and test data, followed by a 80/20 split of that training data into training and validation sets). To achieve reasonable training time, the samples were decimated by a factor of 2 to achieve an effective sampling rate of 120 Hz from the original 240 Hz. Each sample, thus, is of size (600×6). Each sample is then assigned a ground truth label from its corresponding alarm, regardless of the length of the actual alarm (as the data is padded if it is shorter than our 5 second window).

Type	Original	Downsampled
Sinus rhythm	126,435	4,000
Artifact/Noise	18,953	3,000
Ventricular tachycardia	249	249
Atrial fibrillation	1,452	1,452
Bigeminy	896	896
PVC	2,374	2,374
Total	150,359	11,971

Table 3.1: ECG dataset profile

3.2 Model Architectures

LSTM

Long short-term memory units are used to build recurrent neural networks. In particular, they address the issue of exploding/vanishing gradients in standard recurrent architectures. Several gates control updates from inputs, as well as how much of a memory state is retained and passed on to the proceeding timestep. Notably, the cell state c_t (for timestep t) is only ever gated element-wise with the forget gate f and the input activation gate g without undergoing linear transformation. This allows gradients to pass through many steps in time without vanishing (hence 'memory'), and thus the model is able to learn long-term dependencies within the data. Formally, we define the relationship between cell inputs x_t, h_{t-1}, c_{t-1} and cell outputs h_t, c_t for data inputs x_t, hidden states h_t, and cell states c_t at a given timestep t as follows (further detailed in [13]):

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) = \sigma(\hat{i}_t) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) = \sigma(\hat{f}_t) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) = \sigma(\hat{o}_t) \\ g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) = \tanh(\hat{g}_t) \end{aligned} \tag{3.1}$$

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{3.2}$$

The first four equations (barring nonlinearities) can be summarized as the following:

$$\begin{pmatrix} \hat{i}_t \\ \hat{f}_t \\ \hat{o}_t \\ \hat{g}_t \end{pmatrix} = W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_g \end{pmatrix} \tag{3.3}$$

W, b define the weights that multiply cell inputs and added bias terms respectively, and are thus the parameters learned by the model. The hidden state vector (h_t) corresponds to cell

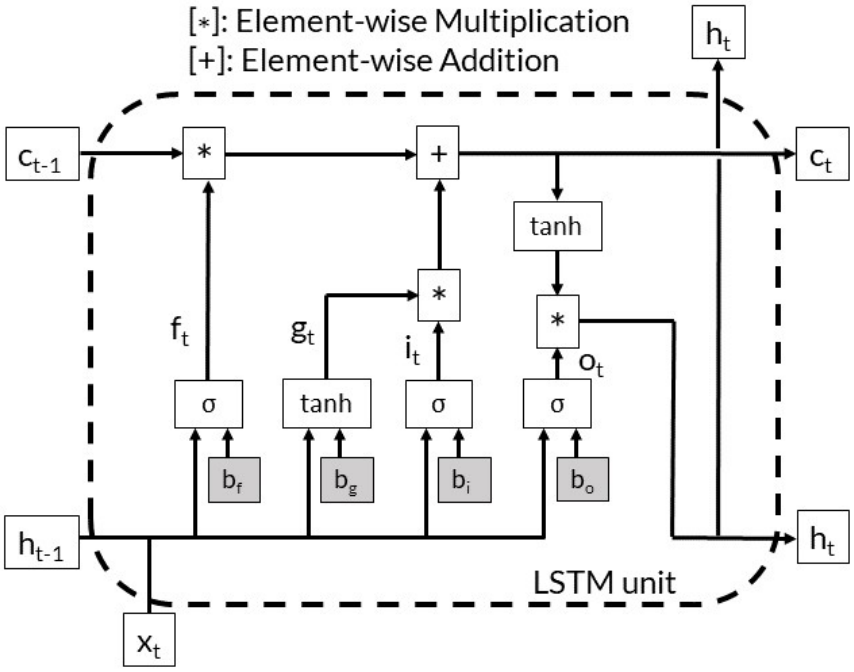


Figure 3.1: An LSTM unit

'output' and is passed along to the next unit, at the next time step, along with cell state c_t. We can either consider the output of just the last unit (for final output classification from the last layer), or from all time steps (when sequences are needed, or to pass into another layer of LSTM units). A visual representation of a single LSTM unit is shown in figure 3.1.

