MultiFusionNet: Atrial Fibrillation Detection With Deep Neural Networks

Luan Tran, B.S.^{1}, Yanfang Li, M.S.^{1}, Luciano Nocera, Ph.D.^{1}
Cyrus Shahabi, Ph.D.^{1}, Li Xiong, Ph.D.^{2}
^{1}University of Southern California, Los Angeles, CA, USA
^{2}Emory University, Atlanta, GA, USA

Abstract

Atrial fibrillation (AF) is the most common cardiac arrhythmia as well as a significant risk factor in heart failure and coronary artery disease. AF can be detected by using a short ECG recording. However, discriminating atrial fibrillation from normal sinus rhythm, other arrhythmia and strong noise, given a short ECG recording, is challenging. Towards this end, we propose MultiFusionNet, a deep learning network that uses a multiplicative fusion method to combine two deep neural networks trained on different sources of knowledge, i.e., extracted features and raw data. Thus, MultiFusionNet can exploit the relevant extracted features to improve upon the utilization of the deep learning model on the raw data. Our experiments show that this approach offers the most accurate AF classification and outperforms recently published algorithms that either use extracted features or raw data separately. Finally, we show that our multiplicative fusion method for combining the two sub-networks outperforms several other combining methods.

1 Introduction

Atrial Fibrillation (AF) is a critical health condition among cardiovascular diseases due to its prevalence, complexity and healthcare costs. Early diagnosis of AF is crucial as it can help prevent other diseases such as stroke [1]. Electrocardiogram (ECG) is the most prevalent and effective diagnostic tool for detecting cardiac arrhythmia. However, identifying whether there is AF in an ECG recording remains a challenging task since AF is asymptomatic [2] and may only appear briefly in the ECG signal.

In this paper, we are interested in detecting whether an ECG signal includes AF and distinguish it with other heart rhythm issues. This is orthogonal to detecting the starting and ending positions of AF in an ECG signal which is not in the scope of this study. Although using ECG is common practice, the actual AF detection is generally performed in a manual or semi-automated manner by cardiologists who visually inspect the signal for abnormalities. This process is time-consuming, requires an expert trained to interpret the signals, and is error-prone as signals may be misinterpreted or not seen. During AF, the P-waves in the ECG signal degenerate into a series of fibrillatory f-waves with small magnitudes [3], short and variable RR-intervals also appear [4]. P-waves, f-waves, and RR-intervals are described in more details in Section 3. Many AF detection methods based on the statistical information of P-waves, f-waves, and RR-intervals were developed as in [5, 6, 7, 8].

More recently, with the development of data-driven approaches in machine learning, some methods [9, 10, 11] for automated classification of normal/abnormal ECG signals have been proposed, which offer higher accuracy than previous approaches. However, these machine learning algorithms require relevant hand-crafted features as input which can be time-consuming to generate. The resulting performance heavily depends on the features and may not be robust. With the advance of GPU technology, deep neural networks can approximate very complicated learning function with a reasonable training time. Deep convolutional neural networks [12] that take raw data as input were proposed to automatically learn latent features. One challenge of AF detection using this approach is that it requires a large labeled ECG training dataset. To address these challenges, we develop a method to utilize both extracted features and raw data values for more effective and robust AF detection without relying on a large training dataset or hand-tuned feature selection benefiting from the best of the two worlds.

In particular, we propose a neural network **`MultiFusionNet`** using a multiplicative fusion method to combine two networks trained with two different sources of knowledge, i.e., the raw data and the relevant extracted features, to classify short single ECG signals. This approach utilizes the hand-crafted features to enhance the power of deep learning model taking raw data as input. Our experiments show that the proposed approach offers higher F1-scores and AUC_ROCs for AF classification than the recent algorithms using only one source of knowledge. We also show that our multiplicative fusion method to combine two sub-networks, the raw data sub-network and extracted feature sub-network, offers better classification performance than several other combining approaches.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we present the problem definition and our proposed solution. In Section 4, we report our experiments and the results. Finally, in Section 5, we conclude the paper.

2 Related Work

ECG classification is a problem that has been studied for decades. There are three main approaches to detect AF from an ECG signal which are listed as follows.

Identifying the absence of P-waves and the presence of f-waves. During AF, the P-waves in the ECG signal degenerate into a series of fibrillatory(f)-waves with small magnitudes [3]. Meanwhile, short and variable RR-intervals appear [4]. Therefore, the absence of P-waves, the appearance of f-waves [5, 6, 7, 8] and the irregularity in the RR-intervals [7, 13, 3, 14, 15] can be used to detect AF. Pan et al. [16] introduced a method to automatically extract RR-interval sequence. Chandra et al. [17] proposed a CNN [18] network to extract RR-interval sequence. Gawde et al. [19, 20, 21] proposed techniques to compute the features and classify ECG signals on GPU to reduce running time.

Non-linear classifiers using the extracted features. Many machine learning algorithms such as Random Forest [9, 10], Decision Tree Ensemble [11], Support Vector Machine [22], Linear Discriminant Analysis [23], Fully Connected Neural Network (FCNN) [24], Bivariate Gaussian Markov Model [19, 20, 21] focusing on streaming ECG data, were developed with the extracted features, e.g., RR-intervals statistics or statistics of the data transformed in the frequency space. This approach requires expertise in the field of ECG analysis, feature generation and selection process which can be time-consuming and error-prone.

