Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 1192–1201

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Automated detection of cardiac arrhythmia using deep learning techniques

Swapna G*, Soman KP, Vinayakumar R

Centre for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita VishwaVidyapeetham, India

Abstract

Cardiac arrhythmia is a condition where heart beat is irregular. The goal of this paper is to apply deep learning techniques in the diagnosis of cardiac arrhythmia using ECG signals with minimal possible data pre-processing. We employ convolutional neural network (CNN), recurrent structures such as recurrent neural network (RNN), long short-term memory (LSTM) and gated recurrent unit (GRU) and hybrid of CNN and recurrent structures to automatically detect the abnormality. Unlike the conventional analysis methods, deep learning algorithms don't have feature extraction based analysis methods. The optimal parameters for deep learning techniques are chosen by conducting various trails of experiments. All trails of experiments are run for 1000 epochs with learning rate in the range [0.01-0.5]. We obtain five-fold cross validation accuracy of 0.834 in distinguishing normal and abnormal (cardiac arrhythmia) ECG with CNN-LSTM. Moreover, the accuracy obtained by other hybrid architectures of deep learning algorithms is comparable to the CNN-LSTM.

* Swapna G.
*E-mail address:*swapna.goutham@gmail.com

10.1016/j.procs.2018.05.034

Keywords: ECG; cardiac arrhythmia; deep learning; CNN; RNN; LSTM; GRU

1. Introduction

Cardiac arrhythmia is a condition where irregular heart rhythms occur. According to World Health Organization (WHO), about 17 million people in the world die every year due to cardiovascular diseases. This is about 31% of the total deaths globally. According to the statistics of American Heart Association (AHA), one out of every three deaths in US is related to cardiovascular diseases. The deaths due to cardiovascular diseases are more than due to all types of cancer and chronic lower respiratory diseases combined. A 2014 study indicates that approximately 2 to 3% of the people in North American and European countries are affected by atrial fibrillation.

A heart rate which is high (above 100 beats per minute in adults) is called tachycardia and a heart rate that is slow (below 60 beats per minute) is called bradycardia. If the beat is too early, then it is called premature contraction. Irregular beat is called fibrillation or flutter. Other than the criteria of heart rate, there are a number of other classifications for cardiac arrhythmia depending upon different types of criteria. Another type of classification is in terms of the site of origin of the irregular heart rate.

Atrial arrhythmias originate in the atrioventricular (AV) node. The AV node is positioned between the atria (each of the two upper cavities of the heart from which blood is passed to the ventricles is referred to as atria) and the ventricles. Atrial fibrillation (AF), atrial flutter, atrial tachycardia, premature atrial contractions and sinus bradycardia are some examples of atrial arrhythmias. Atrial fibrillation and atrial flutter are examples of arrhythmia which may lead to serious consequences. In AF, the atrium is contracted in a very fast and irregular manner with the heart's electrical signals originating from a different part of the atria or in the adjacent pulmonary veins instead of sino-atrial (SA) node. The walls of the atria fibrillate (quiver very fast) instead of beating in a normal way, making atria unable to pump blood properly into the ventricles. Stroke and heart failure are two complications to which atrial fibrillation can lead to. Conditions like high blood pressure, overactive thyroid gland, coronary and rheumatic heart diseases can lead to AF. Atrial flutter has similar symptoms and complications as AF. But in atrial flutter, the advancement of electrical signals of the heart through the atria happens in a fast and regular manner instead of the irregular manner in which it happens in AF.

Ventricular arrhythmias are premature rhythms occurring in an ectopic ventricular focus. Ventricular fibrillation, ventricular tachycardia, premature ventricular contractions are some examples of ventricular arrhythmias. Some arrhythmias are symptomless and not at all life threatening. But some symptomless arrhythmias can even lead to serious complications like blood clotting, stroke, heart failure and sudden cardiac death.

Arrhythmias occur when the electrical signals to the heart that co-ordinate heart beat are not working properly. The first step in the diagnosis of this abnormality is the analysis of electrocardiogram (ECG) and the confirmation that the ECG is not indicative of cardiac arrhythmia. ECG is a bio signal representing the activity of the autonomous nervous system (ANS) controlling heart rhythm. Thus, the electrical activity of the heart is recorded in ECG. It is a non-invasive and efficient tool to study cardiac rhythms and diagnose arrhythmias.

