



Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data

Hari Mohan Rai¹ · Kalyan Chatterjee¹

Received: 10 January 2021 / Accepted: 16 July 2021 / Published online: 11 August 2021
© Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Automatic and accurate prognosis of myocardial infarction (MI) using electrocardiogram (ECG) signals is a challenging task for the diagnosis and treatment of heart diseases. MI is also referred as a “Heart Attack”, which is the most fatal cardiovascular disease. In many cases, MI does not show any symptoms, hence it is also called a “silent heart attack”. In such cases, patients do not get time to prepare themselves. Hence this disease is more dangerous and fatal with a high mortality rate. Hence, we have proposed an automated detection system of MI using electrocardiogram (ECG) signals by a convolutional neural network (CNN), hybrid CNN- long short-term memory network (LSTM), and ensemble technique to choose the optimum performing model. In this work, we have used 123,998 ECG beats obtained from the “PTB diagnostic database (PTBDB)” and “MIT-BIH arrhythmia database (MITDB) to develop the model. The experiment is performed in *two* stages: (i) using original and unbalanced datasets and (ii) using a balanced dataset, obtained from synthetic minority oversampling technique (SMOTE) data sampling technique. We have obtained the highest classification accuracy of 99.82 %, 99.88 %, and 99.89 % using CNN, hybrid CNN-LSTM, and ensemble techniques, respectively. Hence the proposed novel data balancing technique (SMOTE-Tomek Link) not only solves the imbalanced data problem but also increases the minority class accuracy significantly. Now our developed model is ready for the clinical application that can be installed in hospitals for the detection of MI.

Keywords ECG · BIG Data · Deep Learning · CNN · LSTM · MI · SMOTE

1 Introduction

In the word myocardial infarction (MI), ‘myo’ refers to muscle, ‘cardial’ refers to heart and ‘infarction’ represents the death of the tissue, which means the death or damage of heart tissue due to deficiency of blood supply is called as myocardial infarction, commonly known as “heart attack” [1]. It is the most common and fatal cardiovascular disease (CVD), which impedes blood flow to the heart muscle due to partial or complete blockage of the coronary arteries [2]. The oxygenated blood to the cardiac muscle is supplied by coronary arteries, if there is any obstruction into it, the heart muscle segment may

die due to lack of blood flow into it [3]. The damage or death of cardiac muscle tissue causes a change in the normal cardiac conduction system, resulting in life-threatening arrhythmias which may lead to sudden cardiac arrest. Several symptoms may be seen in the case of MI such as chest pain, breathing problems, and unconsciousness but many individuals do not experience any symptoms, that why it is also called “silent heart attack”. According to one estimate, it has been found that about 22 to 65 % of MIs do not show any symptoms, which means that they are silent. Therefore, patients do not get time to prepare themselves, which makes the disease more dangerous and fatal and the mortality rate is very high as a result, the mortality rate of MI is very high [4]. Hence the early detection of MI is very much important to provide timely treatment and reduce the mortality rate. The Electrocardiogram (ECG) signal analysis is the most appropriate technique for detecting the MIs at an early stage because the ECG signals are the recording of the electrical activity of the cardiac functionality which can be operated at a very low cost and also it is non-invasive in nature. But the manual and incorrect detection of cardiac abnormality may lead to loss of life of the patients suffering from MIs, for example, the expert

✉ Hari Mohan Rai
harimohanrai@gmail.com

Kalyan Chatterjee
kalyanchatterjee@iitism.ac.in

¹ Department of Electrical Engineering, Indian Institute of Technology (ISM), Dhanbad, India

cardiologist may recognize only 82 % of the MI ST-segment elevation [5]. According to the research report, 10 % of people who have significant symptoms such as cardiac diseases, cancer, and infections, are either incorrectly, diagnosed or diagnosed at a later stage of the disease. Among them, 53.9 % are either completely disabled or died due to medical error or wrong diagnosis [6]. Hence the automated detection with utmost accuracy along with high individual class accuracy (recall) is very much essential to assist the medical practitioner (cardiologist) by accurate, and quick detection of MI from ECG signals.

In the recent era, numerous researchers have contributed a lot in the area of arrhythmia detection such as MI, Premature ventricle contractions, atrial fibrillation, and ischemia, and many other arrhythmias using ECG patterns. In automatic detection approaches for recognition MI, machine learning (ML) is extensively investigated to analyze and diagnose cardiac abnormality. For the same, the publically available variety of databases are utilized to validate the performance of the ML techniques used for arrhythmia classification and the most used database for the detection of MIs is PTB diagnostic database. The traditional machine learning (ML) based classifier performance depends upon the preprocessing of the ECG signal as well as various feature extraction. Since features are the most important parameters of traditional ML-based classifiers, researchers have employed different types of features to boost their classifiers performance, and some of the commonly used features are WT (wavelet transform) based features [7–11], PCA (principal component analysis) [12], linear discriminant analysis (LDA) [8], Empirical Mode Decomposition (EMD) [13], time-domain features [14], Statistical features [2, 7], time and frequency domain features [15], Power Spectral Density (PSD) and Discrete Fourier Transform (DFT) [16], and Hilbert Transform (HT) [17].

The extraction of features from ECG waveforms is an essential task, which is performed during the classification of ECG arrhythmia using a traditional ML-based approach. Also, it takes a lot of hard work and time to extract a feature from the ECG database, one by one, based on patients or records. Besides, incorrect, inappropriate, and inadequate features may result in poor classification performance rather than improvement. As the size of the dataset increases, the time and effort it takes to extract the feature also increases, and tools installed on smaller databases may also be inadequate. To overcome all these issues, scientists, engineers, and researchers came up with the solution of deep learning (DL) based deep neural network (DNN). DNN technology automatically extracts features without human intervention or using any additional tools or techniques. It also performs well on large datasets, and on extensive learning, greatly reduces the effort and time that was applied when extracting features, and provides very good performance measure values. Over the past few years, DL-based DNN methods have become widely

and mostly used tools in the field of medical healthcare, particularly in biomedical signal and image processing.

For the past few years, the use of DL based DNN methods such as CNN (convolutional neural network) [3, 18–20], RNN (recurrent neural network) [4], Residual Network (ResNet) [21], GAN (Generative Adversarial Networks) [22], autoencoders [23], have been used for the feature extraction and detection of MI from ECG signals.

Rahhal et al. [24] presented a technique for the dynamic categorization of ECG waveforms using the DL method. After the completion of the feature learning process, the deep neural network (DNN) was developed by adding a softmax layer above the hidden representation layers. Kora proposed a Hybrid Firefly Algorithm for the detection of Myocardial infarction (MI) from ECG waveforms. Instead of extracting the features using the existing techniques, the RAW ECG signal was directly optimized using hybrid firefly and particle swarm optimization (FFPSO). The result reported for MI detection using ANN with the FFPSO algorithm provides 99.3 % accuracy, 98.7 % of specificity, and 99.7 % of sensitivity on the MIT-BIH and NSR database [25]. Acharya et al. [3] proposed the automated CNN-based algorithm for the automatic detection of MI from ECG signals. The study is to predict myocardial Infarction (MI) from noisy and noise-free ECG waveform. The detection of MI on the noisy database was achieved 93.53 % accuracy and filtered ECG received 95.22 % accuracy using the CNN technique. Acharya et al. [26] presented an automated technique for the detection of Coronary artery disease (CAD) which may cause myocardial infarction, using two modified Deep CNN models with 94.95 % accuracy of 94.95 and 95.18 %, respectively. Liu et al. [18] presented a myocardial infarction (MI) detection algorithm using a convolutional neural network (on) on multi-lead ECG signals. Their ML-CNN model uses sub-2D convolution which can use the complete character of total leads, it also uses a 1D filter to generate local optimal features. The Algorithm was evaluated on the PTB diagnostic ECG dataset, and it achieved 95.40 % of sensitivity, 97.37 % of specificity, and 96 % of accuracy. Sharma et al. [27] proposed a new method for the classification of MI from the PTDBD database using a filter bank named optimal biorthogonal. The k-nearest neighbor classifier has been used for the classification of the MI with an accuracy of 99.62 % on noisy ECG signals and 99.74 % on denoised ECG signals. Faust et al. [28] proposed automated atrial fibrillation (AF) detection technique using the LSTM network on the MIT-BIH dataset with 98.51 % accuracy. Fujita & Cimr [29] presented an automated detection technique of fibrillations and flutters using a deep CNN model with a 98.45 % classification accuracy of 99.27 %.

Kayikcioglu et al. [30] proposed a technique to categorize the ST segments using the distribution of time-frequency dependent features for the detection of myocardial infarction (MI) from multi-lead ECG waveforms. The weighted

K-nearest neighbor (KNN) algorithm provided the best performance with an average sensitivity of 95.72 %, an accuracy of 94.23 %, and a specificity of 98.18 % using the choi-williams frequency distribution (CWFD) features. Liu et al. [31] Presented a new hybrid RNN (LSTM) based method for the prediction of MI from 12 lead ECG and obtained an overall 93.08 % accuracy. Despite these excessive efforts, still there are few scopes of improvements base on the review carried out on the MI detection from ECG signals. Almost all the ECG databases are having the issue of data imbalance, where the majority class samples are in large amounts whereas the minority class is very less. Data imbalance issues may not be visible in overall or average accuracy but it can be visualized on the class-wise accuracy and other metrics score such as precision, F1-score, recall, etc. Many researchers have not looked at this issue and hence this area is mostly untouched.

To resolve the issue of the imbalanced dataset, we have come up with the solution by applying SMOTE and Tomek link resampling techniques. We have also applied two separate deep learning models CNN and CNN-LSTM to predict the MI from the MIT-BIH and PTB datasets. The models are tested on 24,800 ECG beats and also MI is predicted using ensemble technique which produces 99.89 % overall accuracy on the balanced dataset.

The novelty of our work is that we have proposed: CNN, hybrid CNN-LSTM, and ensemble techniques for the detection of myocardial infarction from ECG signals. To the best of our knowledge, this is the first work that jointly uses MITDB and PTB to detect MI. Furthermore, it is 1 work that employs the SMOTE + Tomek link sampling technique to solve the issue of data imbalance or class imbalance. Also, this work utilizes 123,998 single lead ECG beats including 24,800 test beats, and provides more than 99 % accuracy using all three proposed models.