Deep neural network using raw ECG signal. Deep neural network can make predictions directly from raw data. Hence the effectiveness of the learning process is increased when large datasets are available. Ghiasi et al. [25] combined the output of CNN network using RR-sequence and a feature based approach which compares each feature to an optimum threshold. Rajpurkar et al. [26] proposed a deep residual neural network for a multi-label classification task for ECG recordings. Pourbabaee et al. [12], Xiong et al. [27], Zihlmann et al. [28] proposed CNN neural networks to classify ECG recordings using raw data. Xiong et al. [29] proposed a recurrent convolutional network combined with CNNs and showed that it outperforms other recent traditional machine learning techniques. We implement our raw data sub-network by following this last approach.

Fusion Neural Network. Fusion neural networks which combine several sub-networks have been introduced in many data mining tasks such as traveling time estimation [30], action recognition [31], and object detection [32].

3 Method

3.1 Problem Formulation

We represent the discretized ECG signal as a raw data vector $X = [x_1, ..., x_k]$ and formalize the AF detection problem as a classification task that takes as input an ECG signal X and outputs its label y. We consider 4 possible labels: $L = [Normal, AF, Other, Noisy]$, where $Normal$ represents a normal ECG signal, AF represents an ECG signal that has AF, $Other$ represents an ECG signal that has other abnormal heart rhythms and $Noisy$ represents an ECG signal that is too noisy to be classified. Note that we are not concerned with detecting the actual position of AF within the ECG signal, we are only interested in detecting whether the ECG signal can be classified according to one of the labels.

3.2 MultiFusionNet

As observed from existing studies, traditional machine learning methods using extracted features are shown to be useful for ECG classification. However, their performance heavily depends on the correct feature extraction and feature selection process. On the other hand, deep recurrent convolutional networks show great promise in inferring latent features and temporal dependency of the ECG raw values that are not otherwise intuitive for feature engineering. However, the training of such networks require obtaining a large labeled dataset, which is a challenging task because ECG is part of private health record and the labeling process is time-consuming and error-prone. When trained on a small dataset, the accuracy of the networks may be less than accurate. To utilize both the extracted features and the raw data values, one approach is to use an ensemble of classifiers that are trained separately on the extracted features

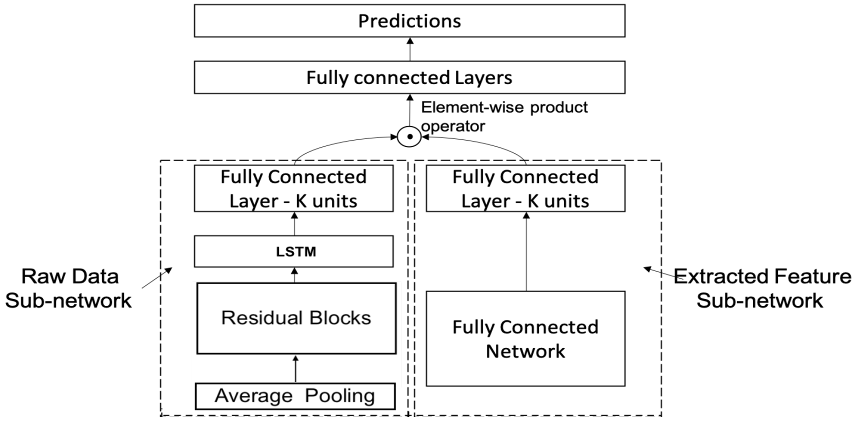

Predictions
Fully connected Layers
Element-wise product operator
Fully Connected Layer - K units
LSTM
Residual Blocks
Average Pooling
Fully Connected Layer - K units
Fully Connected Network
Raw Data Sub-network
Extracted Feature Sub-network

Figure 1: **`MultiFusionNet`** Architecture

and raw values. However, this approach cannot learn the importance of latent features from the raw data compared to the extracted features. Therefore, we have developed **`MultiFusionNet`**, a method to combine two sub-networks, one with raw data and the other one with the extracted features so that the combined network can be learned efficiently with a small training dataset and without using a careful feature selection process, benefiting from the best of the two approaches.

