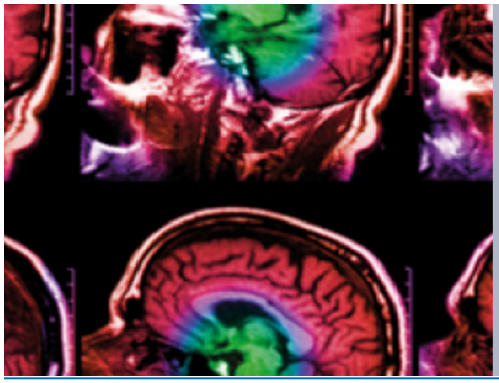
PAPER

A robust deep convolutional neural network for the classification of abnormal cardiac rhythm using single lead electrocardiograms of variable length



IOP Publishing Physiol. Meas. 39 (2018) 035006 (10pp) https://doi.org/10.1088/1361-6579/aaaa9d

Physiological Measurement

IPEM Institute of Physics and Engineering in Medicine

PAPER

RECEIVED
28 October 2017

REVISED
25 December 2017

ACCEPTED FOR PUBLICATION
25 January 2018

PUBLISHED
27 March 2018

A robust deep convolutional neural network for the classification of abnormal cardiac rhythm using single lead electrocardiograms of variable length

Rishikesan Kamaleswaran, Ruhi Mahajan and Oguz Akbilgic

UTHSC-ORNL Center for Biomedical Informatics, University of Tennessee Health Science Center, Memphis, TN, United States of America

E-mail: rkamales@uthsc.edu

Keywords: atrial fibrillation, deep learning, convolutional neural network, hyperparameter optimization, electrocardiogram

Supplementary material for this article is available online

Abstract

Objective: Atrial fibrillation (AF) is a major cause of hospitalization and death in the United States. Moreover, as the average age of individuals increases around the world, early detection and diagnosis of AF become even more pressing. In this paper, we introduce a novel deep learning architecture for the detection of normal sinus rhythm, AF, other abnormal rhythms, and noise. *Approach*: We have demonstrated through a systematic approach many hyperparameters, input sets, and optimization methods that yielded influence in both training time and performance accuracy. We have focused on these properties to identify an optimal 13-layer convolutional neural network (CNN) model which was trained on 8528 short single-lead ECG recordings and evaluated on a test dataset of 3658 recordings. *Main results*: The proposed CNN architecture achieved a state-of-the-art performance in identifying normal, AF and other rhythms with an average F_1-score of 0.83. *Significance*: We have presented a robust deep learning-based architecture that can identify abnormal cardiac rhythms using short single-lead ECG recordings. The proposed architecture is computationally fast and can also be used in real-time cardiac arrhythmia detection applications.

1. Introduction

Atrial fibrillation (AF) is a chronic arrhythmia associated with significant adverse outcomes, contributing to over 467 000 hospitalizations and 99 000 deaths annually (Go *et al* 2014). If left undetected, it can lead to blood clots, stroke, heart failure and other potentially deadly consequences (Banerjee *et al* 2011). It is estimated that between 2.7 million and 6.1 million American adults are affected by AF (January *et al* 2014). By 2050, it is projected that the prevalence of AF may exceed 15.9 million, with more than half the patients being aged 80 or older (Go *et al* 2001). The costs of managing AF continue to rise, with the expected costs of nonvalvular AF care to be $67.4 billion in 2012 (Tang *et al* 2014), and expenditure expected to continue rising over the next several decades.

The electrocardiogram (ECG) is the most commonly used measure for detecting AF. Specific characteristics are observed in the ECG that indicate the presence of AF, including irregular R–R intervals, the absence of P waves, and irregular atrial activity (January *et al* 2014). The current clinical practice involves manual interpretations of ECGs that are captured through a 12-lead ECG monitor (Harris *et al* 2012). However, the manual interpretation of an ECG can potentially be influenced by the presence of exogenous noise, including signal noise, nursing intervention at the bedside, movement, or improper lead placement. Endogenous noise, such as the influence of beta-adrenergic blocking agents, such as esmolol (Erdil *et al* 2009) or certain classes of opioid agonists, such as fentanyl, may also contribute to morphological changes in the ECG (Chang *et al* 2008). Additionally, challenges exist in the availability of a 12-lead ECG and trained personnel to interpret those findings (Cooke *et al* 2006). Therefore, automated methods of AF detection can address some of the challenges relating to interpretation while also potentially reducing the burden of AF detection in an increasingly aging population (Desteghe *et al* 2017).

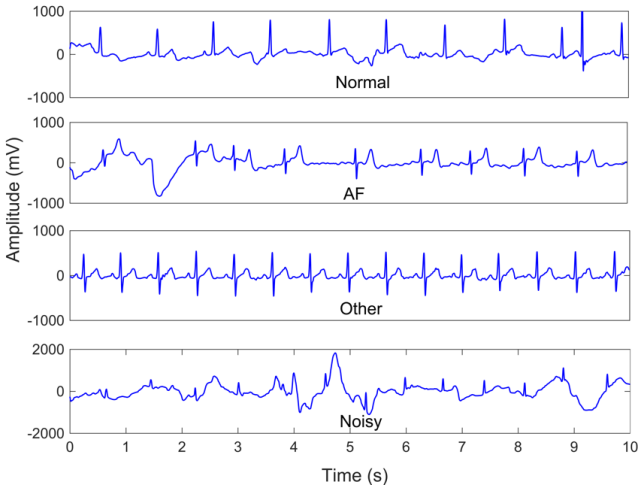


Figure 1. Example of ECG recordings representing normal, AF, other, and noisy rhythms.