The structure of this paper is arranged as; Sec. 2 briefs about the materials and methods employed in this chapter including ECG dataset, data preprocessing, data balancing, CNN, and LSTM models. Section 3 explains the proposed methodology using a workflow diagram and Sec. 4 describes the result and experimentation. Section 5 briefs the discussion over the result and outcome, and the final Sec. 6 provides concluding remarks and future work of this paper.

2 Materials & methods

2.1 ECG dataset

The datasets employed in this work include two benchmark datasets, MITDB (“MIT-BIH arrhythmia database”) and PTBDB (“Physikalisk-Technici Bundesenstadt database”). The MITDB includes 30 min (half-hour) recordings of ECG signals from 47 individuals in a total of 48 records. This is

open access freely available database that can be downloaded from the PhysioNet website [32]. The annotation used for this database is categorized into 5 beats type according to the AAMI (“Association for the Advancement of Medical Instrumentation”) standard [33].

The PTBDB comprises 549, 12 lead ECG recordings from 290 individuals publically available at the PhysioNet website to download [34]. Especially, this database contains 148 individuals diagnosed with MI in 368 recordings and 52 individuals with healthy control in 80 recordings, apart from this other records are diagnosed with 7 different types of arrhythmias. The ECG signals in PTBDB and MIT-BIH have been digitized at a sampling frequency of 1000 Hz, and for this work, from 12 lead recordings lead-II recordings have been used, and 2 types of beats MI and healthy controls (normal) have been utilized for analysis purpose. Both the databases, MITDB and PTBDB are available in the segmented form to download at Kaggle respiratory and contributed by [35, 36]. There are 109,446 ECG beats are available in the MITBD of 5 classes, and 14,552 ECG beats of two classes (abnormal and normal) are available in PTBDB.

For the detection of myocardial infarction, we have grouped all non-MI classes (N, S, V, F, Q) as a normal class and Myocardial infarction beats into MI class, hence the total number of beats to MI class is 10,506 and in normal class is 113,492. The total number of ECG beats in the combined dataset and the number of beats into the grouped classes are presented in Table 1. Figure 1 shows the different types of arrhythmia present in the combined dataset, where myocardial infarction is presented by the M class and all other types N, S, V, F, Q are combined as a non-MI class. The ECG beats are already segmented from p-peak to QRS complex and the maximum length of a single ECG beat is 188 samples, the remaining beats are zero-padded to make all the beats 188 sample size [36].

2.2 Data balancing

If the difference between the two classes is huge in terms of the number of samples, then the data is said to be unbalanced. In this case, the class that holds the maximum number of data is called the majority class, and which contains the minimum number of samples is called the minority class. This means that in the case of data imbalance, one class (majority) has more dominance or influence over the other classes [37]. This database is one of the best examples of data imbalance problems because the majority class is more than ten times larger (113,492 beats) than the minority class (10,506 beats).

Hence the imbalanced data sets mostly impacted the classifier performance because classifiers are mainly designed for balanced class problems. However, the slight difference in the number of classes is acceptable but large skewness in the data leads to inadequate training of the classifier and resulting in

Table 1 Number of beats in each class and combined class in the dataset

(Arrhythmias type)	Types of ECG beats	No. of Beats	Grouped Class
N	Normal	94,635	113,492 (N)
S	Supra-ventricular premature (SVP)	2779	
V	Premature ventricular contraction (PVC)	7236	
F	Fusion of ventricular and normal (FVN)	803	
Q	Paced	8039	
M	Myocardial Infarction (MI)	10,506	10,506 (M)
Total ECG beats		123,998	123,998

improper classification. This may also result in overfitting problems for the majority class and “underfitting” problems for the minority classes. The classifier will be trained well for the majority class and will be untrained for the minority class, hence it is inclined towards predicting majority classes more successfully as compared to the minority class. Another issue is that this imbalanced data may not be visualized in the overall accuracy of the classification for that it is required to also validate the performance using different evaluation metrics.

2.2.1 SMOTE and Tomek links sampling

The method applied to oversample the minority class data (MI) is SMOTE (“Synthetic Minority Over-sampling Technique”) which creates the synthetic samples close to the minority class instead of simply generating multiple copies. The SMOTE generates the minority data samples based on the

K-nearest neighbor algorithm, it selects the nearest neighbor from the minority data samples, and then based on linear interpolation new samples are generated. The number of minority class and the amount of oversampling its complexity increases, in this case, the auto sampling technique have been selected to automatically decide the amount of oversampling to create [38].

Tomek links are the technique of under-sampling which reduces the number of instances from majority classes based on the data samples which belong to the borderline. It makes pairs of majority and minority classes near the border classifying the classes and removes the majority of samples from each majority class. This process is kept on repeating and creates the gap between the borderline of majority and minority classes by deleting the majority samples [38].

There are mainly three ways to apply SMOTE and Tomek link sampling methods, First way is to Apply SMOTE

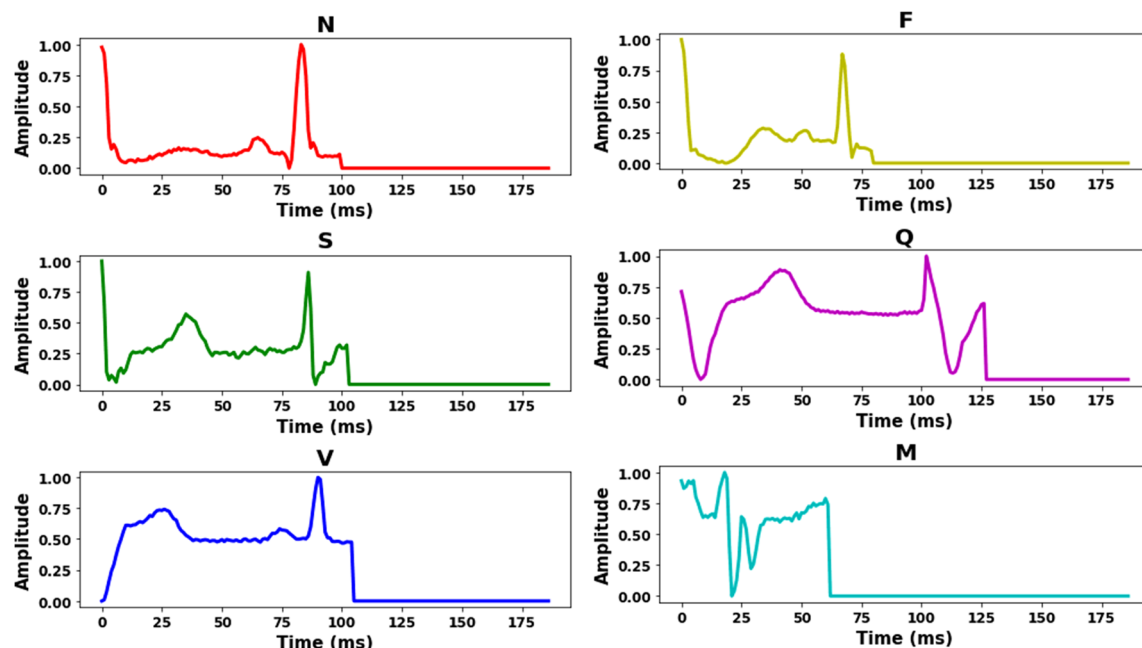


Fig. 1 Visualization of the arrhythmias types, MI (M) and non-MI (N, S, V, F, Q)

oversampling 1st which will generate synthetic samples of minority samples depending upon the application, generally, in “auto” mode it equates minority and majority samples and then applies Tomek link undersampling to remove the majority samples which overlaps with the minority on borderline. The 2nd way is to apply Tomek link sampling 1st and then apply SMOTE oversampling to generate a minority number of samples. A third way is the hybrid method, In this case, automatically it generates the minority samples, and parallelly it also assures that the generated samples should not overlap with the minority class, in rare cases if any samples overlap then it automatically removes the majority class and preserve the minority samples.

For this work, we have used hybrid SMOTE + Tomek links which means it parallelly generates oversamples and maintains the data dissimilarity between the minority and majority samples, and also makes the balance between the data classes. The algorithm used for the hybrid SMOTE + Tomek link data balancing technique is presented in Algorithm-I.

ALGORITHM-I

ECG Dataset (D): Training (Tr) + Testing (Ts)

Input: Training (Tr), Majority(Mj), Minority (Mn)

Output: Tri = (Tri_X, Tri_Y)

Tri=(Tr_Xi, Tr_Yi)

Class(i)= (n,m), i=0,1

Class(0) = n

Class(1) = m

Find Majority and minority class:

for i in Class:

Mj= max(Class(i))

Mn=Min(Class(i))

Return (Mj, Mn)

Apply SMOTE-Tomek Link:

Smt_i=SMOTETomek(Mj, Mn)

(Tr_Xi, Tr_Yi) = smti.fit_smample(Tr_X, Tr_Y)

Tri=((Tr_Xi, Tr_Yi)

repeat (i=0,1)

Update Classi

End

Since two types of ECG beats are used to detect MI, normal beats (N) and myocardial infarction (MI) beats, where normal beats are in the majority and MI in the minority. Hence the synthetic samples will be generated to MI beats (minority) only to equate the number of samples to the majority class.

The MI beats generated using the proposed SMOTE + Tomek link data sampling technique are randomly selected to compare similarity with the original MI beats. So for comparison purposes, we have randomly chosen two original MI beats (M1 and M2) and three SMOTE + Tomek link generated MI beats (G1, G2, and G3). Figure 2 pictures the sample of

original MI beats and SMOTE + Tomek link generated beats, respectively.

2.2.2 Originality analysis

To analyze the originality between original minority classes MI beats samples with SMOTE-Tomek link generated MI beats, we have applied two types of quality assessment metrics, mean square error (MSE) and structural similarity index measure (SSIM). MSE shows the disparity or difference between two images at pixel levels, the higher the MSE value the greater the difference between the images and vice versa. The MSE signifies the aggregate average dissimilarity between the pixels of the overall images, which is represented by squared errors among the two images. However, its mathematical computation is imperative to the edges of the images, the mathematical equation of MSE is given by Eq. (1) [39].

$$MSE = \frac{1}{xy} \sum_{i=0}^{x-1} \sum_{j=0}^{y-1} [M(i,j) - G(i,j)]^2 \quad (1)$$

where ‘M’ represents the original MI beat and ‘G’ represents the generated MI beat, and ‘x’ and ‘y’ represent the row and column of the images used, respectively.