The structure of this network is shown in Figure 1. The extracted feature set is passed through a fully connected sub-network, since with more layers we can approximate more complicated function of the input data. The raw data is passed through a raw data sub-network to extract latent features and capture the temporal dependency. The extracted features are passed through the extracted feature sub-network. The output of the fully connected sub-network and the recurrent convolutional sub-network are merged by using a multiplicative fusion operator which will be described later. With this approach, we can simultaneously learn the parameters in both sub-networks. After merging, data is passed through fully connected layers. The output is a one-hot vector $y = [y_0, y_1, y_2, y_3]$ with $y_i = 1$ corresponding to the predicted label $l = L_i$, $L = [Normal, AF, Other, Noisy]$. After each fully connected layer, we apply a rectified linear activation [1]. To reduce over-fitting, in each fully connected layer, we apply penalties on layer parameters during optimization by using an L2 regularizer [2]. Because of the imbalance between classes in the training dataset, we use a weighted categorical cross entropy loss. Let w be the vector of weights for the classes, where a larger w_i value corresponds to a higher penalty applied to incorrect predictions for the label i. The weighted cross entropy loss function [3] is defined as follows:

$$CE = -\frac{1}{D}\sum_{j}^{D}\sum_{i}^{C} w_i t_{ji} \log p_{ji} \tag{1}$$

where D is the number of training examples, C is the number of classes, $t_{ji} = 1$ if example t_j has class i, $t_{ji} = 0$ otherwise and p_{ji} is the predicted probability that example j has class i.

3.3 Raw Data Sub-network

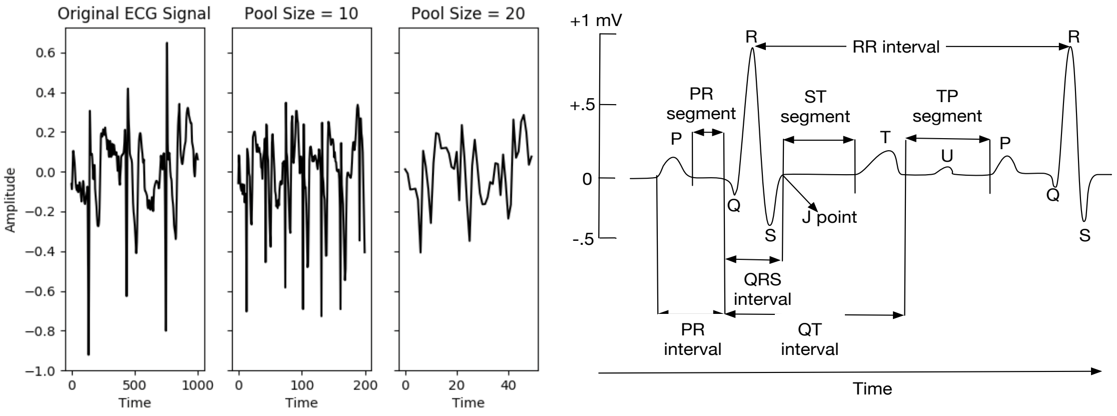
The raw data sub-network takes the raw-data as input. It contains residual blocks and LSTM layers to capture temporal dependency of the input data. Due to the long sequence of the inputs, the training time of CNN and LSTM layers can be too high, thus in the raw-data sub-network, we apply an Average Pooling layer to the raw data, which works as a down-sampling operator. Figure 2(a) shows an original ECG signal and its down-sampled signals after applying average pooling operator with the pool sizes of 10 and 20. This aggregate operator reduces the length of the input sequence with little information loss. We process the aggregated data in a sliding window fashion with windows of W values and slides of S values. Each sliding window is the input for the deep residual blocks. In each residual

[1] `https://keras.io/activations/`
[2] `http://faroit.com/keras-docs/1.2.2/regularizers/`
[3] `https://www.tensorflow.org/api_docs/python/tf/nn/weighted_cross_entropy_with_logits`

block, each convolutional layer feeds into the next layer and directly into the convolutional layers 2 hops away. The skip connections in the residual blocks help to increase the number of layers with less over-fitting. After the residual blocks, we use 3 recurrent layers to capture the temporal dependency between sliding windows.

3.4 Extracted Feature Sub-network

The extracted feature sub-network contains several fully connected layers and takes extracted features as input. We used the source codes provided by the PhysioNet/Computing in Cardiology Challenge 2017 [33] to identify RR sequence and then compute related RR-intervals features. In addition, we used the same statistical features as in [23], including Poincaré plot features, average beat, cross correlation, fiducial points, named intervals and their amplitudes composed of fiducial points, existence of P-wave and atypical ventricular morphologies.



(a) ECG signal aggregated at different pool sizes

(b) ECG signal named waves and related interval features

Figure 2: ECG signal with different pool sizes and related interval features

These ECG signal named waves and related interval features are shown in Figure 2(b) and defined in the following. ECG signals are composed of five waves named P, Q, R, S, and T. Based on the positions of these waves, one can derive PR, ST, TP segments, and PR, QRS, QT, RR intervals. RR-intervals features are extracted by using HRV (heart rate variability) analysis, average beat morphology analysis, and analysis of f-waves based on RR-interval sequences. The features are described as follows.

RR-intervals statistical features: The mean (RRmean), standard deviation, min, max, median, mean deviation, proportions of the standard and mean deviation from the mean value, ratio of the mean-to-median value of RR-intervals.

RR-intervals first-order difference statistical features: The mean value, min, max, median, standard deviation, median deviation and their proportions to RRmean, proportion of RR-intervals differing by $> 50ms$ from the preceding RR-interval, square root of the mean squared differences of successive RR-intervals (RMSSD) and its proportion to RRmean (RMSSD).

RR-intervals second-order difference statistical features: The mean value, min, max, standard deviation value of the RR-intervals second-order differences.