There is a significant and growing literature on the automated detection of ECG morphologies such as the QRS complex and T peaks. The Pan-Thompkins algorithm is one of the most widely used algorithms in detecting the QRS complex based on the digital analysis of slope, amplitude, and width (Pan and Tompkins 1985). Yochum *et al* (2016) proposed a novel QRS, P, and T detection algorithm using continuous wavelet transformation. Their proposed algorithm has performed with very high accuracy when applied to some commonly used ECG databases published by *Computers in Cardiology* (Mark *et al* 1982). Moreover, Karimipour and Homaeinezhad (2014) proposed using discrete wavelet transformation for real-time QRS, P, and T detection. Their method has reached a specificity and sensitivity >.99 when applied to the same ECG datasets from the Computing in Cardiology 2011 Challenge (Silva *et al* 2011). In the proposed method, real-time signal preprocessing, which includes high frequency noise filtering and baseline wander reduction, is performed by applying a discrete wavelet transform (DWT).

Recently, Rajpurkar *et al* (2017) developed a 34-layer convolutional neural network (CNN)-based algorithm to detect a variety of heart arrhythmias from a single-lead ECG recording generated via a monitor. The authors showed that their CNN-based algorithm performed better than an average board-certified cardiologist in terms of sensitivity and precision. In other work, Rahhal *et al* (2016) proposed a novel deep learning approach for active classification of ECG recordings. These authors showed that their method outperforms other methods when applied to arrhythmia detection problems from publicly available databases. Furthermore, Pourbabaee *et al* (2016) applied feature learning with deep CNN for the screening of patients with paroxysmal atrial fibrillation. The authors showed that combining CNN extracted features with other classifiers increases the classification accuracy. Furthermore, Acharya *et al* (2017) addressed the CNN-based arrhythmias detection problem using different intervals of tachycardia ECG segments. Their 11-layer CNN model was highly accurate in detecting ECG segments using two and five second ECG segments.

In this paper, we present a method for the robust and rapid detection of AF and other abnormal cardiac rhythms using a single-lead ECG of variable length ranging between 9–61 s. We utilize CNN, a class of deep learning techniques developed originally for computer vision including image recognition. We present two experiments that use CNN to classify each ECG tracing into one of four discrete classes, namely AF, other rhythms, normal, and noisy. We also present a series of novel design considerations in the development of the CNN topology that were important factors that contributed to the state-of-the-art classification achieved on a hidden dataset hosted by the PhysioNet 2017 Challenge. We further discuss alternative methods that we evaluated but did not use in the final submission, that was evaluated against the hidden competition dataset. The paper is structured as follows: in this section we have discussed related work, which is followed by methods, results, discussion, and a summary with concluding thoughts.

2. Methods

2.1. Dataset

We have used a training dataset containing 8528 ECG recordings which were provided for the PhysioNet/Computing in Cardiology Challenge 2017, which aims to detect AF from a single short-lead ECG and broad taxonomy of rhythms. The recordings were collected through AliveCor's single channel (lead I) ECG device, which digitized the data in real time at 44.1 kHz. The digitized data were then stored at a sampling rate of 300 Hz with 16-bit resolution, and a bandwidth of between 0.5–40 Hz. The training set consisted of four categories of

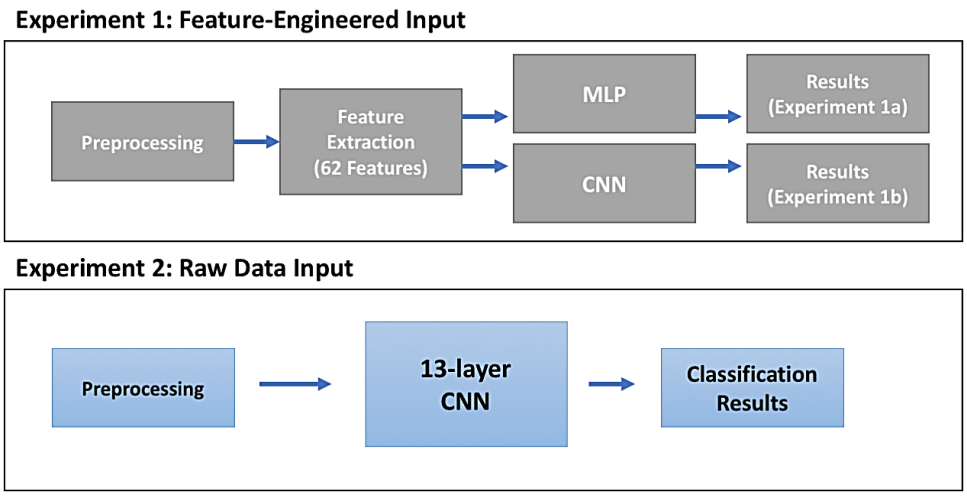


Figure 2. Block diagram of two experiments designed in this study for cardiac rhythm classification.

rhythms to classify, with the class distribution of 5050 normal, 738 AF, 2456 other, and 284 noisy. To evaluate the performance of the classification model, the challenge organizers used a hidden test dataset of 3658 recordings. The length of the ECG recordings in the training and test dataset varied from 9 s to 61 s. Figure 1 shows examples of 10 s excerpts of ECG recordings representing each class.