For our LSTM networks, an individual sample is treated as a 600-timestep sequence, where each timestep is a 6-feature data point. In addition to a single-layer unidirectional baseline model, we also train 2- and 5-layer stacks of LSTM units, where the output of each LSTM unit not in the final layer is treated as input to a unit in the next. To increase the contextual information available to the model, we also train a bidirectional variant of the 2-layer LSTM model. These use the same update equations, but add a second LSTM flowing backwards in time; cell outputs from both networks are then concatenated at each time step. As this allows propagation of information from future timesteps, the model can often learn more contextual information and has been shown to improve performance in many tasks [14]. Full high-level architecture of the 2-layer bidirectional LSTM network is shown in figure 3.2.

Residual Networks

Residual networks, as introduced by [15], address several of the concerns in using deep networks for classification (among other learning tasks), including exploding or vanishing gradients [16] and accuracy degradation from increased network depth [17]. Techniques such as normalized initialization of weights [18] and independent normalization of each training

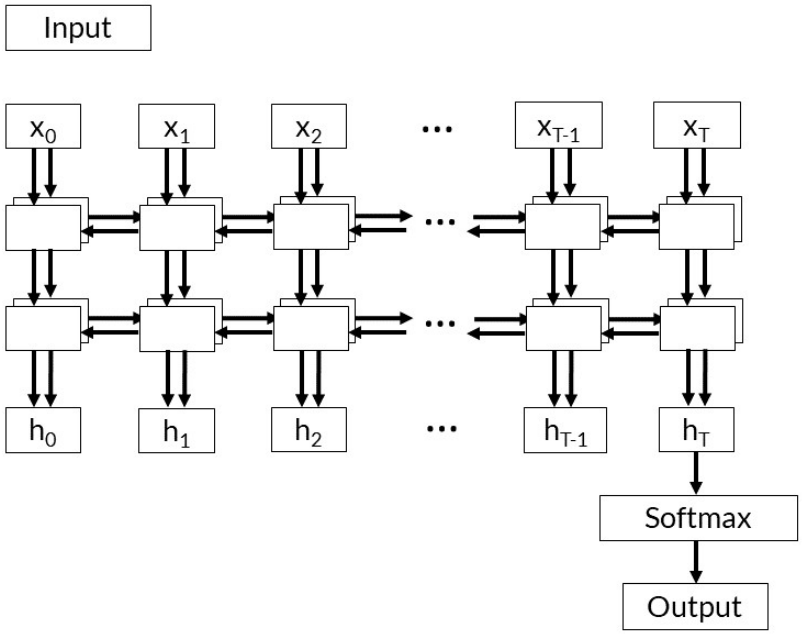


Figure 3.2: Bidirectional LSTM network with 2 stacked layers

batch [19] have shown to be effective in allowing deep networks to converge using standard gradient descent and backpropagation.

However, the saturation (and eventual degradation) of accuracy at high depth levels remains an issue. As it is reflected in decreased training accuracy [15], this is not a result of overfitting training data. By adding identity shortcuts over various blocks in the network, gradients can flow through many layers during training without vanishing. We can thus take advantage of the increased representational power provided by deep networks.

In the general case, we have an input x to some set of layers (for example, a block of convolution, max-pooling, and activation layers) and an output $F(x; W)$ for a set of internal parameters W. The output of a residual block, which includes this set of layers and a residual connection, is defined to be:

$$y = H(x) = F(x; W) + x \tag{3.4}$$

$F(x; W)$ is the *residual*, or the difference between input x and output y, that is to be learned. Residual connections may take different forms; in this case, our residual connections are maxpooling layers (due to reduction in size from convolutional layers).

For our baseline and deep CNN architectures, each sample is treated as set of 6 single-channel 600-timestep recordings. These subnetworks are trained in parallel before their outputs are concatenated and fed into the fully connected and softmax block for classification output. Each subnetwork contains multiple one-dimensional convolution and maxpool layers are stacked to form the network, and we also utilize batch normalization [19] and dropout as regularization techniques. In the case of the deeper network, we add a shortcut connection