The ECG signal is generated as a result of the following processes. The heartbeat is originated as an electric pulse from the SA node situated in the right atrium of the heart. After contracting both atria, this electric pulse, then activates atrioventricular (AV) node that connects electrically the atria and the ventricles. This is followed by the activation of both ventricles. The complete heart activity is represented in the ECG waveform. Abnormalities in the morphology of ECG waveforms are indicators of cardiac arrhythmias. ECG waveform is analysed to ascertain the risk associated with any type of arrhythmia.

Extensive research has been done in the area of arrhythmia detection. The below are works in a serious type of arrhythmia called as myocardial infarction (MI) commonly known as heart attack. Data from a single lead ECG was used for MI detection achieving an accuracy of 94.74%[1]. Multiscale eigenspace analysis was carried out on 12 lead ECG data to achieve the same objective with an accuracy of 96%[2]. Analysis of 12 nonlinear parameters extracted from 12 lead ECG data using discrete wavelet transform (DWT) were used to detect MI to achieve an accuracy of 98.8%[3].

Deep learning techniques are now being increasingly employed in this area. The automated detection of normal and MI was conducted with CNN with an accuracy of 95.22%[4]. An accuracy of 84.54% was achieved in the detection of inferior MI in ECG using CNN[5]. Four types of arrhythmia were classified with an accuracy of 99.38% with MIT BIH data set along with another dataset as input[6]. Classification of MIT Arrhythmia database of ECG into normal and abnormal was conducted using artificial neural network (ANN) achieving an accuracy of 96.77%[7]. There are many works of classifying specific types of cardiac arrhythmia with ECG as normal input data. Often these specific cardiac arrhythmia cases addressed in most of the previous research work will be serious arrhythmia types like myocardial infarction. In short, researches were conducted into classifying normal ECG and many types of arrhythmia affected ECG.

Cardiac arrhythmia, though identified by the irregularity in cardiac rhythm, is due to the anomalies happening in the heart. These anomalies cause anatomical differences in the structure of atria and ventricles, thus producing changes in its activation, depolarization and repolarisation. These changes are reflected as deviation of ECG waveform from its normal shape and size. Different types of cardiac arrhythmia are caused by unique factors, thus causing unique changes in the morphology of the ECG wave[8], [9], [10].

The objective of this work is to develop an automated method for the diagnosis of cardiac arrhythmia. We perform a two class classification of the given ECG signal, whether cardiac arrhythmia is present or not. We use ECG recordings from the publically available MIT-BIH arrhythmia database in Physionet[11]. The MIT-BIH arrhythmia database is the first generally available dataset which is widely used for ascertaining the efficiency of cardiac arrhythmia detection algorithms. We employ deep learning based analysis methods using CNN, CNN-RNN, CNN-LSTM, CNN-GRU. Our work can be an assisting automated tool to cardiologists for the initial screening of people having cardiac arrhythmia.

The organisation of the paper is as follows: Section 2 presents brief descriptions of the deep learning techniques of CNN, CNN-RNN, CNN-LSTM and CNN-GRU used in this work. Section 3 tells about experiments conducted and results obtained. Section 4 includes discussions, conclusion and future work.

2. Background

This section explains about the various deep learning algorithms used in mathematical way.

2.1. Recurrent Neural Network (RNN)

It is an extension of feed forward network having feedback loops[12]. This results in a cyclic graph. These loops are the short-term memory used to store and retrieve past information over time scales. Temporal tasks can be executed very effectively with this improvement. Unlike multilayer perceptrons (MLPs), RNNs can handle temporal sequences of arbitrary length. RNN can also share its parameters across time-steps to avoid not being generalized on dealing with the unseen sequence of arbitrary length. In short, RNNs are models that can effectively learn dynamic temporal behaviours for input-output sequences of any arbitrary length. RNN is used extensively, especially for long standing AI tasks in the field of machine translation, language modelling and speech recognition.