2.2. Experimental design and data preprocessing

In this paper, we present two experiments where we analyzed single-lead ECG data using deep CNNs (figure 2). The first experiment used feature input, while the second experiment used raw data as input. The second experiment explores the inherent feature extraction characteristics of the CNN technique. We hypothesized that the inherent convolutional method of the CNN would automatically identify salient features that can discriminate between the four cardiac rhythms without engineered features. In addition to the classical deep CNN method, other extensions were also considered, however, those approaches did not achieve sufficient results and are discussed briefly in the alternative approaches section 3.3. Our goal was to maximize the accuracy across all four classes; therefore, we use the F_1-score as our metric to determine the optimal model. The given training dataset was unbalanced and skewed towards the normal sinus rhythm class. We identified the optimal model which performed the best across the AF and other abnormal rhythm classes. Finally, as part of our model selection criteria, we imposed an early stopping criterion where the training ceases when the validation accuracy does not improve for at least 40 epochs. We saved the model weights if the validation accuracy improved and thereby used those weights for our testing.

A variety of data preprocessing steps were considered before we proceeded to the training phase. These steps were informed largely through our exploration or prior work in the area, including the use of discrete wavelet transforms, features generated from a series of linear, nonlinear, temporal and frequency based signal processing approaches, and raw data at various sampling frequencies (Mahajan *et al* 2017a). We also noted that in the raw data, some samples contained inverted signals, resulting from an inversion of the lead placements. We decided to keep inversions without correcting them to minimize our pre-processing of the data that may potentially increase our latency should the model be applied prospectively. The details of two designed experiments are given in the section below.

2.2.1. Experiment 1: feature engineered input to deep learning networks

In the first experiment, we extracted various time–frequency domain, complexity-based, and morphological features using a custom software in MATLAB. To do so, we firstly down sampled all ECG recordings at 200 Hz and then low-pass filtered data between 1–35 Hz to repress physiological and environmental artifacts. To extract temporal information, we used descriptive and heart rate variability metrics based on RR intervals, first order RR intervals, and second order RR intervals. Further, we evaluated the power spectrum of different rhythms (Kara 2007) and also computed sample entropy-based complexity measures (Alcaraz 2010). In total, we extracted a set of 62 features, described in detail in Mahajan *et al* (2017b), that can be fed to a machine-learning model for supervised classification of rhythms.

2.2.2. Experiment 2: raw data input to CNN

In this experiment, we analyzed raw data through the deep CNN by using the Biosignal Processing Library (Carreiras *et al* 2015) in Python. Specifically, we applied a digital 90th order finite impulse response bandpass

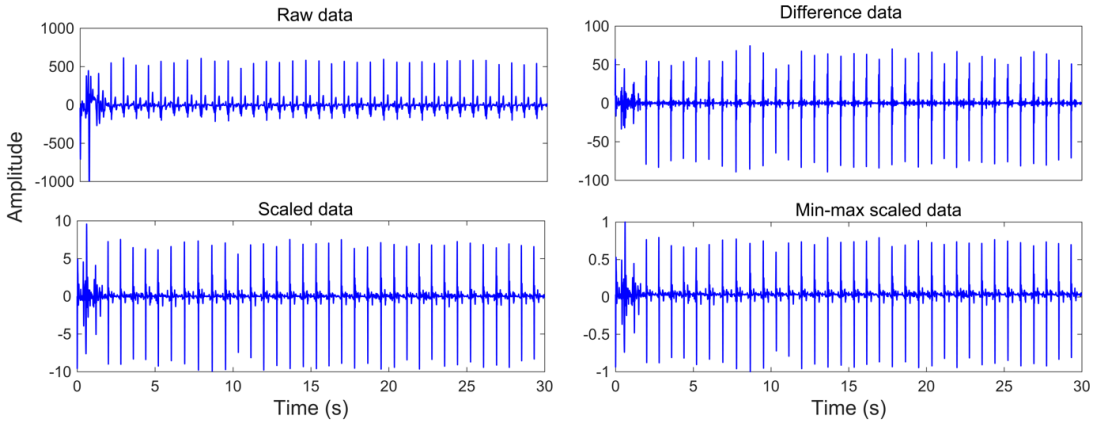


Figure 3. Data preprocessing steps: raw data is initially filtered (top left), then the first difference is taken of each consecutive sample (top right), the data is then scaled sample-wise (bottom left), and as an alternative we also investigated normalization.

Table 1. Formulations for different activation functions used in this study.

Activation function	Equation
Linear	$f(x) = x$
TanH	$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Softplus	$f(x) = \log_e(1 + e^x)$
Softmax	$f(x) = \frac{e^x}{\sum_{j=0}^{k} e^{x_j}}$
ReLU	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x > 0 \end{cases}$

filter to limit the signal bandwidth between 3–45 Hz and a Hamilton segmentation method (Hamilton 2002) for detection of the QRS complex. Following the filtering of the signal, we then took the first difference of each consecutive sample for input into a neural network. Figure 3 shows a representative raw and processed ECG recording from the normal class. The dataset contained variable lengths of recordings, some records were as short as 9 s while others were up to 61 s. In order to control for fixed dimensions as required by the CNN method, we tried repeating the segments of ECG up to a maximum of 18 286 samples. For instance, if a sample had a length of 10 s we would repeat it six times. We also tried zero-padding, in which we included original samples and inserted zeros where we did not have values, for a total of 18 286 samples. This ensured that the shape of our sample arrays was uniform across the entire training and test sets.