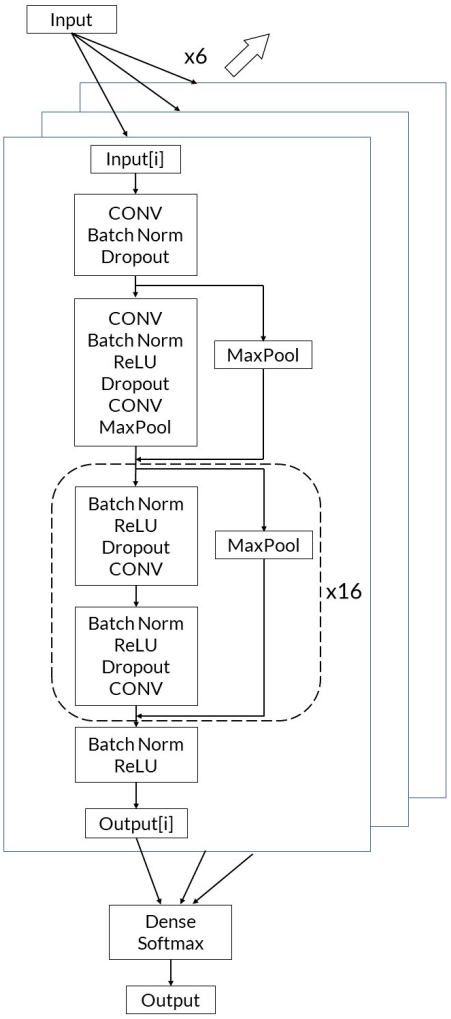


Figure 3.3: Residual network, showing only the CNN architecture

between every two convolutional layers after the first few convolutional layers to form the residual network structure (where the residual connection itself is a maxpool layer). In total, we use 35 convolutional layers in each subnetwork. Full high-level architecture of our residual network is shown in figure 3.3.

Combined Networks

To utilize both the pattern recognition afforded by deep CNNs and the temporal learning ability of LSTMs, we also train an additional architecture that combines them into a single model. We begin with a stacked LSTM to extract temporal structures from the data, and instead of feeding the unrolled hidden state into another LSTM layer, we feed it as input into a (deep) CNN to extract localized features. In the combined model, we begin by feeding the

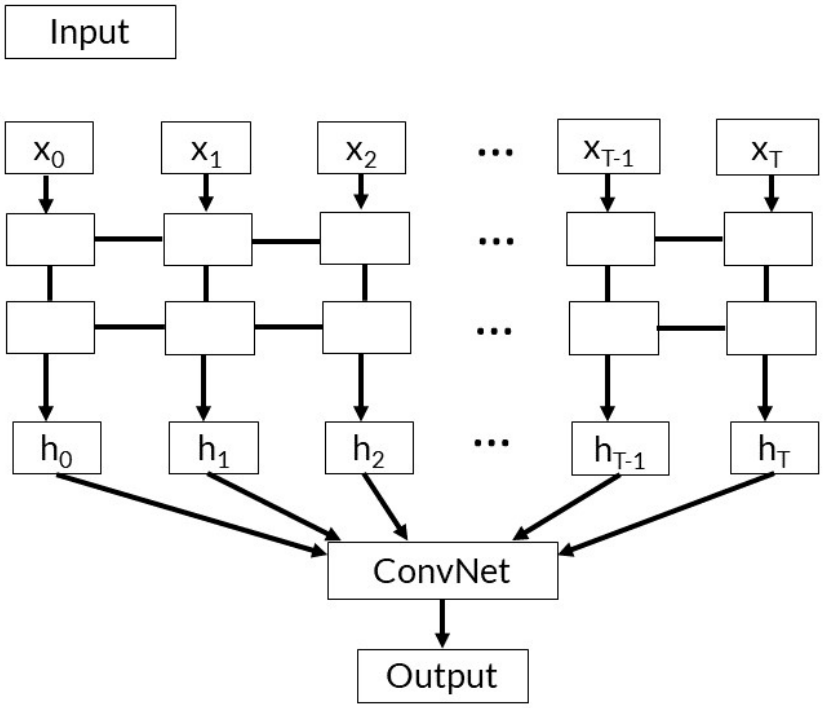


Figure 3.4: Combined LSTM-CNN model (LSTM portion may be uni- or bidirectional)

data into a 2-layer LSTM. The output of the final LSTM layer is treated as a one-dimensional image of size (100×600), and fed into a CNN to extract localized features. We also train a similar architecture with a bidirectional 2-layer LSTM, where the image is of size (200×600).

Full high-level architecture of our combined network is shown in figure 3.4.