The mathematical formulation of RNN is as follows:

$$H = R^d \times R^k \rightarrow R^k \tag{1}$$

(

$$H(x,h) = f\left(w_{xh}x + w_{hh}h + b\right) \tag{2}$$

$$w_{hx} \in R^{k\times d}, w_{hh} \in R^{k\times k} \quad \text{and} \quad b \in R^{k}$$

Here $\boldsymbol{x_t}$, $\boldsymbol{o_t}$ represent the input and output vectors. $\boldsymbol{w_{xh}}$, $\boldsymbol{w_{hh}}$, $\boldsymbol{w_{ho}}$ are weight matrices. f is the nonlinear activation function inbuilt in the hidden layer, particularly *sigmoid function* (σ) applied element wise. H is a short-term memory to the RNN's network.

2.2. Long short-term memory (LSTM)

LSTM is introduced as an advanced version of RNN by [13]. LSTM introduced memory blocks instead of conventional simple RNN units to handle the problem of vanishing and exploding gradient. LSTMs can handle long term dependencies much better than the traditional RNNs whereby LSTMs can remember and connect previous information that really lags back so much in time compared to the present.

A memory block in LSTM is a complex processing unit made up of one or many memory cells. Two multiplicative gates are included, one as input and the other as output gate. The input gate performs the allow or block task for an input flow of a cell activation to a memory cell. The output gate performs the allow or block task for an output state of a memory cell to other nodes. A set of adaptive multiplicative gates control the complete operations of a memory block. Forget gate[14] and peephole connections[15] are new additions in the architecture of the existing LSTM network with the advancement of research in that area. The forget gate is used instead of constant error carousel (CEC) (CEC is a self-recurrent connection with a fixed weight 1.0). The peephole connections are made from a memory cell to every gate. They learn the precise timing of the outputs as well as the internal state of a memory cell.

The working of the LSTM is as follows. The sequence of data of arbitrary length $\boldsymbol{x} = (x_1, x_2,....,x_{T-1}, x_T)$ is fed to the input of the LSTM architecture. The output sequence $\boldsymbol{o} = (o_1, o_2,....,o_{T-1}, o_T)$ with continuous write, read and reset operations by three multiplicative units (input (*in*), output (*ot*) and forget gate (*fr*)) on memory cell (*cl*) is found out iteratively from $t = 1$ to T in the hidden recurrent layer of LSTM. The below given set of equations represent the sequence of operations taking place in LSTMs at time step T. $(x_t, h_{t-1}, cl_{t-1} \rightarrow h_t, cl_t)$

$$in_t = \sigma\left(w_{xin}x_t + w_{hin}h_{t-1} + w_{clin}cl_{t-1} + b_{in}\right) \tag{3}$$

$$fr_t = \sigma\left(w_{xfr}x_t + w_{hfr}h_{t-1} + w_{clfr}cl_{t-1} + b_{fr}\right) \tag{4}$$

$$cl_t = fr_t \odot cl_{t-1} + in_t \odot \tanh(w_{xcl}x_t + w_{hcl}h_{t-1} + b_{cl}) \tag{5}$$

$$ot_t = \sigma\left(w_{xot}x_t + w_{hot}h_{t-1} + w_{clot}cl_t + b_{ot}\right) \tag{6}$$

$$h_t = ot_t \odot \tanh(cl_t) \tag{7}$$

The memory cell stores the information across many time steps with the help of the three adaptive multiplicative gating units. Input and output gates modulate the input and output flow of a cell activation of a memory cell. The task of the forget gate is to reset the self-recurrent value, when it becomes irrelevant. Forget gate uses the value 0 to delete and 1 to keep value to next step by multiplying with a memory cell. Memory cell and all gates have peephole connections which are used for the learning of the precise timings of outputs.