2.3. Hyperparameter search

We focused our efforts on the CNN and the traditional multilayer perceptron (MLP) method for classification. We began our investigation of the CNN method initially by performing a grid search of several hyperparameters. We began with an investigation of the minimal number of layers, followed by a search of various activation functions including: 'linear', 'tanH', 'sigmoid', 'softplus', 'softsign', 'softmax' and 'ReLU'. Table 1 shows the mathematical formulations for each of these activation functions.

We also investigated the use of various batch sizes, including 1, 5, 10, 20, 32, 64, and 128. We investigated the use of kernel initialization methods such as 'uniform', 'lecun uniform', 'normal', 'zero', 'glorot normal', 'glorot uniform', 'he normal', and 'he uniform'. We also searched across various epochs ranging from 10 to 2000. Due to the presence of an unbalanced dataset, we also investigated the efficacy of applying penalties to our optimizer by increasing the error when a misclassification is generated on atrial fibrillation or other abnormal rhythms. We further implemented a five-fold cross-validation technique and generated a unique model for each fold. All of these models are then combined within an ensemble with simple voting to determine the final classification. We also evaluate an alternative model without cross-validation using a simple train and test split of 70% and 30%, respectively.

2.4. Computing environment

We developed our CNN method initially on Tensorflow (Abadi *et al* 2016) and then ported our code to Keras (Chollet 2015) for rapid prototyping of various design considerations. We built our model on a desktop

Table 2. Effect of additional layers on the training and validation loss.

	Engineered features (62 total)			Raw ECG data		
Layers	Train loss	Val. loss	Epoch time (s)	Train loss	Val. loss	Epoch time (s)
3	0.4563	0.5644	6	0.9417	0.9205	114
6	0.4418	0.5862	10	0.6041	0.6220	122
9	0.5766	0.6096	14	0.4115	0.4836	130
12	0.6170	0.6302	18	0.4003	0.4215	136
15	0.6940	0.6875	21	0.5891	0.5442	141
18	0.6740	0.6433	25	0.7380	0.6664	145

computer with a Nvidia Titan X (Pascal series), consisting of 12 GB of GDDR5X memory and 3584 GPU cores. Both Tensorflow and Keras were executed in a Microsoft Windows environment using the Python language. We utilize Pandas (Pandas Community 2017), an open-source data analysis library, and Scikit-learn (Pedregosa *et al* 2011) for data processing and generating cross-validation sets. Initially, we evaluated a CPU implementation of our early models, but due to excessive training time in the CPU we decided to use GPU. All results presented in this paper are derived from our testing on the GPU.

3. Results

3.1. Experiment 1: feature-engineering based approach

In the first experiment, we extracted 62 features from a combination of descriptive, linear, nonlinear, temporal and spectral statistical methods and fed the resulting data into an MLP. The MLP architecture consisted of three hidden layers with 32 neurons in each layer, followed by batch normalization and ReLu activation. We achieved an accuracy of 76.79% in a test dataset consisting of 1706 samples. The F_1-score for each class was as follows: 0.84, 0.63, 0.63, and 0.55 for normal sinus rhythm (class 1), atrial fibrillation (class 2), other abnormal rhythm (class 3), and noisy signals (class 4), respectively. The average training time per epoch was 7 s on the GPU. The total number of CNN-generated parameters were 6788 and the final model was selected after 34 epochs. We noticed that this architecture was converging after 20 epochs, hence, to avoid overfitting, we had implemented an early stopping criterion, in which if the validation loss did not improve after 20 epochs the training would terminate.

We ran the same features through a three-layer 1D CNN architecture with a filter size of 256 in first, 128 in the second, and 64 in the last convolutional layer. We specified a stride of 12 in the first layer and maintained a stride of 5 for all other layers. Due to the limited input size, we did not include any pooling layers. Under those considerations, our CNN model achieved an accuracy of 77.49% and F_1-scores across each of the aforementioned classes as follows: 0.85, 0.71, 0.65, and 0.51. The average epoch computation time in this case was 9 s, total number of parameters were 210 372, and the final model was selected after 68 epochs. We subsequently increased the number of layers to a maximum of ten convolutional layers; however, neither the accuracy nor the F_1-scores showed any improvements (see table 2). Moreover, after 12 layers the receptive field of the convolutions became too small to allow a constant stride. We also investigated the recurrent neural network (RNN) using the same 62 features. Not only did the RNN significantly increase the training time, it also did not generate the accuracy of a shallow CNN. The RNN architecture resulted in an accuracy of 73.7%, with F_1-scores of 0.81, 0.56, 0.47, and 0.37 across class 1–4, respectively. As part of our feature-engineered approach, we utilized wavelet transforms that performed particularly well, hence we decided to further investigate with approximate wavelet coefficients as an input to the CNN. However, the wavelet feature itself did not generate meaningful predictive ability, hence we did not pursue it further.

3.2. Experiment 2: raw data input based approach

Apart from the feature-engineered inputs, we also investigated the use of raw waveforms at 300 Hz as inputs to our CNN model. To that end, we began our raw data experimentation by feeding the CNN with unfiltered ECG signals without any transformation. However, we noted that with preprocessing, especially scaling our data using Scikit-learn's scale function, significantly decreased the training time and improved accuracy. Interestingly, decreased training and improved accuracy were not observed when we attempted to use the MinMaxScaler (Pedregosa *et al* 2011). Additionally, we noted that taking the first difference of the filtered ECG using the BiosPPY library further decreased the training time and improved accuracy.