3.3 Metrics

For a given classification label, we consider how well a model predicts samples to be positive or negative cases of that label. This is broken down into four categories: TP (true positive) for correctly labeled positive predictions, TN (true negative) for correctly labeled negative predictions, and FP (false positive) and FN (false negative) for incorrectly labeled positive and negative predictions respectively. In addition to overall accuracy ($\frac{TP+TN}{\text{all predictions}}$), we also utilize the following metrics, due to their clinical and practical significance:

$$\text{Recall / Sensitivity} = \frac{TP}{TP + FN} \tag{3.5}$$

$$\text{Precision / Positive Predictive Value} = \frac{TP}{TP + FP} \tag{3.6}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.7}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3.8}$$

In the context of this problem, recall reflects accurate positive identification of an arrhythmia when it occurs in the patient, precision reflects accurate positive identification when identified by the algorithm, and high specificity indicates a low false alarm rate. The F1 score provides a convenient metric for comparing performance as it is the harmonic mean of precision and recall. These metrics are particularly significant in clinical applications, where misdiagnoses of any kind may have severe consequences.

Chapter 4

Experiments and Results

4.1 Setup

We formulate this as a sequence classification problem, and train multiple models to investigate the effectiveness of a combining residual connections with LSTM architecture. We train six model architectures from three categories:

1. LSTM
    - 2-layer unidirectional LSTM
    - 5-layer unidirectional LSTM
    - 2-layer bidirectional LSTM
2. Residual networks
    - 16-block deep residual convolutional neural network
3. Combined models
    - Unidirectional LSTM-CNN
    - Bidirectional LSTM-CNN

Baselines

As baselines, we train a 4-block (each block containing a sequence of convolution, max-pool, batch normalization, and dropout layers) CNN with maxpool residual connections as well as a single-layer, unidirectional LSTM with a hidden dimension of 100 (the dimensionality of the hidden/output space as defined in section 3.2).

Aggregate Accuracy Comparison			
Model	Training	Validation	Test
Baseline (unidirectional) LSTM	66.9%	66.4%	65.7%
Baseline CNN (4 blocks)	68.5%	72.1%	68.9%
Stacked unidirectional LSTM (2-layer)	80.6%	78.2%	79.3%
Stacked bidirectional LSTM (2-layer)	**82.0%**	**79.6%**	**80.1%**
Stacked unidirectional LSTM (5-layer)	80.3%	79.5%	79.4%
Deep residual CNN (16 blocks)	**84.6%**	**75.4%**	**74.9%**
Combined unidirectional LSTM-CNN	83.5%	**77.6%**	**79.5%**
Combined bidirectional LSTM-CNN	**93.3%**	74.9%	76.9%

Table 4.1: Aggregate test set accuracy for all classes using multi-channel data

4.2 Training

All networks are trained using a minibatch size of 64, and 100 epochs of training. We minimize categorical cross-entropy loss in all cases, and we use the Adam optimizer [20], an extension of the classic stochastic gradient descent optimizer.

All LSTM networks use a hidden dimension of 100, as well as a small dropout parameter ($p = 0.1$) for all gates (and thus applied between cell input and output as well as between 'unrolled' instances of each unit). All dropout layers in CNN models used $p = 0.5$ to maximize regularization [21].

To investigate the usefulness of multi-channel ECG data, we also retrain a representative model from each architecture using only one of the available channels per sample (specifically, the BDLSTM with two stacked LSTM layers, deep residual network, and combined LSTM-CNN with two stacked LSTM layers).

4.3 Results

The tables in this section summarize our experimental results. We investigate the overall accuracy of all models (table 4.1), and derive more detailed performance metrics (recall, precision, specificity, and F1 score) of the representative models mentioned above (table 4.3). A comparison between F1 scores of the same three models using both multi-channel (6 channel) and single-channel data is also provided (table 4.4).

Except for VT (whose classification showed relatively poor performance in all models) and PVC, the 2-layer bidirectional LSTM had the highest F1 score out of the three representative models, as shown in table 4.2. The low F1 scores for VT classification are likely a result of its low support in the dataset (figure 3.1). This is also seen to a lesser extent in bigeminy classification, with low recall and F1 scores in the residual network.