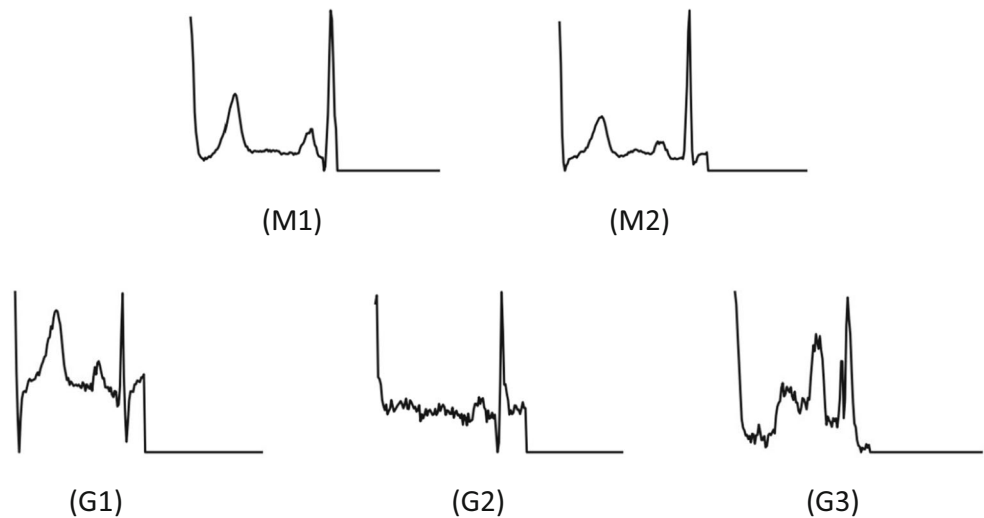
SSIM is mainly used to calculate the similarity between two images based on their structure. It is also a quality estimation metric which is an improved version of MSE and Peak Signal to Noise Ratio (PSNR). For the estimation of perceived errors, the (MSE) and (PSNR) methods are employed, whereas the SSIM is the measure of the deterioration in picture quality as a perceived structure information change. Structural similarity (SSIM) is mainly calculated using the structural term, contrast term, and luminance term, the higher the value of SSIM, the higher the similarity between the two image structures. The computation of SSIM is done on the several windows of the images, it is given by the Eq. (2) [39].

$$SSIM(M, G) = \frac{(2\mu_M\mu_G + c_1)(2\sigma_{MG} + c_2)}{(\mu_M^2 + \mu_G^2 + c_1)(\sigma_M^2 + \sigma_G^2 + c_2)} \quad (2)$$

Where μ_M , μ_G , σ_M^2 , σ_G^2 , and σ_{MG} are the means, variance, and covariance of the two images (M, G) respectively. C1 and C2 are stabilization variables for stabilizing the division of feeble denominators.

A visualization of the calculated MSE and SSIM of the generated images along with the original images is shown in Fig. 3. Where, M1 and M2 represent sample images of the first and second original MI beats and, likewise, G1, G2, and G3 are sample images of the generated MI beats. In addition, (M1, G1), (M1, G2), (M1, G3) represents the first original MI beat compared to the first, second and third generated MI beats respectively. Similarly, (M2, G1) indicates that the MSE and SSIM are calculated for the second original MI beat versus the first generated MI beats and so on. From Fig. 3, it is observed

Fig. 2 The Original MI beats (M1, M2) and SMOTE-Tomek link Generated beats (G1, G2, G3)



that the average SSIM is around 90 % and the average MSE value is around 0.035 indicating a fairly good similarity between the original and synthetically generated beats.

2.3 CNN (Convolutional Neural Network) model

CNN was initially designed for the pattern recognition tasks in the field of computer vision for edge detection, segmentation, and object detection task. But due to its simple architecture and wide applicability, it is now being used in almost every application of the machine learning field such as computer vision, text recognition, data mining, speech and signal processing, image and object detection, and many other areas. It is also known by the name ConvNet and its capability of generating features without human intervention makes it versatile and it is also available in 1D, 2D, and 3D options[40].

CNN has generally a prescribed structure that may be modified depending upon the applications. The commonly used layers in CNN are Convolution (Conv) Layer, ReLU Layer, Pooling Layer, Batch Normalization, and Fully-connected Layer [41]. The Conv layer is the 1st and foremost layer in the architecture of CNN which extracts the feature from the input provided to it based on some parameters such as kernel (filter) size and step size (stride). ReLU is the non-linear operation that stands for “Rectified Linear Unit”, it is a type of activation function which introduces non-linearity into the network. The pooling layer does not have parameters to learn, it reduces the feature size by half depending upon the type of pooling applied. Mainly, two types of pooling operations are available, max-pooling (MaxPool) and average pooling (AvgPool) [42]. The batch normalization process normalizes every layer’s input by applying the variance and mean of the present batch, during the training process. This layer is always the last layer of the network but it doesn’t come under the characteristics of CNN. It is an MLP (multilayered

perceptron) wherein the output layer contains a softmax activation function.

2.4 LSTM (Long Short-Term Memory Network)

LSTM network is the widely and most preferred DL model used for sequential and time-series data, providing a solution to short-term memory (STM) as to why it is named “long-‘short-term memory’”. This issue of STM is resolved by LSTM with the help of gates called in its internal structure which is used to regulate the information flow. The LSTM is the best-suited model for time series data such as language translation, weather prediction, and speech recognition and its operation process of gates carries forward the important information in the longer data sequence to make accurate and efficient predictions.

The basic concepts of LSTM operation are in gates and cell states, and these are, Forget Gate(f_t), Input Gate (i_t), Cell state (C_t), Output gate (O_t), Hidden state (h_t)[43]. Forget Gate (f_t) is generally the 1st gate of the LSTM network in which the present input (x_t) is concatenated (cn) with the previous hidden state (h_{t-1}) and passes through the sigmoid activation function to produce the output between 0 and 1. Input Gate (i_t) is mainly used for updating the memory or cell state, in this gate also sigmoid function is used between the concatenated value of current input (x_t) and previous hidden state (h_{t-1}). Cell Sate (C_t) is also known as the memory cell, it is the point-wise multiplication of the previous cell state value (C_{t-1}) with the forget gate (f_t) value and addition of the candidate (C_t) vector multiplied with the input gate (i_t) vector. Output Gate (O_t) is used for calculating the hidden state value, and it is also calculated in a similar way as input and forget gate but with different weights (w) and bias (b) values, here also the sigmoid activation (σ) function is used for providing the output between 0 and 1. The hidden state (h_t) value will be

(M1, G1)	MSE: 0.0331, SSIM: 0.9013	0.0331	0.9013
(M1, G2)	MSE: 0.0356, SSIM: 0.8937	0.0356	0.8937
(M1, G3)	MSE: 0.0388, SSIM: 0.8811	0.0388	0.8811
(M2, G1)	MSE: 0.0275, SSIM: 0.9208	0.0275	0.9208
(M2, G2)	MSE: 0.0323, SSIM: 0.9031	0.0323	0.9031
(M2, G3)	MSE: 0.0421, SSIM: 0.8729	0.0421	0.8729
Average		0.035	0.90

Fig. 3 Visualization of MSE and SSIM of generated beats (G1, G2, G3) compared to original MI beats (M1, M2)

computed with the help of output gate and cell state value, the updated cell state value will pass across the ‘*tanh*’ activation (t) to produce the output between -1 and 1 and these values will get multiplied (vector multiplication pointwise (\times)). This is the actual output of the LSTM network which will also be

carried forward to the next stage or timestamp. LSTM uses sigmoid and tanh activation functions throughout the operation, where sigmoid activation receipts a real value in the input and provides the output range between 0 and 1 [44] and tanh activation function regulates the output data between -1 and

1. The basic structure of the three-stage LSTM network is shown in Fig. 4.

3 Proposed methodology

The methodology employed for the prediction of myocardial infarction (MI) from ECG signals using CNN and the hybrid CNN-LSTM model is demonstrated in Fig. 5, where the workflow diagram is visualized step-by-step. From Fig. 5, the raw ECG has been collected from the MIT-BIH and PTB databases in the first stage, in the 2nd step, the pre-processing of the raw data has been performed, in 3rd step, the data distribution process has been accomplished, in the 4th step, the training using proposed hybrid deep learning models have been completed and in the last and final step, the testing and verification of the dataset have been done.

In this work, automatic MI detection methods have been employed on the big ECG datasets. The ECG signals are 1st preprocessed by filtering and segmenting them and then the time interval and gradient of these time series data were calculated. Based on the time interval values the ECG graph is plotted and this complete ECG signal has been used as the input to the proposed CNN model. Before applying the input to the CNN model the data has to be distributed into train and test class because the resampling technique will be only applied on the training dataset, test data will be reserved for only testing the model, it is not exposed in the training. In the next step, the preprocessed imbalance data is directly trained on the training dataset using the CNN model and also the CNN + LSTM model. In the final step, the imbalanced dataset is balanced using the SMOTE + Tomek link method and then trained using the CNN and CNN + LSTM model. Many researchers have applied the data augmentation (DA) technique for generating more synthetic samples but it does is not preferable in the case of time series data or signals. Since the ECG signal is time-series and non-linear data we have preferred LSTM in combination with CNN among many deep learning models.

3.1 Data preprocessing

Data preprocessing is one of the most important tasks in an object or class categorization or detection using ML or DL classifiers. In this stage, the study and analysis of data, filtering, resizing, segmentation, augmentation, and many things are included which makes data more smooth and appropriate before providing it to the classifier. The dataset utilized for this work was already filtered and segmented to a fixed length of 188, and it was also down-sampled zero-padded to obtain shorter beats, which makes it more suitable for deep learning applications [36].