To further improve our model performance, we performed a hyperparameter search by investigating the influence of adding incremental convolutional layers see table 2. We identified that a batch size of ten performed optimally among our test set of 1, 5, 10, 20, 32, 64, and 128 sizes. While a larger batch size improved training time, it resulted in an undertrained model. An online learning option (batch of one) was excessive in training time and

Table 3. Classification performance of different activation functions in the convolutional layers of a multilayer network.

	Test accuracy (%)						
k-fold	Sigmoid	linear	ReLU	tanH	Softmax	Softplus	Softsign
1	82.21	77.67	83.82	77.09	84.41	83.38	70.35
2	75.60	77.22	83.88	72.67	83.88	84.10	77.73
3	81.01	78.96	83.28	77.27	85.63	79.25	75.44
4	83.87	77.43	84.46	76.76	84.24	83.14	70.45
5	76.01	77.74	83.49	73.88	83.05	81.00	73.59
Avg. time per epoch (s)	97	95	98	98	128	98	98
Mean accuracy	79.74	77.80	83.79	75.53	**84.24**	82.17	73.51
Standard deviation	3.736	0.678	0.449	2.114	**0.936**	2.002	3.197
95% confidence interval	76.078; 81.607	77.139; 78.143	83.345; 84.010	73.462; 76.591	**83.324; 84.710**	80.212; 83.174	70.378; 75.110

overfit towards the dominant normal class. We investigated epochs beginning with 10 and increasing to 2000 in a series of steps where we doubled epochs at each step. Shorter epochs greatly underfit, while any epoch over 350 was found to plateau. In our later evaluations, we diverged from imposing strict epochs and elected to use early stopping criteria, which monitored changes in our validation accuracies over a minimum number of epochs.

A search of activation functions is outlined in table 3. Each activation function was evaluated on a five-fold cross validation set. Notably, the Softmax activation achieved the highest validation accuracy in each of the five-fold cross validation models; however, it also had the highest average epoch runtime of 128 s. The second was ReLU, however, in this case the variation among the different folds was much smaller, as was the average epoch time. The worst performing activation function was Softsign, with an overall accuracy of 73.51% and significant variation among each of the evaluated folds. We performed both the parametric T-test and the non-parametric Mann–Whitney U-test to determine whether the F_1 scores obtained across different folds for each different activation function are significantly different to the F_1 scores obtained for ReLU. We found that there is no statistically significant difference between the F_1 scores obtained for ReLU, Sigmoid, Softmax, and Softplus ($p < 0.01$) while F_1 scores for Linear, tanH, and Softsign are significantly lower than the F_1 scores obtained for ReLU.

The traditional approach to implement convolutional layers is to impose a small filter size at the lower layers and increase the filter size iteratively to identify salient higher-level nonlinearities. We initially implemented such architectures; however, we found that a reversed approach, allowing for greater nonlinearity at the lower level and then rigorously constraining them in the higher levels, leads to a palpable increase in accuracy. Finally, in contrast to many implementations where convolutional layer outputs are received by a fully connected layer before the classification, we found that directing the convolutional layers to the output in lieu of a fully connected layer improved training time and accuracy.

3.3. Alternative deep convolutional neural network designs

Feeding raw QRS complex data abstracted from each of the ECG samples (a total of 110, 055 QRS complexes abstracted) generated an accuracy of 74.84% with an F_1-score of 0.84, 0.69, 0.60 and 0.07 in each of the classes 1–4 using a 20% test set, respectively. We then investigated a hybrid model for the raw QRS complex input, in which we maintained our existing CNN layer, and in place of a dense output layer, we implemented an LSTM neuron. This approach also significantly increased our training time, with the average epoch lasting 353 s on our GPU and a decrease in accuracy at 69.7%. Since none of these architectures resulted a significantly higher F_1-score than our experiment 2 architecture (refer table 3), we did not use these architectures in our final model. However, we report the accuracy statistics and confusion matrices in the supplementary material (stacks.iop.org/PM/39/035006/mmedia) for most of the alternative designs we experimented with.

3.4. Final model determination

Using results from the multiple experiments detailed above, we settled on a 13-layer 1D CNN architecture. In our architecture, we began with a large stride of 320 filters and aggressively reduced the filter size, as well as the stride in, at every successive batch of three layers. Based on earlier results mentioned above, we decided not to include a fully connected layer, and instead elected to pass the output of the convolutional layers directly to the output layer. We maintained ReLU against earlier results where 'softmax' activation showed seemingly better results (table 3), due to improved test accuracy on our hidden set. We also implemented a minor dropout of 0.3 and batch normalization.

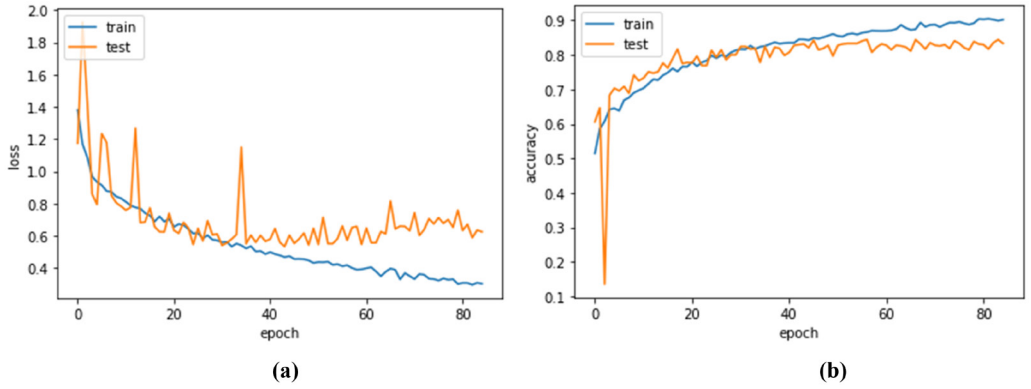


Figure 4. Results from training a 13-layer CNN architecture depicting (a) loss function and (b) changes in the validation accuracy over time.