2.3. Gated Recurrent Unit (GRU)

GRU is a modified version of LSTM with lesser number of parameters[16]. The computational cost and memory consumption is much less compared to LSTM. The processes taking place in GRU can be represented by the following equations:

$$i - f_t = \sigma\left(w_{FMi-f} FM_t + w_{h-f} h_{t-1} + b_{i-f}\right) \text{(update gate)} \tag{8}$$

$$f_t = \sigma\left(w_{FMf} FM_t + w_{hf} h_{t-1} + b_f\right) \text{ (forget or reset gate)} \tag{9}$$

$$cl_t = \tanh(w_{FMc} FM_t + w_{hc}(f \odot h_{t-1}) + b_c) \text{ (current memory)} \tag{10}$$

$$h_t = f \odot h_{t-1} + (1-f) \odot cl \quad \text{(updated memory)} \tag{11}$$

where *FM* denotes the new feature vectors that are computed from CNN. The above equations are written viewing the GRU as a part of the hybrid network CNN-GRU where the GRU receives input from the output of the CNN.

2.4. Convolutional Neural Network (CNN)

CNN is a peculiar type of MLP. CNN is build up of three main layers. They are named as convolutional layer, pooling layer and a fully connected layer. CNN, in our case, takes the time series input data in one dimensional form where the data are arranged in the order of sequential time instants. The CNN may also include a non-linear activation function such as rectified linear units (ReLUs).

The CNN has the architecture of INPUT-CONV-POOL-FC. The primary building block of CNN is the convolution layer (CONV). In our case, the input one dimensional data vector is represented as $\boldsymbol{x} = (x_1, x_2, \ldots, x_{n-1}, x_n, cl)$ where $x_n \in R^d$ denotes features (here time series normal and abnormal (arrhythmia stricken) ECG data) and $cl \in R$ denotes a class label (either normal or abnormal). Convolution1D constructs a feature map *fm* by applying the convolution operation on the input data with a filter $w \in R^{fd}$ where *f* denotes the features inherent in the input data producing at its output, a new set of features which is fed as input to the next block in line. A new feature map *fm* from a set of features *f* is obtained as follows:

$$hl_i^{fm} = \tanh\left(w^{fm} x_{i:i+f-1} + b\right) \tag{12}$$

The filter *hl* is employed to each set of features *f* in the input data represented by $(x_{1:f}, x_{2:f+1}, \ldots\ldots, x_{n-f+1})$ to generate a feature map as $hl = (hl_1, hl_2, \ldots\ldots hl_{n-f+1})$ where $b \in R$ denotes a bias term and $hl \in R^{n-f+1}$.

The output of the convolutional layer is given to the pooling (POOL) layer. Convolutional layer contains *ReLU* activation function that use *max(0,X)* to each of the inputs to the *ReLU* represented by *X*. The pooling layer (POOL) performs a down sampling operation. Here, the max-pooling operation is applied on each feature map as $hl = \max\{hl\}$. This obtains the most significant features, meaning selection of features with highest values. These selected features are fed to fully connected layer, containing the *softmax* function that gives the probability distribution over each class. Thus, the fully connected layer (FC) will compute the class which is the final output obtained of the CNN.

2.5. Hybrid networks of CNN-RNN, CNN-LSTM, CNN-GRU

In hybrid networks, CNN consists of convolution1D and maxpooling1D layers only. The output of the maxpooling1D layer is fed to the input layer of the subsequent network.

$$y_i = CNN(x_i) \tag{13}$$

$\boldsymbol{x_i}$ is the initial input vector to the CNN network with class label attached to each type of data. $\boldsymbol{y_i}$ is the output of the CNN network. It is the feature vector formed from the max-pooling operation in CNN. This is fed to the next deep learning architecture we use. In this work, we are using hybrid networks of CNN-RNN, CNN-LSTM, CNN-GRU and a comparison is done on their performances.

3. Experiments and Results

For all our experiments, we used Keras[17] and TensorFlow[18] as backend with graphics processing unit (GPU) enabled deep learning framework in Ubuntu 14.04 operating system (OS). Deep learning algorithms contain several parameters like number of units and learning rate. The performance of deep learning algorithms relies heavily on choosing optimal values for these parameters. We conducted various trails of experiments to identify suitable value for these parameters. Backpropagation through time (BPTT)[19] approach is employed for the training of the deep learning networks.

3.1. Data used

We use MIT-BIH arrhythmia database in the Physionet[11]. The database is comprised of 48 half-hour excerpts of two-channel ambulatory ECG recordings. These ECG recordings are collected from 47 people at the BIH Arrhythmia laboratory between years 1975 and 1979. The recordings were digitized at a sampling frequency of 360 samples per second per channel with 11-bit resolution over a 10 mV range.