The ECG signal sample is a time series one-dimensional signal hence in this stage the time interval of the signal is computed on the filtered and segmented dataset. Since the signal is time-series data that varies with respect to time in a one-time stamp to next, it is preferred to find the time interval of the signal. The time interval of the signal is calculated by differentiating the next timestamp (ECG_{t+1}) from the current timestamp (ECG_t) of the ECG signal. Time interval (ΔECG) is given in Eq. (3):

$$ECG = ECG_{t+1} - ECG_t \quad (3)$$

Instead of extracting any other type of features of the ECG signal, we have preferred to utilize the time-series graph (time interval) as a feature. The gradient of the graph is given by Eq. (4).

$$gradient = \frac{ECG_{t+1} - ECG_t}{unittime} = \frac{ECG}{t} = \frac{d(ECG)}{dt} \quad (4)$$

The graph will be plotted using the gradient and the value, hence the complete ECG Waveform is used as the feature for the proposed model. Data preprocessing is one of the most important tasks in an object or class categorization or detection using ML or DL classifiers.

3.2 Data distribution

The data distribution also plays a very important role in the detection task using classifiers. In this work, 123,998 ECG

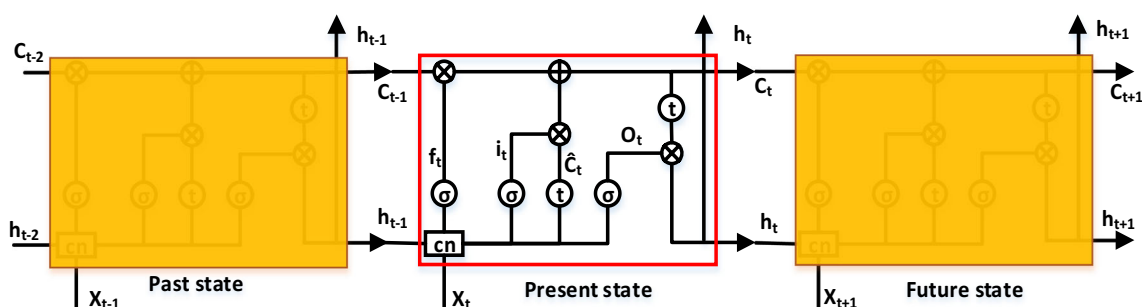
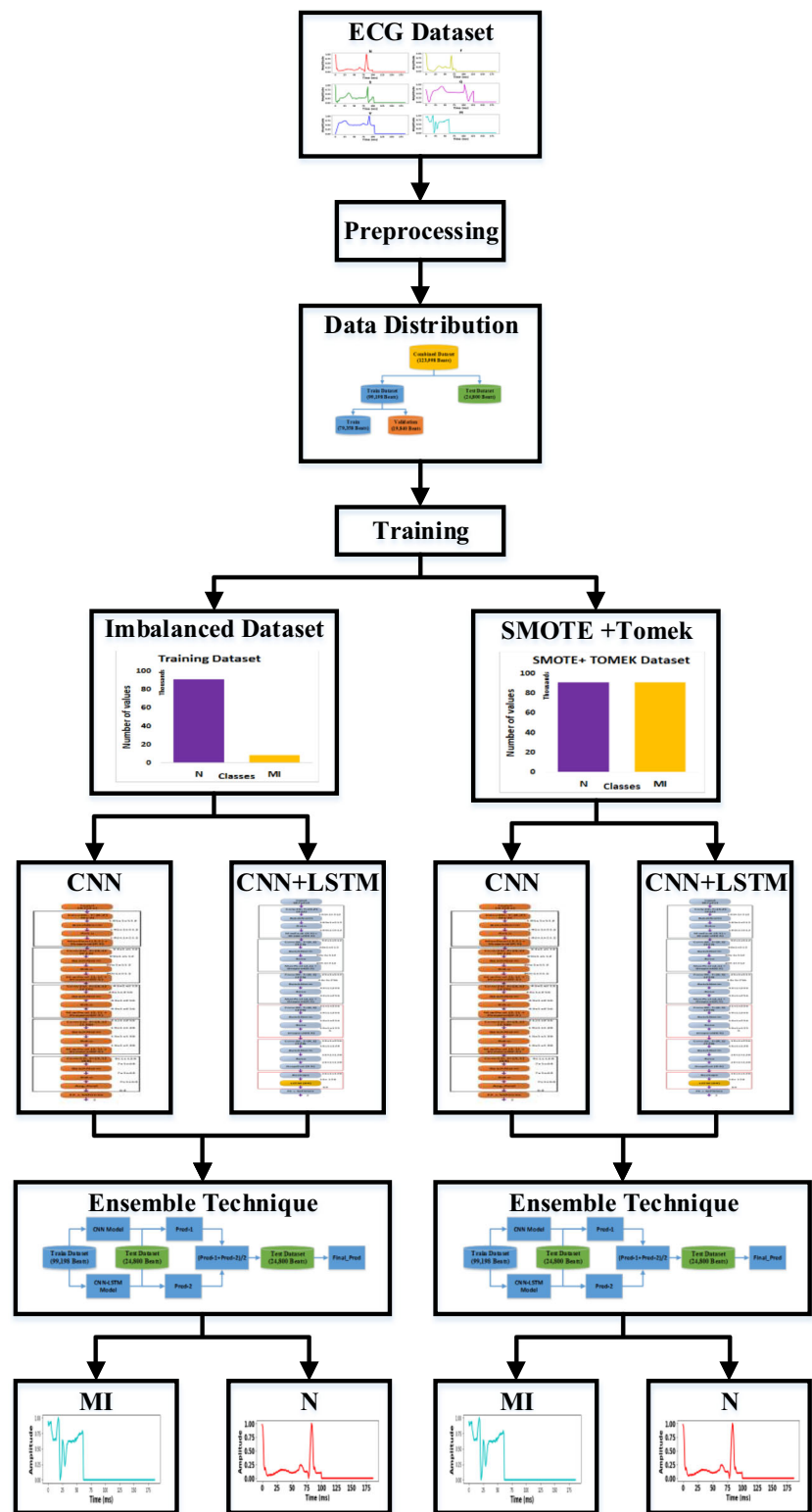


Fig. 4 Basic structure of three-stage LSTM network

Fig. 5 Block diagram of the proposed methodology



beat samples for detection of MI, are used for training and validating the proposed model.

Initially, the complete dataset was divided into two parts, (i) training, and testing. The test dataset is used for testing the

model, it is not used for the training process. The ratio of distribution is 80:20 for the train and test dataset, respectively i.e. 99,198 beats for training and 24,800 beats for testing the model. Further, the train data set have been divided into two

parts, training, and validation in the ratio of 80:20, respectively. Finally, 79,358 ECG beats are used for training the proposed model and 19,840 beats are used for validating the training performance. The distribution of datasets along with the number of ECG beats is visualized in Fig. 6.

3.3 Proposed CNN model

The 21 layers CNN deep learning model has been proposed to accurately and automatically detect the MI from the ECG signal, the structure of the proposed CNN model layer-wise is illustrated in Fig. 7. The complete structure is designed into four parts (segments), each part contains the convolutional layer followed by batch normalization, Relu non-linearization, and max-pooling with drop out layer. In the 1st segment, the Con2D layer is applied with parameters, such as filter size ($f=3,2$) and 512 number of filters to extract the maximum features in the 1st layer. Throughout the model, the stride is considered to be 1, and the filter size of the Con2D layer in each segment, except 1st, ($f=3,1$) is used. Also the max-pooling layer with size (2,1) and drop out value 0.5 is used throughout the network. Con2D layer varies in the filter number from 1st to 5th segment as 512, 512, 256, 128, and 64, respectively. The last segment does contain the max-pooling layer and dropout instead we have utilized a global average pooling layer for averaging the features, and these features are given to the final fully connected (FC) layer along with softmax activation to categorize the values into two classes, MI or Normal.

3.4 Proposed CNN-LSTM model

The 23 layer hybrid CNN + LSTM model is designed for the detection of MI from ECG signals and demonstrated the structure of the model, layer by layer in Fig. 7. The complete design I built up into 6 segments. The initial 3 segments are completely the same as the CNN model, but the 4th 5th and 6th segments are modified to optimize the better result. In this model also the filter size was kept constant for the con2D layer except for 1st and 5th where we have used ($f=3,2$) and ($f=4,1$) to match the feature size with the followed layer. The stride

and padding throughout the model are of one and valid, respectively. The segment 4th is replicated as the 3rd except for the max-pooling layer, in this segment we have not used the max-pooling layer for voiding further reduction of feature size. 5th segment also does not contain the max-pooling layer, and the drop-out layer (0.5 throughout) is shown in combination with the max-pooling layer but shown separately where the max-pooling layer is not available. In the last 6th segment the LSTM is combined with the CNN module, the 1st to 5th segments represent the CNN model and the 6th segment is added extra for embedding the LSTM layer into it and completed with FC + softmax layer similar to the CNN model [45].

3.5 Proposed ensemble technique

As the name implies the ensemble technique is the method used to combine the best predictions (pred) of several models together to improve the overall performance accuracy. This may also be called a post-processing method where after training the individual models and accomplishing the outcome of each model. The ensemble technique is applied such as averaging, max voting, weighted averaging, and many more to further improve the overall performance. In this technique, every individual model is known as a base-learner which contributes to the final outcome, and the flaws of each model are replenished by other members [46]. The ensemble model is the combined outcome of the base-learner which is known as meta-learner.

The motivation behind using the ensemble technique is to (1) enhance the overall performance of the proposed models, (2) to overcome the problem of data imbalance issue, and (3) to compensate for the biasing involves during the training process of any models.

In this work, we have used a simple and sequential ensemble method called the average ensemble technique. In this technique, the averaging is performed for each prediction on the test dataset and finally, the prediction is the mode on the average prediction model. In our work, the two individual models, CNN and CNN-LSTM are used separately to learn on the same training dataset and, make the prediction on the test dataset, the prediction of each individual model is averaged and the final prediction is made on this average prediction model. The complete process used for the average ensemble technique is demonstrated in Fig. 8.