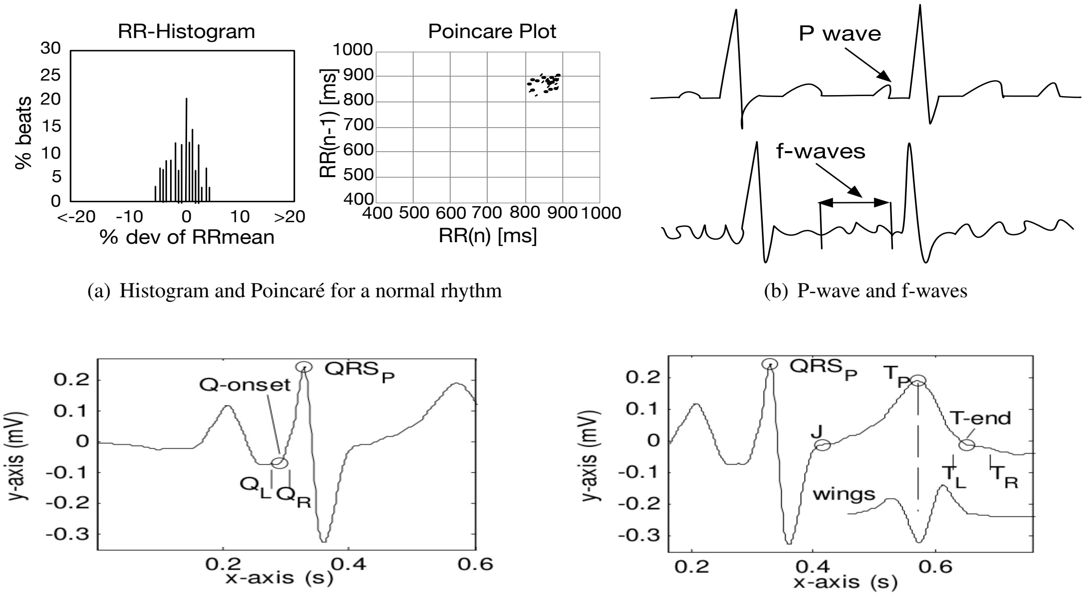
RR-Histogram: The ratio of total number of RR-intervals to the number of RR-intervals in the modal bin. HRV Triangular Index is used to report this feature at a re-sampling frequency of 128Hz, as shown in the left part of Figure 3(a) [34].

Poincaré Plot: The geometry of the Poincaré plot is represented by 3 features which are short-to-long term HRV, variability in the temporal structure (CCM), and the correlation coefficient respectively(corRR), as shown in the right part of Figure 3(a). When there is an AF in the signal, the Poincaré plot is much more scattered.

Average beat calculation: We choose the most sustained beats which are in the window of $[-300ms; 0.6 \cdot RRmean]$ around the QRS fiducial point and have peak-to-peak amplitudes within a standard deviation of the mean of all beats, and then signal averaging them to get a robust average beat.

Cross-correlation analysis: Maximal cross-correlation (corBeat) describes the relationship between the morphologies of all beats and the average beat, which can be used to show the effects of abnormal morphologies.

Detection of fiducial points (R, S, Q, J, T-end, Tpeak, Ppeak): As shown in Figure 2(b), R point has the biggest positive amplitude, while S point can be recognized with the most negative value. Q, J, T-end, T-peak is detected using the algorithm described in [35], as shown in Figure 3(c) and 3(d). P-peak has a peak in the window $[Q - 330ms; Q]$.
Calculation of intervals (QRS, PQ, QT): QRS=J-Q, PQ=Q-Ppeak, QT=Tpeak-Q.
Calculation of amplitudes (QRS, J, T, P): The amplitudes of QRS can be calculated by definition above. The amplitudes of J, T, P points are the values relative to the value of Q point.
Detection of P-wave: As shown in Figure 3(b), P-wave is successfully detected if the requirements with certain thresholds in [23] are satisfied. Otherwise, in AF, we may detect a series of fibrillatory(f)-waves with small magnitudes.
Detection of atypical ventricular morphologies: Atypical ventricular morphologies are the representative features of "Other" category. Four types of the abnormal morphologies are QRS fragmentation, Inverted QRS and T-wave, Left bundle branch block, and J-shift.
Calculation of curvature: $c = 1/r$ is used to evaluate the curvature. We calculate three important curvature ratios which are $maxc(QRS/P), maxc(QRS/T), maxc(T/P)$ based on the maximal curvature during P, QRS to evaluate the relative activity of different waves.