From figure 4.1, we see that VT is often mislabeled as PVC; both these arrhythmias have wide QRS complexes [3], which are the characteristic waveform morphologies visible

F1 Score Class Comparison			
Rhythm class	BDLSTM	Residual	LSTM-CNN
Sinus rhythm	**0.832**	0.754	0.783
Artifact/Noise	**0.854**	0.808	0.823
Ventricular tachycardia	0.225	0.069	**0.407**
Atrial fibrillation	**0.827**	0.783	0.774
Bigeminy	**0.683**	0.116	0.543
PVC	0.779	**0.801**	0.714
Overall	**0.803**	0.718	0.752

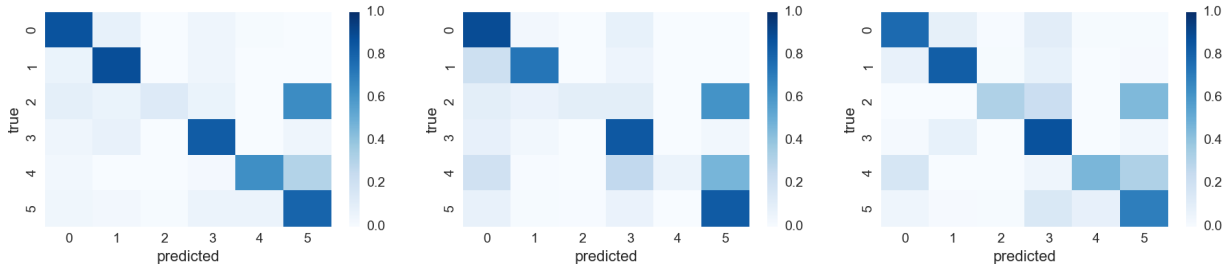
Table 4.2: F1 score comparison over classes for test set, using representative LSTM (2-layer BDLSTM), residual (CNN), and combined (LSTM-CNN) models. Overall F1 score is a weighted average given the class's support in the test set.

Classification Metrics Comparison												
	BDLSTM				Residual				LSTM-CNN			
Class	R	P	S	F1	R	P	S	F1	R	P	S	F1
S	0.83	0.84	0.95	0.83	0.89	0.65	0.87	0.75	0.78	0.79	0.94	0.78
A/N	0.89	0.83	0.95	0.85	0.73	0.90	0.98	0.81	0.82	0.82	0.95	0.82
VT	0.15	0.50	0.96	0.23	0.04	0.47	0.98	0.07	0.32	0.55	0.98	0.41
AF	0.81	0.84	0.95	0.83	0.85	0.73	0.91	0.78	0.87	0.70	0.89	0.77
B	0.71	0.66	0.83	0.68	0.06	0.90	0.99	0.12	0.46	0.66	0.98	0.54
PVC	0.79	0.77	0.89	0.78	0.84	0.77	0.92	0.80	0.67	0.76	0.93	0.71

Table 4.3: Performance metric comparison over classes for test set

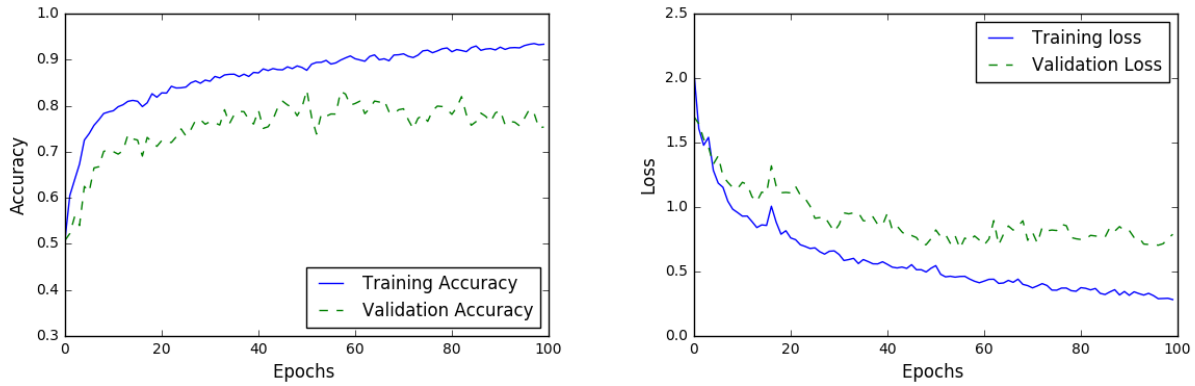
F1 Score Class Comparison						
Class	BDLSTM		Residual		LSTM-CNN	
	Multi	Single	Multi	Single	Multi	Single
S	**0.832**	0.619	**0.754**	0.697	**0.783**	0.706
A/N	**0.854**	0.766	**0.808**	0.744	**0.823**	0.770
VT	**0.234**	0.021	0.069	**0.075**	**0.407**	0.143
AF	**0.827**	0.360	0.783	**0.798**	**0.774**	0.712
B	**0.683**	0.247	**0.116**	0.074	**0.543**	0.526
PVC	**0.779**	0.671	**0.801**	0.707	**0.714**	0.701

Table 4.4: F1 score comparison over classes for test set, comparing single and multi-channel input data



(a) BDLSTM (b) Residual (c) LSTM-CNN

Figure 4.1: Confusion matrices for the three representative models where each row (true class) is scaled by its sum, for a visualization of classification recall. True and predicted labels are numbered in the order presented in tables 4.2-4.4.