3.6 Performance and parameter evaluation

There are several methods to evaluate the performance of the classifier, but we have used confusion matrix (CM), accuracy, and loss history curve and computed the various performance evaluation metrics to verify the model performance result. A CM is a table that summarizes the outcome of any

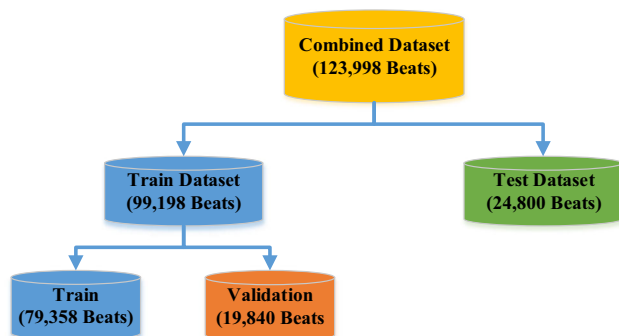
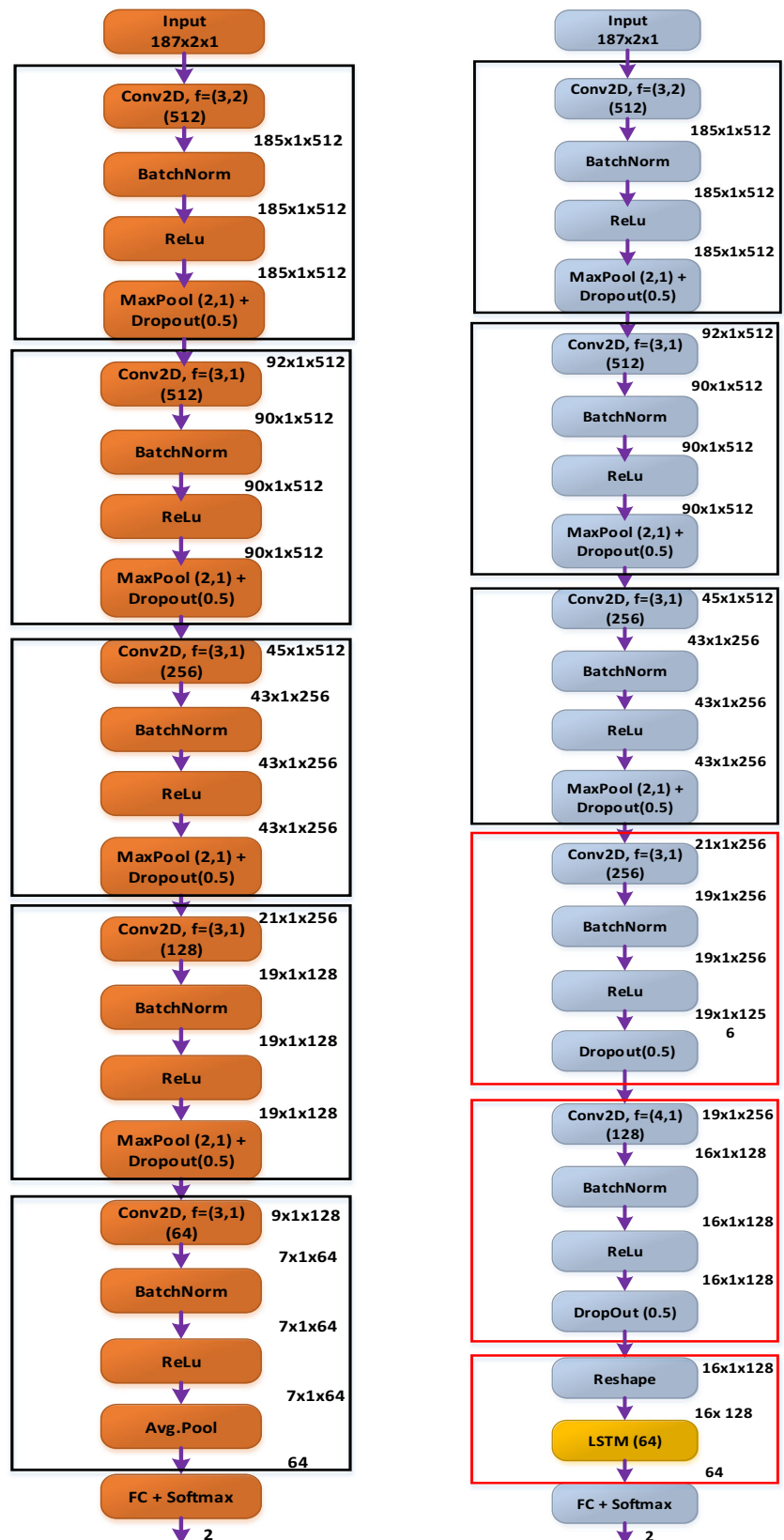


Fig. 6 Illustration of dataset distribution used for this work

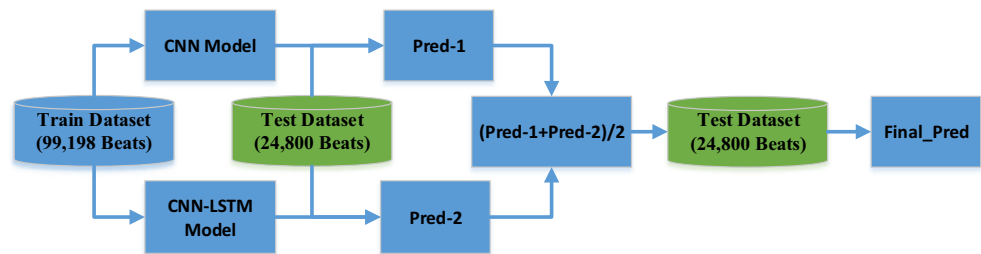
Fig. 7 Layer-wise structure of proposed CNN (left) and CNN-LSTM (right) model



classification algorithm by any classifier, it is a plot of actual classification Vs predicted classification. Using only

classification accuracy to authenticate the classifier model may be misleading or may not give the correct evaluation

Fig. 8 Structure of average ensemble technique used in this work



especially in case of an unequal number of observations. Hence the CM can not only provide a better idea of any classifier performance but also provides the many types of performance and parameter evaluation metrics. From the CM table, the accurately classified sample and wrongly classified (misclassified sample) can be easily obtained. It can be plotted by applying the simple command in python confusion_matrix (Actual, predicted) from the sklearn library [47].

Eight types of performance evaluation metrics, Recall (Re %), Specificity (Sp%), Precision (Pr%), Accuracy (Acc%), F1-score (F1 %) and Classification Error rate (CER%), Youden's index (YI%), and False positive rate (FPR%) are used for validating the classifier performance.

$$\text{Precision}(Pr\%) = \frac{TP}{TP + FP} * 100 \quad (5)$$

$$\text{Recall}(Re\%) = \frac{TP}{TP + FN} * 100 \quad (6)$$

$$\text{Specificity}(Sp\%) = \frac{TN}{TN + FP} * 100 \quad (7)$$

$$\text{Classification Error Rate}(CER\%) = \frac{FP + FN}{Totalbeats} * 100 \quad (8)$$

$$\text{Accuracy}(Acc\%) = \frac{TP + TN}{Totalbeats} * 100 \quad (9)$$

$$F1 - \text{Score}(F1\%) = \frac{2 * Precision * Recall}{Precision + Recall} * 100 \quad (10)$$

$$\text{Youden's index}(YI\%) = (Re + Sp) - 100 \quad (11)$$

$$\begin{aligned} \text{False Positive Rate}(FPR\%) &= (100 - Sp) \\ &= \frac{FPB}{TNB + FPB} * 100 \end{aligned} \quad (12)$$

4 Result and experimentation

The experiment is performed on 123,998 ECG beats from two databases: MITDB and PDBDB consisting of 10,506 MI beats and 113,492 non-MI beats (N) respectively. The experiment is performed in two parts, 1st on the original unbalanced dataset and 2nd is on the oversampled dataset, by applying SMOTE-Tomek link combined oversampling. The 24,800 ECG beats, 20 % of the total, are separated for testing the model performance, the remaining 99, 198 beats are used for training the model for 1st experiment. The python environment is used for validating the model with TensorFlow and Keras libraries. Laptops with hardware configuration, 8 GB RAM, 1 TB HDD, i5 core Pentium processor, and NVIDIA graphics card have been used for the experimentation.

Hyperparameters assume to be a fundamental part in model training because it straightforwardly controls DL performance and decidedly influences the model classification performance. In this manner, choosing the proper hyperparameters assumes a fundamental part in DL development's fruitful preparation as an inappropriate determination of hyperparameters can prompt wrong mastering abilities. For instance, if the model's learning level is excessively high, it might strike; it can miss the necessary information design if it is low. Accordingly, choosing the best hyperparameter helps look for capabilities in the hyperparameter space and assumes a fundamental part in the hyperparameter's simple administration, anticipating a huge experimental group [48].

After trying several hyperparameters and performing rigorous training of all the proposed models we selected the best hyperparameters for our proposed models presented in Table 2. Also, we found that both the DL models provide the best result on the same hyperparameters and also used the same data for training, testing, and verification for unbiased classification. $187 \times 2 \times 1$ input ECG signals are given as

Table 2 Optimization hyperparameters used for training the models

Model	Input Size	Optimizer	Learning Rate	Momentum	Decay	Mini batch size	Epochs
CNN CNN-LSTM	$187 \times 2 \times 1$	Adam	1e-3	0.9	1e-7	128	100

the input to the DL model and the final hyper-parameters employed during the training process, Adam optimizer, batch size 128, momentum 0.9, learning rate 1e-3, epoch 100, and 1e-7 decay.

Since the ECG datasets used in this work are highly imbalanced, we have used SMOTE + Tomek link data sampling or balancing or resampling methods to equalize the number of samples between the majority and minority classes. 80 % of the total ECG beats i.e. 99,198 beats are set aside for training and 20 % (24,800) are reserved for testing, including both arrhythmia classes. The contribution of each class to the training before sampling is as follows, with the general class being 91.5 % and the MI class 8.5 %. We have performed two independent experiments, one before sampling and another one after sampling. The total number of normal beats in the data set is 113,492 and MI beats are 10,506, which is a total of 123,998 beats. Among all training beats (99,198), we have 90,793 training beats for the normal class and 8,405 beats before sampling. After applying the Smote + Tomek link sampling method using Algorithm-I, we have 90,713 beats for each individual class, i.e. total of 181,426 beats after sampling which contributes equally (50 %) to the training samples. The distribution of the dataset before and after sampling is presented in Table 3.