(a) Histogram and Poincaré for a normal rhythm

(b) P-wave and f-waves

(c) Q-Onset [35]

(d) T-End [35]

Figure 3: ECG signal named waves and related interval features

3.5 Multiplicative Fusion of Raw Data and Extracted Feature Sub-networks

In this section, we describe in more details how we merge the two sub-networks. As can be seen in Figure 1, after the recurrent layer, LSTM, (in the raw data sub-network), we apply a fully connected layer to flatten the data. A fully connected layer with the same size is applied after the fully connected network for the extracted features (the extracted feature sub-network). Let $A \in R^M, A = [a_1, a_2, ..., a_M]$ and $B \in R^N, B = [b_1, b_2, ..., b_N]$ be the feature vectors outputted from the last RNN layer with M units in the raw data sub-network and from the last fully connected layer with N units from the extracted feature sub-network, respectively. Let K be the number of units of the fully connected layer (in Figure 1) which are used to combine these two sub-networks. There are multiple options to combine two feature vectors, e.g., element-wise addition or concatenation. We choose an element-wise product operator [31] which was shown to be more effective at concurrently learning the feature importance of the features generated by the two sub-networks. The output of the element-wise product operation is as follows:

$$c_k = (\sum_{i=1}^{M} \alpha_{ki} a_i + \theta_k) \odot (\sum_{j=1}^{N} \beta_{kj} b_j + \delta_k), k \in [1, ..., K] \quad (2)$$

where $\odot$ is the element-wise product operator, α, β are weight matrices, θ, δ are the bias terms.

4 Experiments

4.1 Experimental Methodology

We compare **`MultiFusionNet`** with **`Random Forest`** [9, 10], **`Fully Connected Neural Network (FCNN)`** that take the extracted features as input and **`R-Resnet`** [29] with 21 convolutional layers that takes the raw data as input. **`Fully Connected Neural Network (FCNN)`** corresponds to a **`MultiFusionNet`** without the raw data sub-network. With FCNN, the computed features are passed through multiple fully connected layers. We used 5-fold cross-validation and the scores were averaged over all the folds. The algorithms were implemented in Python using Keras with Tensorflow backend. We first vary the pool size, network depth, and fusion operator to justify our choice of parameters. Then, we compare the F1-score and AUC_ROC of the methods. Finally, we compare the impact of the extracted feature and training data size on the methods. The experiments were executed on a Linux machine with 4 cores 2.7 GHz and 24GB memory.

4.1.1 Dataset

We use a dataset with short single lead ECG recordings [33], including 8528 ECGs which are sampled at 300 Hz, for scoring of 4 classes: Normal rhythm (*Normal*), Atrial fibrillation (*AF*), Other arrhythmia (*Other*), and Too noisy (*Noisy*). Each recording lasts from 9 seconds to 61 seconds. The statistics of the dataset are reported in Table 1.

Type	No. Recordings	Time Length (s)				
		Mean	SD	Max	Median	Min
Normal	5050	31.9	10.0	61.0	30	9.0
AF	738	31.6	12.5	60	30	10.0
Other	2557	34.1	11.8	60.9	30	9.1
Noisy	46	27.1	9.0	60	30	10.2
Total	**8528**	**32.5**	**10.9**	**61.0**	**30**	**9.0**

Table 1: Statistics of the dataset.

4.1.2 Parameter Setting

In the raw data sub-network, the Average Pooling layer has a pool size of 1, the CNN layer in each residual block has a kernel size of 15×1, the recurrent layer has 128 units. The window size W and slide size S of sliding windows are set to 20 and 1, respectively. The extracted feature sub-network contains one fully connected layer of 512 units. The fully connected layers that are the input of the element-wise product operator contains 512 units. After merging two sub-networks, data is passed through fully connected layers with 1000, 200, 50 and 4 units. In the weighted categorical cross-entropy loss function, we set the weight of class Normal, AF, Other and Noisy to be 1.0, 1.2, 1.2 and 1.0, respectively, according to unbalanced number of samples in each class. We use Adam optimizer, set the learning rate to be 0.001, the batch size to be 64. For Random Forest, we used 250 estimators.

4.1.3 Evaluation Metrics

In this study, we compare the F1-scores and AUC_ROCs of the algorithms. F1-score is the harmonic average of the precision and recall, and reaches its best value at 1 and worst value at 0. AUC_ROC represents the capability of distinguishing between classes. Higher the AUC_ROC, better the model is at prediction.

4.2 Experiment Results

Trade-off between Classification Performance and Training Time. Training the Raw data sub-network with a sequence of thousands of data points is time consuming. The average pooling layer is used for reducing training time with a small information loss. We vary the pool size from 1 to 20. When the pool size increases, the training time decreases but the information loss grows. Table 2 reports the F1-scores and AUC_ROCs of **MultiFusionNet** as we vary the pool size of the Average Pooling Layer. As observed from these tables, when the pool size increases, the F1-scores and AUC_ROCs of the **MultiFusionNet** decrease as there is more information loss. Figure 4 shows the training time for one epoch of **MultiFusionNet** in seconds as the pool size changes. When the pool size increases, the sequences going through the CNN and LSTM layers are shorter, therefore the running time decreases. The default value of pool size used in our following experiments is 1. When the pool size is set to 5, the training time is reduced significantly but the F1-score and AUC_ROC only decrease slightly.