Generally, most algorithms use the MIT-BIH database in the original format as the same data recorded in 48 sequences (101.dat to 124.dat, 200.dat to 234.dat (total 48 sequences with some intermediate numbers absent)). This database has a major issue when applied to deep learning networks in the original format. It is due to the fact that these 48 sequences largely consist of two types of data (either related to normal or abnormal heartbeats). Due to this, beat-to-beat dependence is very much possible in the data sequences. Another issue is the difference in baseline voltages of different sequences. In order to tackle both these issues, we extracted individual heartbeats from continuous sequences of database. These extracted heartbeats are used to train our deep learning networks.

The extraction of the heartbeat from the original sequence is done as follows: A series of samples centred on the R peak of the heartbeat is extracted. Each heartbeat is annotated at its R peak. So, to extract a heartbeat, we need to fix only one parameter, i.e. the size of the window used for the extraction purpose. We took the average distance between the two R peaks to be 0.75 seconds. The window size was chosen as 1 second to allow for error margin. In this way, individual heartbeats were extracted from all the 48 sequences of data. The extraction of these heartbeats is the only pre-processing we performed on MIT-BIH arrhythmia database.

We obtained 361 samples for most of the examples since the window size chosen is 1 second. Some examples contained 360 samples where we duplicated the last sample in order to make the total sample values as 361 in all examples. We removed examples which carried an annotation different from normal or some type of arrhythmia. This is because the aim of this work is to classify the given ECG data as belonging to either normal or abnormal (arrhythmia) category. After the removal, we are left with 93521 examples out of which 18482 (19.7%) indicated arrhythmia.

3.2. Hyper parameter selection

To find suitable parameter for RNN, LSTM and GRU, we used moderately sized architectures with one hidden layer, 32, 64 and 128 units in RNN and 32, 64 and 128 memory blocks in LSTM and GRU. Three trails of experiment are done for each parameters related to units/memory blocks. Each experiment is run till 300 epochs. 64 units/memory blocks have shown highest accuracy in five-fold cross-validation for most of the deep learning architectures. By considering the training cost, we decided to use 64 units/memory blocks for the rest of the experiments. We run two trails of experiment with a CNN and pooling layer including number of filters 32, 64, 128 and filter length 2, 3 and 5. CNN with number of filters 64 and filter length 3 has attained highest accuracy in five-fold cross-validation. For the rest of the CNN experiments, we decided to use these newly found parameters.

In order to find an optimal learning rate, we run two trails of experiment for all deep learning architectures till 500 epochs are reached, with learning rate varying in the range [0.01-0.5]. The average five-fold cross validation accuracies corresponding to these deep learning architectures are considered. Most of the deep learning architectures showed sudden peak in their accuracy value at the learning rate of 0.1 and after that the accuracy value showed fluctuations and finally even a decrease was observed. For CNN and hybrid network, there was a sudden decrease in accuracy at learning rate 0.2 and finally attained highest accuracy at learning rates of 0.40, 0.45 and 0.5 in comparison to learning rate 0.1. This accuracy may have been enhanced by running the experiments till 1000 epochs. As the complex architectures we have experimented with, showed less performance within 500 epochs, we decided to use 0.1 as learning rate for the rest of the experiments for all deep learning architectures after considering the factors of training time and computational cost.

3.3. Identifying network structure

We used the following network topologies in order to find an optimum architecture for our input data.

- RNN/LSTM/GRU 1 layer
- RNN/LSTM/GRU 2 layer
- RNN/LSTM/GRU 3 layer
- CNN 1 layer
- CNN 2 layer
- CNN 3 layer
- CNN 1 layer with RNN/LSTM/GRU
- CNN 2 layer with RNN/LSTM/GRU
- CNN 3 layer with RNN/LSTM/GRU

For all the above network topologies, we run 3 trails of experiments. Each trail of experiment was run till 500 epochs. It is observed that most of the deep learning architectures learn the normal category patterns of input data within 250 epochs. The number of epochs required to learn the abnormal category data usually varies. The complex architecture networks required large number of iterations in order to reach the desired high accuracy. Finally, we obtained the best performed network topology in each architecture. The detailed result is displayed in Table 1.