(a) Accuracy curves (b) Loss curves

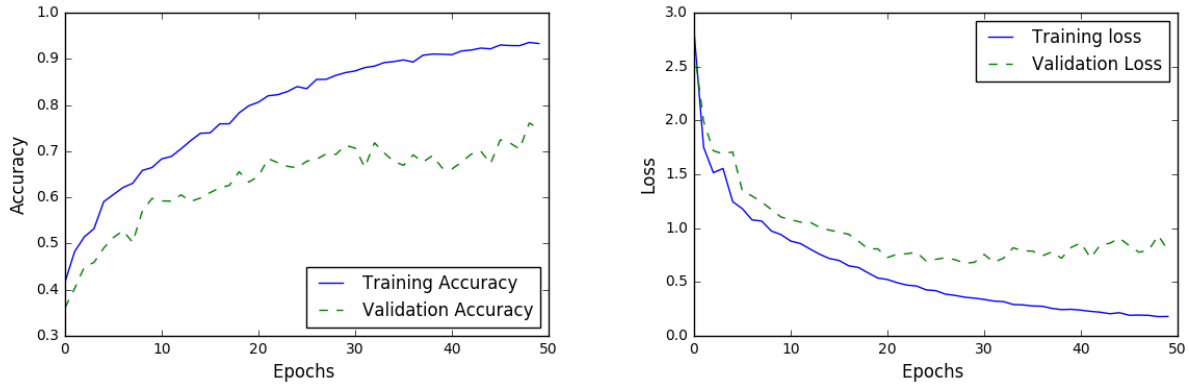
Figure 4.2: Deep residual CNN accuracy and loss curves: This model also shows a degree of overfitting as validation accuracy does not show appreciable increase after about 40-50 epochs, even as performance on training data continued to improve.

in figures 2.1c (occurring repeatedly) and 2.1f (in the single abnormal wave in the middle). The waveform similarity between the abnormal waves in bigeminy and PVC also explains why the residual network often confused the two. The convolutional structure recognizes the existence of similar patterns, but might not distinguish the number of times it occurs or its regularity, which is what distinguishes bigeminy from PVC.

We note that the overall F1 score of 0.803 achieved by the 2-layer BDLSTM exceeds cardiologist scores as reported in [2], both sequence level[1] F1 score of 0.719 and set level[2] F1 score of 0.751.

[1]Accuracy metric is the average second-level overlap between ground truth annotations and predictions

[2]For each sample, the set of identified rhythm classes present is compared (no penalty for time misalignment)



(a) Accuracy curves

(b) Loss curves

Figure 4.3: Combined BDLSTM-CNN accuracy and loss curves: Much like the deep residual network, this model also shows a degree of overfitting as performance on validation data does not improve after about 30 epochs. Validation loss started to increase slightly as well.

Although models utilizing residual structures showed the highest training accuracies (table 4.1), the relatively low validation and test accuracies, particularly for the combined BDLSTM-CNN model, seem to imply some degree of overfitting. This is evident in the loss/accuracy curves for these models (figures 4.2, 4.3). Experiments with hyperparameter adjustment and increasing regularization and dropout either did not show marked improvement in validation performance or led to underfitting.

We can also see that utilizing multi-channel data is beneficial (table 4.4). For the BDLSTM and combined models in particular, all classes show notable decrease in F1 score when the feature space for each timestep is reduced to 1. In the cases where single-channel data performed better (ventricular tachycardia and atrial fibrillation detection in the residual network), the differences are small, at 0.006 and 0.015 respectively. Compare this to the differences in F1 score demonstrated by the other models for the same classes at 0.213 and 0.467 respectively for the BDLSTM, and 0.264 and 0.062 respectively for the combined model (all decreases in scores upon reducing the feature space).

Chapter 5

Conclusions and Future Work

We show that using known deep network algorithms for classifying time-series data allows for accurate classification of normal, benign, and critical arrhythmias as well as distinguishing artifacts and noise from multi-channel ECG recordings. A layered bidirectional LSTM as well as a combined LSTM-CNN architecture is able to achieve relatively high accuracy and precision without the use of feature engineering or extraction of previously known waveform patterns.