The first experimentation is performed on imbalanced data set using CNN and CNN + LSTM models. The CNN model is trained 1st, then hybrid CNN + LSTM is trained on training and validation datasets. From a total of 99,198 training beats 79,358 beats (80 % of total) are used to train the models, and the remaining 20 %, 19,840 are used for validating the model performance. We have also predicted the MI on training and validation datasets to verify the learning performance of the model. Plots of accuracy and loss versus epochs were obtained using the CNN model with the training dataset shown in Fig. 9.

In the 2nd stage of the 1st experiment, the proposed CNN + LSTM model is trained on validation and trained

imbalanced (original) dataset. Plots of accuracy and loss versus epochs obtained history using the CNN-LSTM model during the training dataset are shown in Fig. 10.

Final detection is executed on the test dataset and the prediction results from both the models are ensemble to attain the final results. The training is performed on the original dataset (imbalance) and after training the model is validated on the test dataset (24,800 beats). The CM on the test dataset using CNN, CNN + LSTM, and ensemble model is shown in Fig. 11 which provided an overall accuracy of 99.57 %, 99.70 %, and 99.76 %, respectively. From the classification results, it can be noted that the ensemble model provided marginally better results than CNN and CNN + LSTM models. The performance evaluation metrics of each model are presented in Table 4. It can be seen from Table 3 that, the average class accuracy obtained using CNN, CNN + LSTM, and ensemble model is 98.36 %, 98.80 %, and 99 %, and also minority class (MI) accuracy is further reduced to 96.91 %, 97.72 %, 98.10 %, respectively.

This concludes that although our proposed models learned well and produced excellent overall accuracy. But it did not perform well on the minority class dataset because of the data imbalance problem. This data imbalance problem is resolved by oversampling the data by combining the resampling method SMOTE + Tomek link.

The 2nd experiment is performed on the balanced training dataset, oversampled by applying the SMOTE + Tomek link technique. The training dataset consists of 90,793 non-MI (N) and 8405 MI ECG beats. Hence to equate the majority and minority classes, the SMOTE + Tomek link oversampling technique is applied and then the training is performed on this oversampled dataset. After oversampling, the total number of ECG beats increased to 181,426, having 90,713 beats from each class. After applying the balancing technique, both the models are trained on the training and validation dataset. Plots of accuracy versus epochs and loss versus epochs obtained

Table 3 Optimization hyperparameters used for training the models

Types of Arrhythmias	No. of Beats	Test beats	Training beats	SMOTE + Tomek sampled beats	% beats before sampling	% beats after sampling
Normal (N)	113,492	22,699	90,793	90,713	91.5	50
MI	10,506	2,101	8,405	90,713	8.5	50
Total	123,998	24,800	99,198	181,426	100.0	100

The bold entries indicate the modifications to the data after applying the proposed data balancing technique. In the final row, it also shows the total number of datasets used for training, testing the model before and after sampling

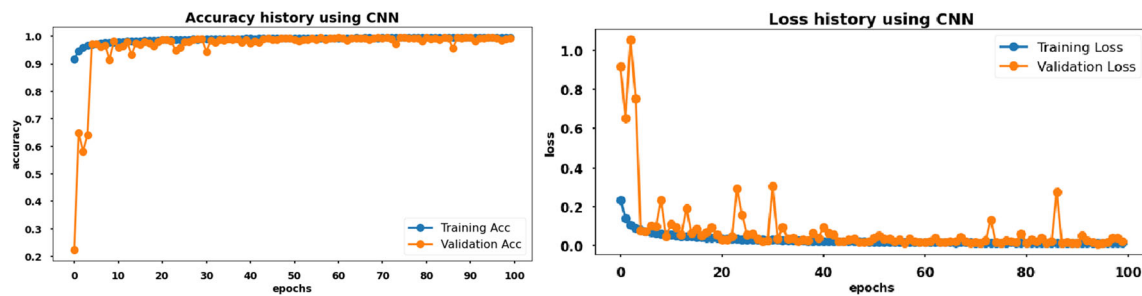


Fig. 9 Plots of accuracy and loss versus epochs obtained using CNN model with the training dataset

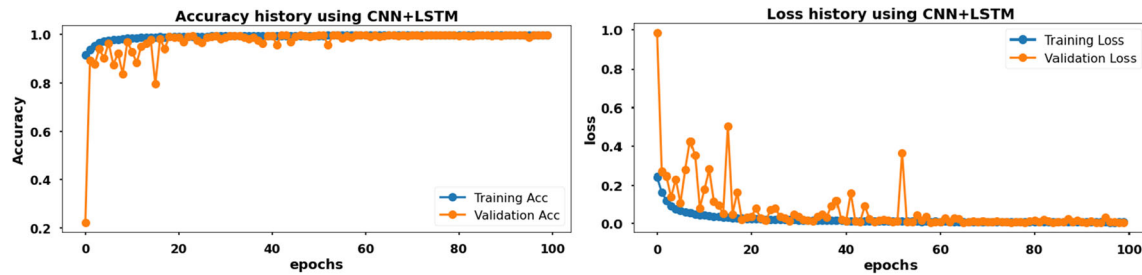


Fig. 10 Plots of accuracy and loss versus epochs obtained using CNN-LSTM model training dataset

using the CNN model for the balanced dataset are shown in Fig. 12.

Similarly, the proposed CNN + LSTM model is also trained on the balanced dataset. The plots of accuracy versus epochs and loss versus epochs obtained using the CNN-LSTM model for the balanced dataset are shown in Fig. 13.

Finally, the MI is detected using a 24,800 dataset which is common for all the experiments. We did not do any modifications to the test dataset. It is also unexposed during the training process to make an unbiased prediction. The confusion matrix (CM) on the test dataset using CNN, CNN + LSTM, and ensemble technique are shown in Fig. 14.

Table 5 shows the performance measures obtained using our proposed models with a balanced test dataset. It clearly indicates the superiority of our proposed models. All our proposed models have yielded more than 99.8 % accuracy. Hence, our model is ready for clinical application.

Also, we have presented the total training time taken by each model's CNN and CNN LSTM on the imbalanced and SMOTE + Tomek link sampled dataset in Table 6. It has been observed that the total time taken by the CNN-LSTM model is 38 min 5 s on the SMOTE + Tomek link sampled dataset because the total number of beats is nearly twice that of the original imbalanced dataset and the number of layers is even higher (23) compared to CNN (21).

5 Discussion

Most of the automated MI detection systems developed in the past few years have utilized the PTB database for the validation of the model performance [3, 5, 9, 14, 18, 25, 27, 49–57]. In this work, we have used two databases MITDB and PTBDB to increase the number of data samples for training

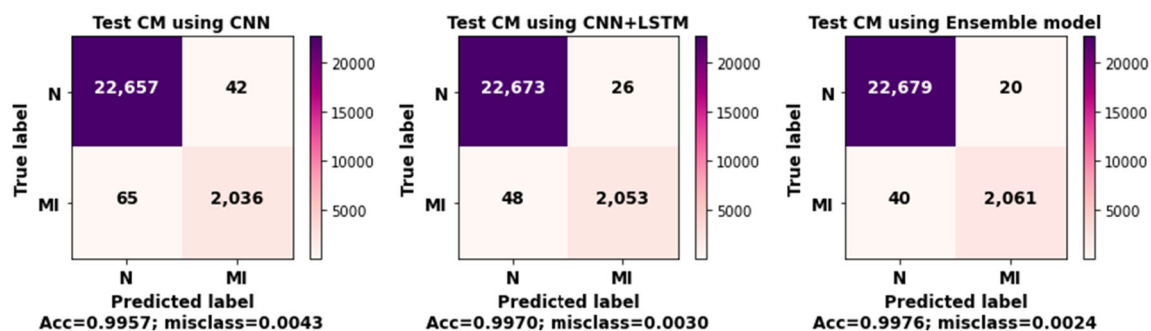


Fig. 11 Confusion matrix obtained for test dataset using our proposed models trained using imbalanced dataset

Table 4 Performance measures obtained using our proposed model with imbalanced test dataset

Method	Class	%Pr	%Re	%Sp	%YI	%FPR	%F1	%Acc	%CER
CNN	N	99.71	99.81	96.91	96.72	3.09	99.76	99.57	0.43
	MI	97.98	96.91	99.81	96.72	0.19	97.44		
Average		98.85	98.36	98.36	96.72	1.64	98.60		
CNN + LSTM	N	99.79	99.89	97.72	97.60	2.28	99.84	99.70	0.30
	MI	98.75	97.72	99.89	97.60	0.11	98.23		
Average		99.27	98.80	98.80	97.60	1.20	99.03		
ENSEMBLE	N	99.82	99.91	98.10	98.01	1.90	99.87	99.76	0.24
	MI	99.04	98.10	99.91	98.01	0.09	98.57		
Average		99.43	99.00	99.00	98.01	1.00	99.22		

The bold entries represent the average value of performance measures achieved using the proposed models, CNN, CNN-LSTM, and Ensemble method