3.4. Proposed architecture

The proposed deep learning architecture for the classification of ECG recordings into either normal or arrhythmia is presented in Fig. 1. Deep learning algorithms don't need explicit feature extraction and analysis like traditional machine learning based classifiers. It just passes raw input data to more than one hidden layer to obtain the optimal feature representation by itself. The newly formed feature representations are further passed as input to the fully-connected layer (dense layer) which uses *sigmoid* activation function to produce output binary values 0 or 1 indicative of arrhythmia or normal ECG.

Table 1. five-fold cross validation accuracy.

Architecture	five-fold cross validation accuracy
RNN 1 layer	0.617
LSTM 1 layer	0.631
GRU 1 layer	0.628
RNN 2 layer	0.652
LSTM 2 layer	0.683
GRU 2 layer	0.672
RNN 3 layer	0.731
LSTM 3 layer	0.774
GRU 3 layer	0.776
CNN 1 layer	0.715
CNN 2 layer	0.732
CNN 3 layer	0.781
CNN 1 layer with RNN	0.728
CNN 1 layer with LSTM	0.749
CNN 1 layer with GRU	0.747
CNN 2 layer with RNN	0.741
CNN 2 layer with LSTM	0.772
CNN 2 layer with GRU	0.779
CNN 3 layer with RNN	0.812
CNN 3 layer with LSTM	0.834
CNN 3 layer with GRU	0.837

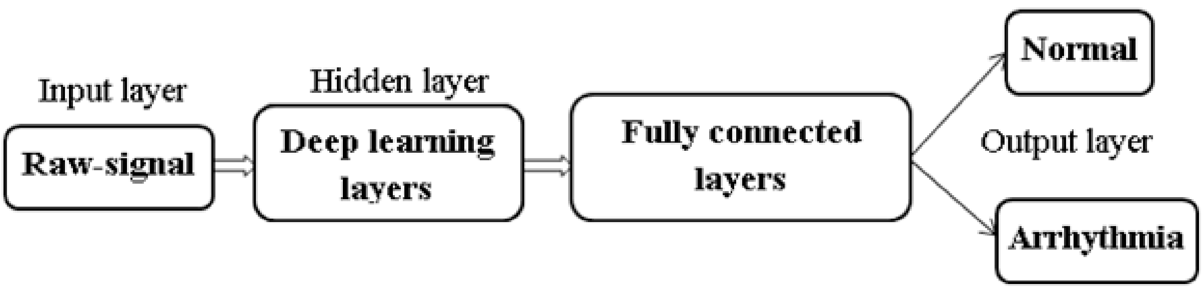

Fig. 1. Overview of the proposed deep learning architecture.

4. Conclusion, Discussions and Future work

Cardiac arrhythmia is basically an irregularity in heart rhythm. Some types of cardiac arrhythmia can lead to complications like stroke, heart attack and may even lead to sudden cardiac death. So, timely detection and diagnosis of arrhythmia is very important. Once arrhythmia is detected, next stage of identification of category of arrhythmia can be done. We developed an automated non-invasive system based on deep learning networks to perform the basic classification of a given ECG data as belonging to normal ECG or abnormal (having arrhythmia) ECG using the most popular publically available MIT-BIH arrhythmia database. We compared the performance using a variety of deep learning architectures of CNN, CNN-RNN, CNN-LSTM and CNN-GRU and obtained an accuracy of 0.834.

With concern on computational cost, we are not able to train more complex architecture. The reported results can be further improved by using more complex deep learning architecture. The complex network architectures can be trained by using advanced hardware and following distributed approach in training that we are incompetent to try.

We have discussed the role of deep learning techniques such as CNN and recurrent structures in the task of arrhythmia classification. The highlight of the proposed method is that it doesn't need any noise filtering and feature engineering mechanisms. The results obtained prove that the performance of our method is better than other published results in effectively classifying ECG as belonging to normal or arrhythmia class. Though deep learning networks produces excellent results, the disadvantage lies in the insufficient understanding of the complex inner mechanisms of the deep learning networks. This could be overcome by remodelling the nonlinear deep networks to a linear form by computing eigenvalues and eigenvectors in different time steps[20]. The future work can be the collection of real world datasets from hospitals having cardiac care units and the application of the same methodologies to the real datasets.