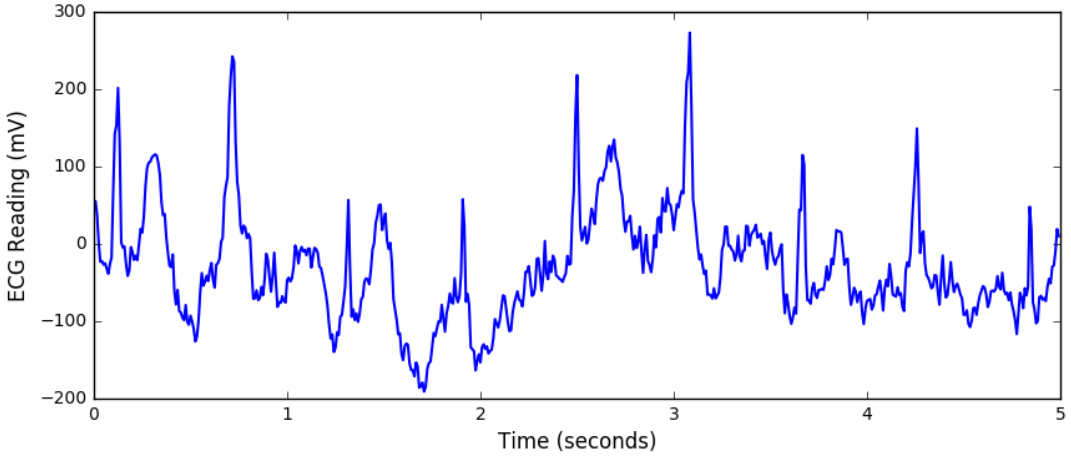
We also show that ECG classification greatly benefits from the use of multi-channel data, with nearly all classes and models showing markedly decreased accuracy when only one channel is used.

5.1 Limitations

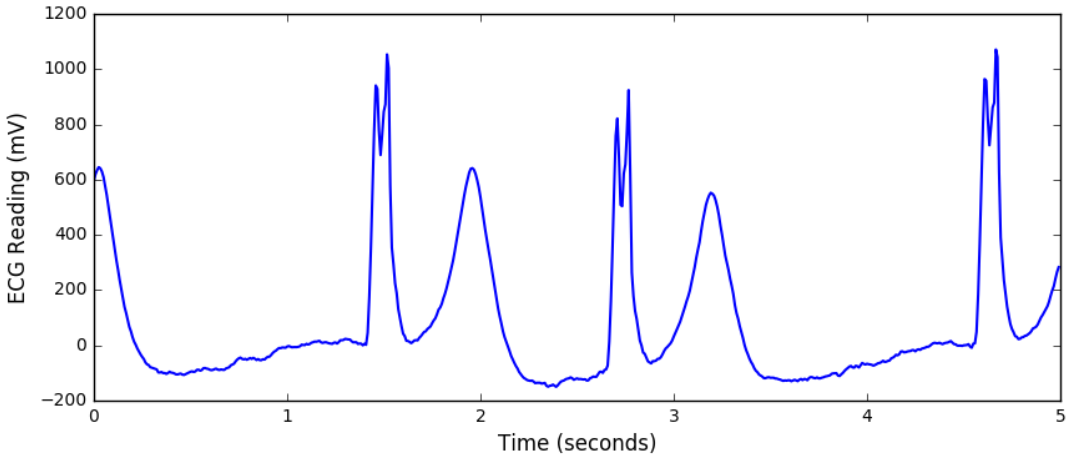
An inherent limitation for this particular problem is the difficulty in obtaining accurate ground truth labels. In the context of general machine learning, humans are typically the experts from whom ground truth labels or optimal actions are learned. However, ECG classification is inherently challenging even for practicing cardiologists; since we train the models under the assumption that cardiologist annotations reflect the ground truth, the models have an inherent limitation in performance.

In addition, a cursory review of the data implies a small proportion of the data labels is actually incorrect (figure 5.1). As it would be infeasible to manually check every available sample, this surely introduced additional error into our model. Some arrhythmias are morphologically very similar (figures 2.1e and 2.1f), and small differences in wave patterns can indicate different classes of arrhythmia—incorrect labels throw off the model's learned assumptions even further.