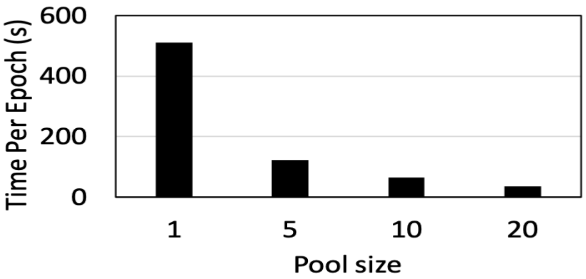


Figure 4: Training time per epoch for different pool sizes

Pool Size	F1-score				AUC_ROC			
	Normal	AF	Other	Noisy	Normal	AF	Other	Noisy
1	**0.9**	**0.83**	**0.75**	**0.69**	**0.87**	**0.9**	**0.82**	**0.88**
5	0.89	0.81	0.72	0.66	0.86	**0.9**	0.79	**0.88**
10	0.89	0.81	0.72	0.54	0.86	**0.9**	0.8	0.76
20	0.88	0.75	0.7	0.58	0.85	0.85	0.79	0.79

Table 2: F1-scores and AUC_ROCs of **MultiFusionNet** for varying pool Size

Impact of the Network Depth in Raw Data Sub-network. We vary the number of residual blocks b from 1 to 8 to examine the impact of the network depth to the classification performance. The network with a higher depth is capable of approximating a more complicated learning function. Because residual blocks are used for latent feature extraction, when the number of residual blocks increases, the performance of the model is supposed to be improved empirically. However, in practice, the model may be overfitted. Table 3 shows the F1-scores and AUC_ROCs of **MultiFusionNet** when b changes. As illustrated in this table, with $b = 3$, **MultiFusionNet** achieves the highest F1-scores and AUC_ROCs for most classes. Therefore, we set b to 3 in our other experiments.

Number of Residual Blocks	F1-score				AUC_ROC			
	Normal	AF	Other	Noisy	Normal	AF	Other	Noisy
1	0.89	0.81	0.68	0.62	0.84	**0.91**	0.76	0.81
3	**0.90**	**0.83**	**0.75**	**0.69**	**0.87**	0.90	**0.82**	**0.88**
8	0.88	0.79	0.69	0.64	0.83	0.90	0.75	0.78

Table 3: Impact of number of residual blocks in raw data sub-network

Fusion Operator Comparison. We compare our multiplicative fusion approach with some baseline fusion methods using element-wise add and concatenate operators. The element-wise add operator takes as input multiple vectors with the same size and returns a vector in which each value is the sum of the value in the corresponding column of the input vectors. The concatenate operator concatenates multiple vectors to return a vector with the length as the total length of the input vectors. Table 4 shows the F1-scores of all the classes with the 3 fusion methods. As illustrated in Table 4, multiplicative fusion method offers the highest F1-scores for all the classes. Concatenate and element-wise add operators perform very close to each other.

Fusion Method	Normal	AF	Other	Noisy
Element-wise Product (`MultiFusionNet`)	**0.90**	**0.83**	**0.75**	**0.69**
Element-wise Add	0.88	0.79	0.72	0.68
Concatenate	0.88	0.79	0.71	0.65

Table 4: Fusion method comparison - F1-score

F1-scores and AUC_ROCs Comparison. As depicted in Table 5, `MultiFusionNet` offers the highest F1-scores and AUC_ROCs for all the classes. `R-Resnet` has lower F1-scores and AUC_ROCs than those of the `MultiFusionNet` due to the difficulty of automatically learning the features from raw data with a small training dataset. `Random Forest` and `FCNN` offer similar F1-scores and AUC_ROCs and lower than those of `R-Resnet` and `MultiFusionNet` because they are trained with the same set of features but without the latent features that can be learnt by neural networks. For all the methods, the F1-scores and AUC_ROCs of the Normal class is the highest because majority of the ECG examples in the dataset are normal. Since noisy examples takes only 3% of the dataset, the F1-scores of the Noisy class is the lowest among all the classes.

Data	Method	F1-score				AUC_ROC			
		Normal	AF	Other	Noisy	Normal	AF	Other	Noisy
Raw Data	R-Resnet	0.88	0.79	0.73	0.63	0.84	**0.88**	0.78	0.73
Extracted Features	Random Forest	0.87	0.78	0.68	0.53	0.85	0.86	0.78	0.77
	FCNN	0.87	0.74	0.67	0.55	0.82	0.81	0.73	0.69
Raw Data and Extracted Features	`MultiFusionNet`	**0.9**	**0.83**	**0.75**	**0.69**	**0.86**	**0.88**	**0.81**	**0.81**

Table 5: F1-scores and AUC_ROCs Comparison

Confusion Matrix. Table 6 shows the confusion matrix of `MultiFusionNet` with 5-fold cross-validation. Normal samples are classified with the highest accuracy. Normal and AF samples are well distinguished. Small portions of AF samples are mistakenly classified as Other and Noisy. `MultiFusionNet` has difficulty in distinguishing Other and Noisy from Normal.

