Title of the work: Convolutional Neural Network Based Atrial Fibrillation Detection from ECG Signal

Summary of the work: Automatic atrial fibrillation (AF) detection is essential for preventing stroke due to silent heart diseases. In this paper, we propose an automatic AF detection by using electrocardiogram (ECG) signals and convolutional neural network. The proposed method is tested by using the ECG signals from Physionet. On the benchmark performance metrics, the proposed method achieved an average accuracy of 98.26\% for detecting AF events. The proposed method can achieve the AF event detection with a processing time of 0.77 $\pm$ 0.037 ms with the selection of optimal hyperparameters. The method has great potential in detection of AF events in ECG signal.

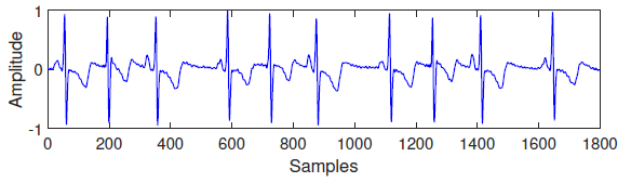

Figure 1: Normalized AF observed in ECG from record number 04015 of MITBIH-AF database.

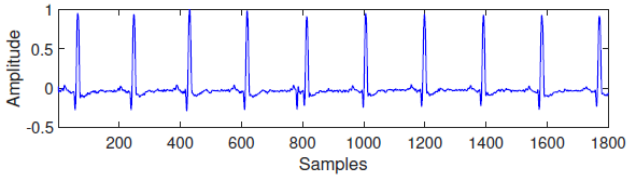

Figure 2: Normalized non-AF observed in ECG from record number 16273 of MITBIH-NSR database

Figure 3: Proposed block diagram for detecting AF and non AF

Table I: The proposed AF detection algorithm is presented as follows.

Initialization: Load AF signals, $s[n]$ (10 hr duration) with sampling frequency of 250 Hz from MITBIH-AF database.
Rule-01: Denoising the signal using Chebyshev high pass filter.
Rule-02: Comparing the signal with annotation file to label as AF and non-AF.
Rule-03: Dividing the signal into window of 1 min.
Rule-04: Find the number of AF segments and non-AF segments
Rule-05: The AF and non AF segments are counted.If dataset is unbalanced (AF: 11532 >> non-AF: 242), then dataset is to be balanced and following steps are followed:
Step-01: Load 1 sinus rhythm signals from MITBIH-NSR database.
Step-02: Resample the signals from both databases to 300 Hz.
Step-03: Divide the signals into 1 min segment.
Step-04: Normalize the MITBIH-NSR dataset containing the segments.
Step-05: Normalize the MITBIH-AF dataset containing the segments.
Rule-06: Create a mega dataset containg both the datasets.
Rule-07: Creating the 2-layer CNN model and train with 80% of the dataset. Also saving the .h5 files of the models for using with R-Pi.
Rule-08: Testing to detect AF and non-AF segments with 20% of the dataset.
Rule-09: Change the number of kernels, activation functions, and kernel sizes.
Rule-10: Calculating the processing time for each test segment.

TABLE II: Performance Comparison

Reference	Database	Method	Accuracy (%)	Processing Time
Kumar et al. [12]	MIT BIH AF	Random Forest	96.84	Not reported
Lira et al. [13]	MIT BIH AF, MIT BIH NSR	ResNet-50	99.7	Not reported
Our Method	MIT BIH AF, MIT BIH NSR	1D CNN	98.26	3.5 ms