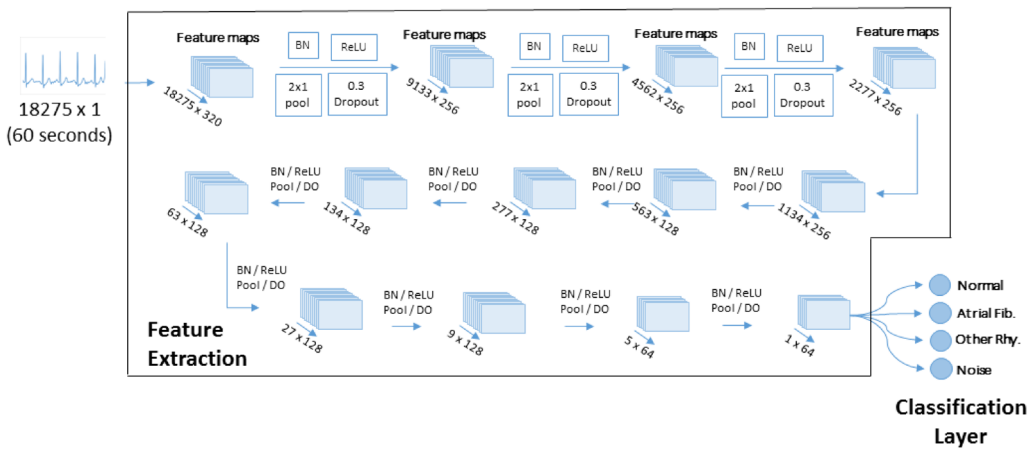


Figure 5. A deep convolutional neural network model used for classification of abnormal cardiac rhythms.

We noted that this architecture greatly reduced our training time, an accuracy in the mid-80s was generally achieved within 45 epochs, and the training generally terminated after 80 epochs, since we had implemented an early stop criterion of 40 epochs without a change in validation accuracy. The change in loss as depicted in figure 4(a). shows that after 40 epochs the train and test loss begin to diverge, with the test loss plateauing, indicating slight overfit in the classification. We also implemented a penalty function where incorrect prediction in class 2 and 3 would result in a greater loss. Figure 4(a) also illustrates the impact of the penalty, for instance, where there are several notable spikes in the loss curve, and figure 4(b) illustrates the corresponding changes to training and validation accuracy after each epoch. We continued to save the best weights throughout our training and used them for our final test, which we performed on a hidden test set generated from 20% of the original data.

The results from the test indicated an accuracy of 85.99%, with an average F_1-score of 0.83 in the first three classes, and individual F_1-score of 0.92, 0.80, 0.79 and 0.64 in classes 1 through 4, respectively. We then submitted this model as an unofficial entry to the PhysioNet 2017 challenge and achieved an averaged F_1-score of 0.84. The same model was selected for final evaluation, and the model received an averaged F_1-score of 0.83 across the first three classes. The individual F_1-scores were 0.9098, 0.8239, 0.7489, and 0.5191 across classes 1–4, respectively. Notably, class 3, the 'other abnormal rhythm', showed a marked decrease, possibly due to relabeling of the classes by the competition organizers. The pictorial representation of our final model is illustrated in figure 5. We also evaluated the test-time accuracy of our model on our local computing environment. The average time to generate one prediction in the test phase was 0.05 seconds.

3.5. Evaluation of duration of misclassified recordings

Due to the short length of given ECG recordings, it is challenging to model the underlying information required to discriminate different classes of rhythms. Nevertheless, the proposed model performs well in identifying normal and AF rhythms, but has a greater classification error rate for the 'other' and 'noisy rhythms' classifications. We evaluated the data length of misclassified recordings in the test set from each class (1003 normal, 162 AF, 486 other, and 55 noisy). It can be observed in figure 6 that as data length increases from 30 s, the percentage of

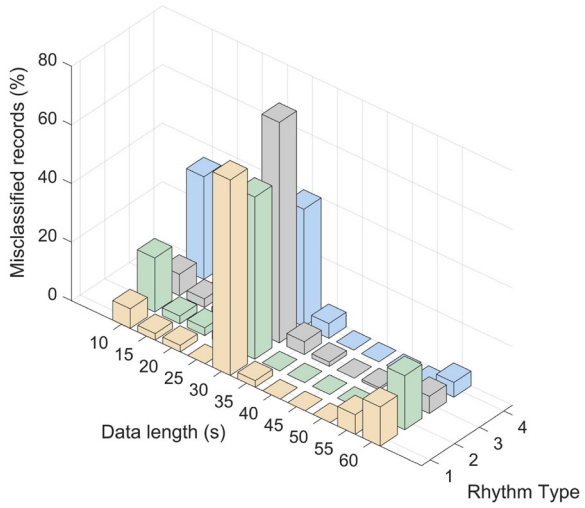


Figure 6. Percentage of misclassified recordings with respect to their data length for each rhythm. Rhythm type 1, 2 … 4 represents rhythms for normal, AF, other, and noisy classes.

misclassified records appear to decrease. As observed in figure 6, the percentage of misclassified labels for ECG recordings of length 30 s for normal, AF, other and noisy classes is 66.67%, 56.26%, 75.18%, 40%, respectively. We further applied an independent sample *t*-test which suggested that there is a significant difference between the length of correctly and incorrectly identified rhythms of only the 'other' class (p-value < 0.001). For the rest of the classes, no significant differences in the mean were observed (p-value > 0.05). This suggests that since the other class contains a broad taxonomy of rhythms, to identify some rhythms our model might need an ECG of longer duration for better classification.