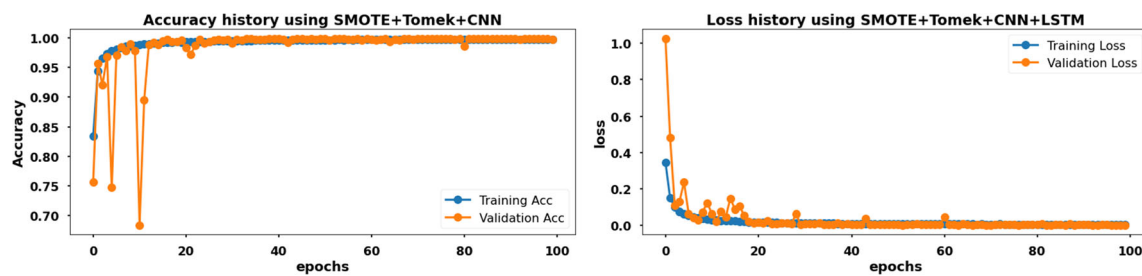
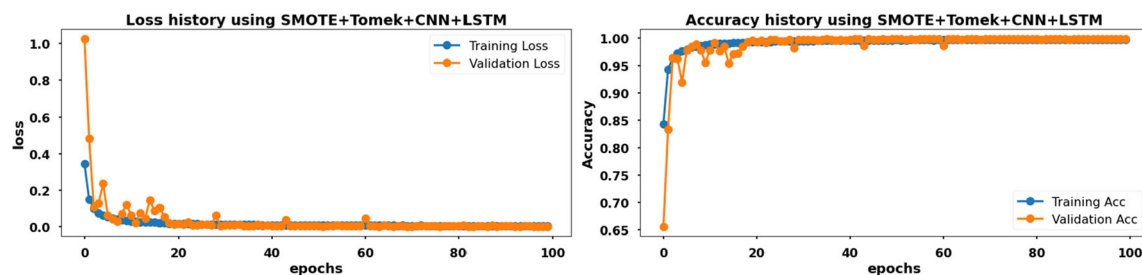
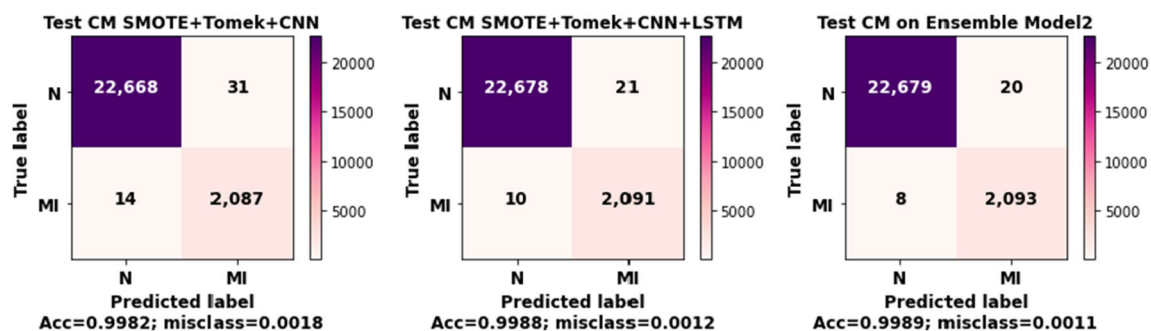
**Fig. 12** Plots of accuracy versus epochs and loss versus epochs obtained using the CNN model for the balanced dataset**Fig. 13** Plots of accuracy versus epochs and loss versus epochs obtained using CNN-LSTM model for the balanced dataset**Fig. 14** Confusion matrix on the test dataset using proposed models trained on balanced dataset

Table 5 Performance measures obtained using our proposed models with balanced test dataset

Method	Class	%Pr	%Re	%Sp	%YI	%FPR	%F1	%Acc	%CER
CNN	N	99.94	99.86	99.33	99.20	0.67	99.90	99.82	0.18
	MI	98.54	99.33	99.86	99.20	0.14	98.93		
Average		99.24	99.60	99.60	99.20	0.40	99.42		
CNN + LSTM	N	99.96	99.91	99.52	99.43	0.48	99.93	99.88	0.13
	MI	99.01	99.52	99.91	99.43	0.09	99.26		
Average		99.48	99.72	99.72	99.43	0.28	99.60		
ENSEMBLE	N	99.96	99.91	99.62	99.53	0.38	99.94	99.89	0.11
	MI	99.05	99.62	99.91	99.53	0.09	99.34		
Average		99.51	99.77	99.77	99.53	0.23	99.64		

and also to maintain the heterogeneity. Since MITDB does not contain the MI subjects and also has the issue of data imbalance [58], we have combined it with PTBDB to resolve it by applying the data oversampling technique.

The obtained results using our proposed technique are compared with state-of-the-art techniques in Table 7. In the comparison table, we have not included the type of ECG lead and types of MI classes. All these studies have been performed using different ECG leads.

It can be noted from Table 7 that, authors in [53] have considered the data imbalance issue on the PTB database. Authors in [9] have used huge ECG beats (611,405) and achieved great accuracy. Furthermore, most of the authors have utilized the PTB dataset except [30] and [4], also the types of ECG beats are different (multi-lead to a single lead. It is also noted that majority and minority class accuracy has not been considered by these studies. In this work, we have proposed CNN and CNN-LSTM models to detect MI using two benchmarked MITDB and PTBDB databases. Only [9] have used more ECG beats than this work, but they used 12 lead ECG data, and also data imbalance problem is not considered in their study. We have employed the SMOTE + Tomek link data resampling technique to resolve the imbalanced class problem. The data imbalance problem does not impact much on the overall accuracy of the classifier and can be visualized from the individual class accuracy (recall). The two separate experiments: 1st on the original imbalance

dataset and 2nd on the SMOTE + Tomek link balanced dataset, are performed using two proposed models CNN and CNN-LSTM respectively. We have obtained an overall accuracy of 99.89 % using our CNN-LSTM ensemble model using subject-wise testing.

The main advantages of our work are as follows:

- Two separate deep learning models CNN and CNN-LSTM with ensemble techniques are proposed.
- Publically available two benchmark datasets with 123,998 ECG beats are used to develop the model.
- The overall accuracy of 99.9 % is obtained for both the proposed models even with the imbalanced dataset.
- The imbalance data issue is resolved by applying the SMOTE + Tomek link data resampling technique.
- There is a significant improvement in the individual minority class accuracy with our technique.

The main disadvantage of our work is that we have not tested our model using cross-validation techniques and it has not yet been tested with hardware compatibility.

6 Conclusions

In this work, we have presented two novel deep learning models CNN and hybrid CNN-LSTM along with ensemble

Table 6 comparison of total training time taken by the models for individual experiments

DL Model	Number of Layers	Training Beats	Optimizer	Epochs	Total Training Time
CNN	21	99, 198	Adam	100	21 min 3 s
CNN-LSTM	23	99, 198			23 min 44 s
CNN with SMOTE + Tomek Link	21	181,426			37 min 11 s
CNN-LSTM with SMOTE + Tomek Link	23	181,426			38 min 5 s

Table 7 Summary of proposed state-of-the-art techniques developed for automated MI detection using ECG signals

Literature	Detection Methods	Database	Number of ECG beats	Accuracy (%)
(Acharya et al. 2016) [9]	KNN	PTBDB	611,405	98.8
(Kumar et al. 2017) [5]	LS-SVM	PTBDB	50,674	99.31
(Acharya et al. 2017) [3]	CNN	PTBDB	50,674	95.22
(Kora, 2017) [25]	FFPSO	PTBDB	2,806	96.7
(Liu, Zhang, et al. 2018) [18]	M-CNN	PTBDB	34,769	96.00
(Dohare et al. 2018) [14]	SVM	PTBDB	-----	96.66
(Liu, Huang, et al. 2018) [49]	MFB-CNN	PTBDB	59,336	94.82
(Lui & Chow, 2018) [4]	CNN-LSTM	PTBDB, AF Challenge 17	-----	97.7 (Sp)
(Sharma et al. 2018) [27]	KNN	PTBDB	50,728	99.74
(Sharma and Sunkaria, 2018) [50]	SVM	PTBDB	6,277	98.84
(Sadhukhan et al. 2018) [51]	LR	PTBDB	20,000	95.6
(Savostin et al. 2019) [52]	KNN	PTBDB	13,905	97.03
(Kayikcioglu et al. 2020) [30]	KNN	EDB, MITDB, LSTDB	111,688	94.23
(Sharma and Sunkaria, 2020) [53]	KNN	PTBDB	15,581	99
(Alghamdi et al. 2020) [54]	CNN	PTBDB	101,456	99.02
	QG-MSVM			99.22
(Jafarian et al. 2020) [55]	SNN	PTBDB	5,968	98.21
(Lin et al. 2020) [56]	KNN	PTBDB	28,700	99.57
(Sridhar et al. 2020) [59]	KNN, SVM, PNN, DT	PTBDB	-----	97.96
(Heo et al. 2020) [57]	KNN	PTBDB	310, 272	96.37
Proposed	CNN	PTBDB + MITDB	123, 998	99.82
	CNN-LSTM,			99.88
	Ensemble			99.89

techniques for automatic and accurate detection of MI. The data imbalance problem with minority class is tackled by applying the SMOTE + Tomek link resampling technique. We have utilized two benchmarked datasets MIT-BIH and PTB and extracted 123,998 ECG beats obtained for the verification of the experiment. We have performed two independent experiments, one is on the original dataset without applying any sampling technique and another one is on the SMOTE + Tomek link sampled dataset. We have achieved 99.89 % classification accuracy using ensemble technique and 99.82 and 99.88 % accuracy using proposed CNN and hybrid CNN-LSTM technique, respectively, on Smote + tomek link sampled dataset. Our models along with the ensemble technique do not require any additional processing such as feature extraction or preprocessing, hence it is an economical and less complex technique. In the future, we can use our developed model for MI localization, detect the severity of MI and cardiac arrhythmias using ECG signals. In future work, we may use a generative adversarial network (GAN) for generating

synthetic datasets and compare the result with the data balancing technique.