Predicted Actual	Normal	AF	Other	Noisy
Normal	**4585**	36	395	34
AF	30	**620**	80	8
Other	550	108	**1852**	46
Noisy	59	21	25	**179**

Table 6: Confusion Matrix of `MultiFusionNet`

Impact of Extracted Features. We compare the average F1-score and AUC-ROC of Random Forest (RF), FCNN, and `MultiFusionNet` when each feature is perturbed with a randomly generated noise within 10%, 20% and 30% of the feature value. Both RF and FCNN only use extracted features. As can be seen in Table 7, both RF and FCNN are quite sensitive to the noise while `MultiFusionNet` is fairly robust. When noise is increasing, the decrease in average F1-score and AUC_ROC of `MultiFusionNet` is significantly less than that of other methods. It shows that our proposed method does not heavily depend on hand-extracted feature input compared to other methods.

Impact of Training Data Size. We vary the training data size from 1000 to 6000 ECG series and report the average AUC_ROC and F1-score of all the methods. As reported in Table 8, with smaller training sizes, the F1-score and AUC_ROC of R-Resnet which only uses raw data as input drop significantly because R-Resnet does not learn latent features well from a small training dataset. In contrast, RF and FCNN, which only use extracted features, have smaller drops in F1-scores and AUC_ROC when reducing the training size. In all cases, our proposed method performs much better than the methods using only either raw data or the extracted features because our method utilizes the advantages of both data types.

Method	Average F1-score				Average AUC_ROC			
	Without Noise	Noise $\leq 10\%$	Noise $\leq 20\%$	Noise $\leq 30\%$	Without Noise	Noise $\leq 10\%$	Noise $\leq 20\%$	Noise $\leq 30\%$
Random Forest	0.74	0.50	0.40	0.32	0.81	0.67	0.62	0.58
FCNN	0.75	0.60	0.52	0.43	0.81	0.76	0.74	0.67
MultiFusionNet	**0.80**	**0.75**	**0.69**	**0.59**	**0.85**	**0.83**	**0.78**	**0.74**

Table 7: Average F1-score and AUC_ROC when perturbing feature inputs

Data	Method	Average F1-score			Average AUC_ROC		
		W = 1000	W = 3000	W = 6000	W = 1000	W = 3000	W = 6000
Raw Data	R-Resnet	0.25	0.48	0.75	0.58	0.69	0.82
Extracted Features	Random Forest	0.60	0.63	0.74	0.72	0.74	0.81
	FCNN	0.63	0.67	0.73	0.74	0.77	0.81
Raw Data and Extracted Features	MultiFusionNet	**0.70**	**0.74**	**0.80**	**0.77**	**0.82**	**0.85**

Table 8: Varying training data size

5 Conclusions

In this paper, we introduced a deep neural network to classify single ECG records. The network combines two sub-networks that take the raw data and the extracted features as inputs using a multiplicative fusion approach. The network architecture works well with a small training dataset. We showed that our approach outperforms the recent techniques which take as input the extracted features or raw data separately. We showed that our proposed method to combine the two sub-networks offers higher F1-scores and AUCs than the other approaches. With the detected AF signals, finding the starting and ending positions of AF can be explored in the future work.

References

1. Jason Andrade, Paul Khairy, Dobromir Dobrev, and Stanley Nattel. The clinical profile and pathophysiology of atrial fibrillation: relationships among clinical features, epidemiology, and mechanisms. *Circulation research*, 114(9):1453–1468, 2014.
2. Irina Savelieva and A John Camm. Silent atrial fibrillationanother pandora's box. *Pacing and Clinical Electrophysiology*, 23(2):145–148, 2000.
3. Chao Huang, Shuming Ye, Hang Chen, Dingli Li, Fangtian He, and Yuewen Tu. A novel method for detection of the transition between atrial fibrillation and sinus rhythm. *IEEE Transactions on Biomedical Engineering*, 58(4):1113–1119, 2011.
4. K Tateno and L Glass. Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of rr and δrr intervals. *Medical and Biological Engineering and Computing*, 39(6):664–671, 2001.
5. Helmut Pürerfellner, Evgeny Pokushalov, Shantanu Sarkar, Jodi Koehler, Ren Zhou, Lubos Urban, and Gerhard Hindricks. P-wave evidence as a method for improving algorithm to detect atrial fibrillation in insertable cardiac monitors. *Heart Rhythm*, 11(9):1575–1583, 2014.
6. Juan Ródenas, Manuel García, Raúl Alcaraz, and José Rieta. Wavelet entropy automatically detects episodes of atrial fibrillation from single-lead electrocardiograms. *Entropy*, 17(9):6179–6199, 2015.
7. R Alcaraz, C Vayá, R Cervigón, C Sánchez, and JJ Rieta. Wavelet sample entropy: A new approach to predict termination of atrial fibrillation. In *2006 Computers in Cardiology*, pages 597–600. IEEE, 2006.
8. Manuel García, Juan Ródenas, Raúl Alcaraz, and José J Rieta. Application of the relative wavelet energy to heart rate independent detection of atrial fibrillation. *computer methods and programs in biomedicine*, 131:157–168, 2016.
9. Morteza Zabihi, Ali Bahrami Rad, Aggelos K Katsaggelos, Serkan Kiranyaz, Susanna Narkilahti, and Moncef Gabbouj. Detection of atrial fibrillation in ecg hand-held devices using a random forest classifier. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
10. Ruhi Mahajan, Rishikesan Kamaleswaran, John Andrew Howe, and Oguz Akbilgic. Cardiac rhythm classification from a short single lead ecg recording via random forest. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
11. Guangyu Bin, Minggang Shao, Guanghong Bin, Jiao Huang, Dingchang Zheng, and Shuicai Wu. Detection of atrial fibrillation using decision tree ensemble. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
12. Bahareh Pourbabaee, Mehrsan Javan Roshtkhari, and Khashayar Khorasani. Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99):1–10, 2017.
13. Doug E Lake and J Randall Moorman. Accurate estimation of entropy in very short physiological time series: the prob-

lem of atrial fibrillation detection in implanted ventricular devices. *American Journal of Physiology-Heart and Circulatory Physiology*, 2010.