Acknowledgements

We are grateful to NVIDIA India, for the GPU hardware support to research grant. We are grateful to Computational Engineering and Networking (CEN) department for encouraging the research.

References

[1] Naser Safdarian, Nader Jafarnia, and Gholamreza Attarodi. (2014) “A new pattern recognition method for detection and localization of myocardial infarction using T-wave integral and total integral as extracted features from one cycle of ECG signal.” *Journal of Biomedical Science and Engineering* **7**(**10**)**:**818-824.

[2] Sharma L.N., Tripathy R.K., Samarendra Dandapat. (2015) “Multiscale energy and eigenspace approach to detection and localization of myocardial infarction.” *IEEE Transactions on biomedical engineering***62**(**7**):1827-37.

[3] Rajendra Acharya U., Hamildo Fujita, Vidya K Sudarshan, Shu Lih Oh, Muhammad Adam, et al. (2016) “ Automated detection and localization of myocardial infarction using electrocardiogram: A comparative study of different leads.” *Knowledge-Based Systems***99**: 146-156.

[4] Rajendra Acharya U., Hamildo Fujita, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Muhammad Adam. (2017) “Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals.” *Information Sciences***415**: 190-198.

[5] Tahsin Reasat and Celia Shahnaz. (2017) “Detection of inferior myocardial infarction using shallow convolutional neural networks.”arXiv preprint arXiv:1710.01115.

[6] Babak Mohammadzadeh-Asl and Seyed Kamaledin Setarehdan. (2006) “Neural network based arrhythmia classification using Heart Rate Variability signal.” In *14th EuropeanSignal Processing Conference*pages 1-4.

[7]Abhinav-Vishwa, Lal M.K., Dixit S, and Vardwaj P. (2011) “Clasification of arrhythmic ECG data using machine learning techniques.” *International Journal of Interactive Multimedia and Artificial Intelligence***1**(**4**):67-70.

[8]Goutham Swapna, Dhanjoo N Ghista, Roshan Joy Martis, Alvin PC Ang, and SubbhuraamVinitha Sree. (2012) “ECG signal generation and heart rate variability signal extraction: Signal Processing, Feature detection, and their correlation with cardiac diseases.” *Journal of Mechanics in Medicine and Biology***12**(**4**):1240012.

[9] Sujadevi V.G., Soman K.P., and Vinayakumar R. (2017) "Real-time detection of atrial fibrillation from short time single lead ECG traces using recurrent neural networks." In *The International Symposium on Intelligent Systems Technologies and Applications* pages 212-221 Springer.

[10] Rahul K. Pathinarupothi , Vinayakumar R., Ekanath Rangan, Gopalakrishnan E., and Soman. K.P.(2017) " Instantaneous heart rate as a robust feature for sleep apnea severity detection using deep learning." In *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*pages 293-296.

[11] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. (2000) "PhysioBank, PhysioToolkit, and PhysioNet." *Circulation***101** (**23**):215-220.

[12] Jeffrey L. Elman. (1990) "Finding structure in time."*Cognitive Science* **14**(**2**): 179-211.

[13] Sepp Hochreiter, and Jürgen Schmidhuber. (1997) " Long short-term memory." *Neural Computation***9**(**8**): 1735-1780.

[14] Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. (1999) "Learning to Forget: Continual prediction with LSTM. "*Neural Computation***12** (**10**): 2451 – 2471.

[15] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. (2002) "Learning precise timing with LSTM recurrent networks." *The Journal of Machine Learning Research* **3**(Aug):115-143.

[16]Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. (2014) "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv*:1406.1078.

[17] François Chollet et al (2015) Keras

[18]Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. (2016) "TensorFlow: A system for large-scale machine learning." In OSDI, volume **16**, pages265-283.

[19] Paul J. Werbos. (1990) "Backpropagation through time: what it does and how to do it." *Proceedings of the IEEE***78** (**10**): 1550-1560.

[20] Reza Moazzezi. (2011) "Change-based population coding." *PhD thesis, UCL (University College London).*