4. Discussion

The results from across many experiments demonstrates the complexity and diversity in hyperparameter selection required to achieve the state-of-the-art performance in the classification of abnormal cardiac rhythms. From the beginning, we set out to identify the optimal model that not only provides the fastest test-time run time, but would also achieve similar accuracies to those achieved by alternative machine learning methods, such as random forests. As such, we include time as the measure of all our experiments, especially in tracking changes across various hyperparameters and its influence on epoch completion time.

Throughout the learning process, we identified some key practices, which allowed us to optimize the model while minimizing complexity. We found that in certain datasets, increasing layers also increased the validation loss (table 2). However, in other cases, increasing layers resulted in improvements to performance, but only up until a certain threshold, after which the performance decreased, as observed in the raw data analysis. Traditionally, filter allocations would begin with a small number of filters in the lower levels and then it would be incremented in each additional layer or batches of layers. We found that with raw ECG data, the reverse provided the best balance between training time and model performance. Allocating a large filter count at the low level and then rigorously decreasing in the higher levels improved training time while also providing improved performance.

We also noticed a relationship between the choice of activation function and accuracy across the four classes evaluated. Specifically, we noticed that Softmax achieved the highest accuracy during training validation across each of the five cross-validated methods (table 3); however, during the test it performed worse than ReLU. For multiclass classification problems, the usual practice is to use Softmax in the final classification layer, due to its ability to normalize each of the weights to unit probability. It is atypical to include Softmax in the intermediate layers due to loss of data that would occur across each of the individual layers. Our results suggest that even after the loss of such information, there appear to be strong classification features that were identified in the convolutional layers. We ultimately decided to select ReLU, due to the stability in accuracy that was observed both in the various cross-validated folds, but also during test in the 20% of original data we set aside. Moreover, we observed a significant difference between Softmax and ReLU in training and test time. As one of our objectives was to reduce test run time, ReLU proved to be the most acceptable alternative.

In the evaluation of small segments of ECG, we noticed that there was potential, especially for real-time applications. To generate short ECG samples, we identified 2.5 segments in each sample and applied the label uniformly across each segment. In the results of this analysis, we noticed that our model performed with high

accuracy even when provided just 2.5 s of ECG segments. This is an area of great potential for future work, especially in wearable devices when sampling might need to be balanced with battery and processor limitations. In our analysis, we did not aggregate results from each segment belonging to the original sample for a collective voting of classes, which may have further improved accuracy. Also, as illustrated in figure 6, most misclassifications in detecting other rhythms are for short 30 s ECG recordings. Notably, segments with 15–30 s ranges also performed well. We believe that if we train the model on longer ECG recordings, the overall classification performance might be improved.

For our final model, we selected a compact 13-layer 1D CNN that ingested raw full-segment data and generated classifications. We applied hyperparameters that were seen to be most influential in model performance from earlier experimentations across the feature engineered, wavelet and short-segment QRS peak analysis. We submitted the model for evaluation as part of the PhysioNet 2017 Challenge and achieved an average F_1-score of 0.83, which tied the official first place contributions using random forest, extreme gradient boosting, CNN, and recurrent neural networks.

In addition to deep CNN approaches, very deep CNNs have been previously demonstrated in specific image recognition scenarios. One such approach is the ResNet approach where shortcut connections exist across several layers to enhance the flow of information across many layers. In this study, we investigated the effects of those architectural designs. We constructed a 30-layer CNN with shortcuts occurring between every five layers, however our evaluation of the ResNet approach yielded poor results, in addition to increased training times, the model saturated quickly and achieved an accuracy of 69.46%. We decided not to pursue this architecture and continued to investigate compact architectures not exceeding 15 layers.

The limitation of our work is that it was performed on a dataset of ECG signals compiled from a single device. Results from our classification may be limited to signal characteristics of that device; future studies in other datasets are required to establish this method's generalizability. Secondly, we developed a model that would be efficient even in applications in the near real-time domain. The test-time to evaluate our model with a batch of one was an average of 0.5 s, which is well within the requirements for near real-time performance.

5. Conclusion

In this paper, we present numerous experiments on deep learning principles applied to a short-segment single-lead ECG waveform. We identified the best design practices that we believe can be applied to other research challenges with similar data. We have demonstrated the performance of the approach and ensured that the generated model would be efficient even in near real-time applications, such as for wearable devices or in the intensive care unit. In future work, we plan to apply these design principles to facilitate transfer learning in other domains where abnormal heart rate characteristics would need to be identified.