14. Deeptankar DeMazumder, Douglas E Lake, Alan Cheng, Travis J Moss, Eliseo Guallar, Robert G Weiss, Steven R Jones, Gordon F Tomaselli, and J Randall Moorman. Dynamic analysis of cardiac rhythms for discriminating atrial fibrillation from lethal ventricular arrhythmias. *Circulation: Arrhythmia and Electrophysiology*, 6(3):555–561, 2013.
15. Marta Carrara, Luca Carozzi, Travis J Moss, Marco De Pasquale, Sergio Cerutti, Manuela Ferrario, Douglas E Lake, and J Randall Moorman. Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy. *Physiological measurement*, 36(9):1873, 2015.
16. Jiapu Pan and Willis J Tompkins. A real-time qrs detection algorithm. *IEEE Trans. Biomed. Eng*, 32(3):230–236, 1985.
17. BS Chandra, Challa Subrahmanya Sastry, Soumya Jana, and S Patidar. Atrial fibrillation detection using convolutional neural networks. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
18. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
19. Purva R Gawde, Arvind K Bansal, and Jeffrey A Nielson. Integrating markov model and morphology analysis for finer classification of ventricular arrhythmia in real time. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 409–412. IEEE, 2017.
20. Purva R Gawde, Arvind K Bansal, Jeffrey A Nielson, and Javed I Khan. Bivariate markov model based analysis of ecg for accurate identification and classification of premature heartbeats and irregular beat-patterns. In *Proceedings of SAI Intelligent Systems Conference*, pages 265–285. Springer, 2018.
21. Purva R Gawde, Arvind K Bansal, and Jeffery A Nielson. Integrating markov model, bivariate gaussian distribution and gpu based parallelization for accurate real-time diagnosis of arrhythmia subclasses. In *Proceedings of the Future Technologies Conference*, pages 569–588. Springer, 2018.
22. Ricardo Couceiro, Jorge Henriques, Rui Pedro Paiva, Manuel Antunes, and Paulo Carvalho. Physiologically motivated detection of atrial fibrillation. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1278–1281. IEEE, 2017.
23. Ivaylo Christov, Vessela Krasteva, Iana Simova, Tatiana Neycheva, and Ramun Schmid. Multi-parametric analysis for atrial fibrillation classification in ecg. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
24. Ricardo Couceiro, Paulo Carvalho, Jorge Henriques, Manuel Antunes, Matthew Harris, and Jörg Habetha. Detection of atrial fibrillation using model-based ecg analysis. In *2008 19th International Conference on Pattern Recognition*, pages 1–5. IEEE, 2008.
25. Shadi Ghiasi, Mostafa Abdollahpur, Nasimalsadat Madani, Kamran Kiani, and Ali Ghaffari. Atrial fibrillation detection using feature based algorithm and deep convolutional neural network. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
26. Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.
27. Zhaohan Xiong, Martin K Stiles, and Jichao Zhao. Robust ecg signal classification for detection of atrial fibrillation using a novel neural network. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
28. Martin Zihlmann, Dmytro Perekrestenko, and Michael Tschannen. Convolutional recurrent neural networks for electrocardiogram classification. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
29. Zhaohan Xiong, Martyn P Nash, Elizabeth Cheng, Vadim V Fedorov, Martin K Stiles, and Jichao Zhao. Ecg signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network. *Physiological measurement*, 39(9):094006, 2018.
30. Zheng Wang, Kun Fu, and Jieping Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 858–866. ACM, 2018.
31. Eunbyung Park, Xufeng Han, Tamara L Berg, and Alexander C Berg. Combining multiple sources of knowledge in deep cnns for action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.
32. Xianzhi Du, Mostafa El-Khamy, Jungwon Lee, and Larry Davis. Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 953–961. IEEE, 2017.
33. Gari D Clifford, Chengyu Liu, Benjamin Moody, H Lehman Li-wei, Ikaro Silva, Qiao Li, AE Johnson, and Roger G Mark. Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.
34. SC Mirescu, MARIA Petrescu, FELICIA Petrescu, NICOLETA-CORNELIA Mirescu, and LEONTIN David. Challenges in implementing heart rate variability testing in a family medicine practice: Strengths pitfalls and caveats. *Studia Univ. Babes-Bolyai, Biol.*, 59(2):105–113, 2014.
35. II Christov and II Simova. Fully automated method for qt interval measurement in ecg. In *2006 Computers in Cardiology*, pages 321–324. IEEE, 2006.