The low success rate of ventricular tachycardia classification is likely due to two primary factors: low support from the dataset, and that the few samples we do have are typically short (73% of the training set examples are only 2 seconds long).



(a) Labeled as **sinus**, despite clearly evident noise and lack of distinct sinus wave



(b) Labeled as **atrial fibrillation**, though this lacks the small but regular perturbations that are characteristic of fibrillation and shows large, unexplained waveforms

Figure 5.1: Examples of mislabeled data

5.2 Future Work

Past experiments with frequency domain transformation and custom feature extraction (such as P-QRS-T wave analysis [22] and clustering [23]), have shown to be useful in identifying specific arrhythmias; an ensemble model using both time series and these additional features will likely improve classification accuracy.

One area of interest is the optimization of these deeper models to perform online classification on a stream of ECG data, as existing measurement devices currently perform ECG interpretation in this way (which is necessary due to the time-critical nature of cardiac arrhythmia). Similarly, patient biometrics, which are typically available in real-life use cases of ECGs, can potentially be used in an integrated classifier. Patient-specific instances can be trained on a large, diverse dataset, and fine-tuned as they receive more readings from the patient. Underlying factors that can affect arrhythmia classification, such as resting heart rate, and existing or prior cardiac conditions, can also be incorporated.

References

1. Mant, J. *et al.* Accuracy of diagnosing atrial fibrillation on electrocardiogram by primary care practitioners and interpretative diagnostic software: analysis of data from screening for atrial fibrillation in the elderly (SAFE) trial. *BMJ* **335,** 380 (2007).
2. Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C. & Ng, A. Y. Cardiologist-level arrhythmia detection with convolutional neural networks (2017).
3. *ECGs for Beginners* (ed de Luna, A. B.) doi:10.1002/9781118821350. <https://doi.org/10.1002/9781118821350> (John Wiley & Sons, Ltd, Sept. 2014).
4. Drew, B. J. *et al.* Insights into the problem of alarm fatigue with physiologic monitor devices: a comprehensive observational study of consecutive intensive care unit patients. *PloS ONE* **9,** e110274 (2014).
5. Bortolan, G., Degani, R. & Willems, J. L. *Neural networks for ECG classification* in *Computers in Cardiology 1990, Proceedings.* (1990), 269–272.
6. Coast, D. A., Stern, R. M., Cano, G. G. & Briller, S. A. An approach to cardiac arrhythmia analysis using hidden Markov models. *IEEE Transactions on Biomedical Engineering* **37,** 826–836 (1990).
7. Cheng, W. & Chan, K. *Classification of electrocardiogram using hidden Markov models* in *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE* (1998), 143–146.
8. Schels, H. F., Haberl, R., Jilge, G., Steinbigler, P. & Steinbeck, G. Frequency analysis of the electrocardiogram with maximum entropy method for identification of patients with sustained ventricular tachycardia. *IEEE Transactions on Biomedical Engineering* **38,** 821–826 (1991).
9. Afonso, V. X. & Tompkins, W. J. Detecting ventricular fibrillation. *IEEE Engineering in Medicine and Biology Magazine* **14,** 152–159 (1995).
10. Zheng, Y., Liu, Q., Chen, E., Ge, Y. & Zhao, J. L. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science* **10,** 96–112 (2016).
11. Yang, J., Nguyen, M. N., San, P. P., Li, X. & Krishnaswamy, S. *Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.* in *IJCAI* (2015), 3995–4001.

12. Goldberger, A. L. *et al.* PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* **101,** e215–e220 (2000 (June 13)).

13. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Computation* **9,** 1735–1780 (1997).

14. Cui, Z., Ke, R. & Wang, Y. Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction (2018).

15. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.

16. Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **5,** 157–166 (1994).

17. He, K. & Sun, J. *Convolutional neural networks at constrained time cost* in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 5353–5360.

18. He, K., Zhang, X., Ren, S. & Sun, J. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification* in *Proceedings of the IEEE International Conference on Computer Vision* (2015), 1026–1034.

19. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015).

20. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2014).

21. Baldi, P. & Sadowski, P. J. *Understanding dropout* in *Advances in Neural Information Processing Systems* (2013), 2814–2822.

22. Karpagachelvi, S., Arthanari, M. & Sivakumar, M. ECG feature extraction techniques-a survey approach (2010).

23. Ye, C., Kumar, B. V. & Coimbra, M. T. Heartbeat classification using morphological and dynamic features of ECG signals. *IEEE Transactions on Biomedical Engineering* **59,** 2930–2941 (2012).