ORCID iDs

Rishikesan Kamaleswaran https://orcid.org/0000-0001-8366-4811
Ruhi Mahajan https://orcid.org/0000-0002-0284-7307
Oguz Akbilgic https://orcid.org/0000-0003-0313-9254

References

Abadi M *et al* 2016 Tensorflow: large-scale machine learning on heterogeneous distributed systems *12th USENIX Symposium on Operating Systems Design and Implementation (OSD 16) (Savannah, GA)* pp 265–83 (https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi)

Acharya U R, Fujita H, Lih O S, Hagiwara Y, Tan J H and Adam M 2017 Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network *Inf. Sci.* **405** 81–90

Alcaraz R R J 2010 A review on sample entropy applications for the non-invasive analysis of atrial fibrillation electrocardiograms *Biomed. Signal Process. Control* **5** 1–4

Banerjee A, Marín F and Lip G Y H 2011 A new landscape for stroke prevention in atrial fibrillation: focus on new anticoagulants, antiarrhythmic drugs, and devices *Stroke* **42** 3316–22

Carreiras C, Alves A P, Lourenço A, Canento F, Silva H and Fred A 2015 *BioSPPy: Biosignal Processing in Python* (https://github.com/PIA-Group/BioSPPy/)

Chang D J, Kweon T D, Nam S B, Lee J S, Shin C S, Park C H and Han D W 2008 Effects of fentanyl pretreatment on the QTc interval during propofol induction *Anaesthesia* **63** 1056–60

Chollet F 2015 Keras (https://github.com/fchollet/keras)

Cooke G, Doust J and Sanders S 2006 Is pulse palpation helpful in detecting atrial fibrillation? A systematic review *J. Fam. Pract.* **55** 130–4

Desteghe L, Raymaekers Z, Lutin M, Vijgen J, Dilling-Boer D, Koopman P, Schurmans J, Vanduynhoven P, Dendale P and Heidbuchel H 2017 Performance of handheld electrocardiogram devices to detect atrial fibrillation in a cardiology and geriatric ward setting *Europace* **19** 29–39

Erdil F, Demirbilek S, Begec Z, Ozturk E, But A and Ozcan Ersoy M 2009 The effect of esmolol on the QTc interval during induction of anaesthesia in patients with coronary artery disease *Anaesthesia* **64** 246–50

Go A S, Hylek E M, Phillips K A, Chang Y, Henault L E, Selby J V and Singer D E 2001 Prevalence of diagnosed atrial fibrillation in adults *JAMA* **285** 2370

Go A S *et al* 2014 Heart disease and stroke statistics—2014 update: a report from the American Heart Association *Circulation* **129** e28–e92

Hamilton P S 2002 Open source ECG analysis software documentation *Computers in Cardiology 2002* (IEEE) pp 101–4

Harris K, Edwards D and Mant J 2012 How can we best detect atrial fibrillation? *J. R. Coll. Phys. Edinburgh* **42** 5–22

January C T *et al* 2014 2014 AHA/ACC/HRS guideline for the management of patients with atrial fibrillation: a report of the American College of Cardiology/American Heart Association task force on practice guidelines and the Heart Rhythm Society *Circulation* **130** e199–267

Kara S O M 2007 Atrial fibrillation classification with artificial neural networks *Pattern Recognit.* **40** 2967–73

Karimipour A and Homaeinezhad M R 2014 Real-time electrocardiogram P-QRS-T detection-delineation algorithm based on quality-supported analysis of characteristic templates *Comput. Biol. Med.* **52** 153–65

Mahajan R, Kamaleswaran R and Akbilgic O 2017a Effects of varying sampling frequency on the analysis of continuous ECG data streams BT *Proc. Data Management and Analytics for Medicine and Healthcare: 3rd Int. Workshop, DMAH 2017 (Munich, Germany, 1 September 2017)* ed E Begoli *et al* (Cham: Springer) pp 73–87

Mahajan R, Kamaleswaran R, Howe A and Akbilgic O 2017b Cardiac Rhythm classification from a short single lead ECG recording via random forests *Comput. Cardiol.* **44** 1–4

Mark R G, Schluter P S, Moody G B, Devlin P H and Chernoff D 1982 An annotated ECG database for evaluating arrhythmia detectors *IEEE Trans. Biomed. Eng.* **29** 600

Pan J and Tompkins W J 1985 A real-time QRS detection algorithm *IEEE Trans. Biomed. Eng.* **32** 230–6

Pandas Community 2017 *Python Data Analysis Library* (O'Reilly Media)

Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30

Pourbabaee B, Roshtkhari M J and Khorasani K 2016 Feature leaning with deep Convolutional Neural Networks for screening patients with paroxysmal atrial fibrillation *Proc. Int. Joint Conf. on Neural Networks* pp 5057–64

Rahhal M M A, Bazi Y, Alhichri H, Alajlan N, Melgani F and Yager R R 2016 Deep learning approach for active classification of electrocardiogram signals *Inf. Sci.* **345** 340–54

Rajpurkar P, Hannun A Y, Haghpanahi M, Bourn C and Ng A Y 2017 Cardiologist-level arrhythmia detection with convolutional neural networks (arXiv:1707.01836)

Silva I, Moody G B and Celi L 2011 Improving the quality of ECGs collected using mobile phones: the PhysioNet/Computing in cardiology challenge 2011 *2011 Computing in Cardiology (CinC)* pp 273–6

Tang D H, Gilligan A M and Romero K 2014 Economic burden and disparities in healthcare resource use among adult patients with cardiac arrhythmia *Appl. Health Econ. Health Policy* **12** 59–71

Yochum M, Renaud C and Jacquir S 2016 Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT *Biomed. Signal Process. Control* **25** 46–52