References

1. Kulick DL, Marks JW, Davis CP (2014) Heart attack (Myocardial Infarction), Medinet.Com, 1–14. <https://my.clevelandclinic.org/health/diseases/16818-heart-attack-myocardial-infarction>. Accessed 28 Aug 2020
2. Acharya UR, Fujita H, Sudarshan VK, Oh SL, Adam M, Tan JH, Koo JH, Jain A, Lim CM, Chua KC (2017) Automated characterization of coronary artery disease, myocardial infarction, and congestive heart failure using contourlet and shearlet transforms of electrocardiogram signal. *Knowl-Based Syst* 132:156–166. <https://doi.org/10.1016/j.knosys.2017.06.026>
3. Acharya UR, Fujita H, Oh SL, Hagiwara Y, Tan JH, Adam M (2017) Application of deep convolutional neural network for

- automated detection of myocardial infarction using ECG signals, *Inf Sci (NY)* 415–416:190–198. <https://doi.org/10.1016/j.ins.2017.06.027>
4. Lui HW, Chow KL (2018) Multiclass classification of myocardial infarction with convolutional and recurrent neural networks for portable ECG devices. *Inform Med Unlocked* 13:26–33. <https://doi.org/10.1016/j.imu.2018.08.002>
5. Kumar M, Pachori RB, Acharya UR (2017) Automated diagnosis of myocardial infarction ECG signals using sample entropy in flexible analytic wavelet transform framework. *Entropy* 19:488. <https://doi.org/10.3390/e19090488>
6. Frellick M, Vega CP (2020) How big a problem is misdiagnosis in medicine?, *Medscape*. <https://www.medscape.org/viewarticle/933116>. Accessed 14 Jul 2020
7. Tripathy RK, Bhattacharyya A, Pachori RB (2019) A novel approach for detection of myocardial infarction from ECG signals of multiple electrodes. *IEEE Sensors J* 19:4509–4517. <https://doi.org/10.1109/JSEN.2019.2896308>
8. Han C, Shi L (2019) Automated interpretable detection of myocardial infarction fusing energy entropy and morphological features. *Comput Methods Programs Biomed* 175:9–23. <https://doi.org/10.1016/j.cmpb.2019.03.012>
9. Acharya UR, Fujita H, Sudarshan VK, Oh SL, Adam M, Koh JEW, Tan JH, Ghista DN, Martis RJ, Chua CK, Poo CK, Tan RS (2016) Automated detection and localization of myocardial infarction using electrocardiogram: A comparative study of different leads. *Knowl-Based Syst* 99:146–156. <https://doi.org/10.1016/j.knosys.2016.01.040>
10. Acharya R, Bhat UPS, Kannathal N, Rao A, Choo ML (2005) Analysis of cardiac health using fractal dimension and wavelet transformation. *Itbm-Rbm* 26:133–139. <https://doi.org/10.1016/j.rbmret.2005.02.001>
11. Jayachandran ES, Joseph PK, Acharya RU (2010) Analysis of myocardial infarction using discrete wavelet transform. *J Med Syst* 34:985–992. <https://doi.org/10.1007/s10916-009-9314-5>
12. Fu J, Yang Y, Singhrao K, Ruan D, Chu FI, Low DA, Lewis JH (2019) Deep learning approaches using 2D and 3D convolutional neural networks for generating male pelvic synthetic computed tomography from magnetic resonance imaging. *Med Phys* 46:3788–3798. <https://doi.org/10.1002/mp.13672>
13. Acharya UR, Fujita H, Adam M, Lih OS, Sudarshan VK, Hong TJ, Koh JE, Hagiwara Y, Chua CK, Poo CK, San TR (2017) Automated characterization and classification of coronary artery disease and myocardial infarction by decomposition of ECG signals: A comparative study. *Inf Sci (NY)* 377:17–29. <https://doi.org/10.1016/j.ins.2016.10.013>
14. Dohare AK, Kumar V, Kumar R (2018) Detection of myocardial infarction in 12 lead ECG using support vector machine. *Appl Soft Comput J* 64:138–147. <https://doi.org/10.1016/j.asoc.2017.12.001>
15. Acharya R, Kannathal UN, Hua LM, Yi LM (2005) Study of heart rate variability signals at sitting and lying postures. *J Bodyw Mov Ther* 9:134–141. <https://doi.org/10.1016/j.jbmt.2004.04.001>
16. Plawiak P, Acharya UR (2020) Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals. *Neural Comput Appl* 32:11137–11161. <https://doi.org/10.1007/s00521-018-03980-2>
17. Gupta V, Mittal M (2020) Efficient R-peak detection in electrocardiogram signal based on features extracted using Hilbert transform and burg method. *J Inst Eng Ser B* 101:23–34. <https://doi.org/10.1007/s40031-020-00423-2>
18. Liu W, Zhang M, Zhang Y, Liao Y, Huang Q, Chang S, Wang H, He J (2018) Real-time multilead convolutional neural network for myocardial infarction detection. *IEEE J Biomed Health Inform* 22:1434–1444. <https://doi.org/10.1109/JBHI.2017.2771768>
19. Baloglu UB, Talo M, Yildirim O, Tan RS, Acharya UR (2019) Classification of myocardial infarction with multi-lead ECG signals and deep CNN. *Pattern Recognit Lett* 122:23–30. <https://doi.org/10.1016/j.patrec.2019.02.016>
20. Acharya UR, Fujita H, Oh SL, Hagiwara Y, Tan JH, Adam M, Tan RS (2019) Deep convolutional neural network for the automated diagnosis of congestive heart failure using ECG signals. *Appl Intell* 49:16–27. <https://doi.org/10.1007/s10489-018-1179-1>
21. Han C, Shi L (2020) ML-ResNet: A novel network to detect and locate myocardial infarction using 12 leads ECG. *Comput Methods Prog Biomed* 185:105138. <https://doi.org/10.1016/j.cmpb.2019.105138>
22. Xu C, Xu L, Brahm G, Zhang H, Li S (2018) MuTGAN: Simultaneous segmentation and quantification of myocardial infarction without contrast agents via joint adversarial learning. In: *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Springer Verlag, Berlin, pp 525–534. https://doi.org/10.1007/978-3-030-00934-2_59
23. Zhang J, Lin F, Xiong P, Du H, Zhang H, Liu M, Hou Z, Liu X (2019) Automated detection and localization of myocardial infarction with staked sparse autoencoder and treebagger. *IEEE Access* 7:70634–70642. <https://doi.org/10.1109/ACCESS.2019.2919068>
24. Rahhal MMA, Bazi Y, Alhichri H, Alajlan N, Melgani F, Yager RR (2016) Deep learning approach for active classification of electrocardiogram signals. *Inf Sci (NY)* 345:340–354. <https://doi.org/10.1016/j.ins.2016.01.082>
25. Kora P (2017) ECG based myocardial infarction detection using hybrid firefly algorithm. *Comput Methods Prog Biomed* 152:141–148. <https://doi.org/10.1016/j.cmpb.2017.09.015>
26. Acharya UR, Fujita H, Lih OS, Adam M, Tan JH, Chua CK (2017) Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network. *Knowl-Based Syst* 132:62–71. <https://doi.org/10.1016/j.knosys.2017.06.003>
27. Sharma M, Tan RS, Acharya UR (2018) A novel automated diagnostic system for classification of myocardial infarction ECG signals using an optimal biorthogonal filter bank. *Comput Biol Med* 102:341–356. <https://doi.org/10.1016/j.compbimed.2018.07.005>
28. Faust O, Shenfield A, Kareem M, San TR, Fujita H, Acharya UR (2018) Automated detection of atrial fibrillation using long short-term memory network with RR interval signals. *Comput Biol Med* 102:327–335. <https://doi.org/10.1016/j.compbimed.2018.07.001>
29. Fujita H, Cimr D (2019) Computer aided detection for fibrillations and flutters using deep convolutional neural network. *Inf Sci (NY)* 486:231–239. <https://doi.org/10.1016/j.ins.2019.02.065>
30. Kayikcioglu İ, Akdeniz F, Köse C, Kayikcioglu T (2020) Time-frequency approach to ECG classification of myocardial infarction. *Comput Electr Eng* 84. <https://doi.org/10.1016/j.compeleceng.2020.106621>
31. Liu W, Wang F, Huang Q, Chang S, Wang H, He J (2020) MFB-CBRNN: A hybrid network for MI detection using 12-Lead ECGs. *IEEE J Biomed Health Inform* 24:503–514. <https://doi.org/10.1109/JBHI.2019.2910082>
32. Physionet MIT-BIH (2005) Arrhythmia Database-v1. <https://doi.org/10.13026/C2F30500>
33. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng C, Stanley HE (2000) PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 101:23. <https://doi.org/10.1161/01.CIR.101.23.e215>
34. Bousseljot AN, Kreiseler R, Schnabel D (1995) The PTB diagnostic ECG database. *Biomed Tech* 40:317. <https://doi.org/10.13026/C28C71>
35. Fazeli S (2018) ECG heartbeat categorization dataset, Kaggle. <https://www.kaggle.com/shayanfazeli/heartbeat>. Accessed 21 Aug 2020
36. Kachuee M, Fazeli S, Sarrafzadeh M (2018) ECG heartbeat classification: A deep transferable representation. In: *Proc. – 2018 IEEE*

- Int. Conf. Healthc. Informatics, ICHI 2018, pp 443–444. <https://doi.org/10.1109/ICHI.2018.00092>
37. Madasamy K, Ramaswami M (2017) Data imbalance and classifiers: impact and solutions from a big data perspective. *International Journal of Computational Intelligence Research*, Volume 13, Number 9 (2017), pp. 2267–2281. Accessed August 23, 2020
 38. Younes C (2019) Resampling to properly handle imbalanced datasets in machine learning |. *Heartbeat*. <https://heartbeat.fritz.ai/resampling-to-properly-handle-imbalanced-datasets-in-machine-learning-64d82c16ceaa>. Accessed 28 Aug 2020
 39. Gonzalez CI, Melin P, Castro JR, Castillo O (2017) Metrics for edge detection methods. *SpringerBriefs Appl Sci Technol*:17–19. https://doi.org/10.1007/978-3-319-53994-2_4
 40. Jahani Heravi E, Habibi Aghdam H, Puig D (2018) An optimized convolutional neural network with bottleneck and spatial pyramid pooling layers for classification of foods. *Pattern Recognit Lett* 105: 50–58. <https://doi.org/10.1016/j.patrec.2017.12.007>
 41. Escontrela A (2020) Convolutional neural networks from the ground up - Towards data science, Towardsdatascience.Com. <https://towardsdatascience.com/convolutional-neural-networks-mathematics-1beb3e6447c0>. Accessed 8 Dec 2020
 42. Kabir Anaraki A, Ayati M, Kazemi F (2019) Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybern Biomed Eng* 39:63–74. <https://doi.org/10.1016/j.bbe.2018.10.004>
 43. Phi M (2018) Illustrated Guide to LSTM's and GRU's: A step by step explanation, Medium.Com, 1–15. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. Accessed 23 Aug 2020
 44. Omkar N (2019) Activation functions with derivative and python code: Sigmoid Vs Tanh Vs Relu. <https://medium.com/@omkar.nallagoni/activation-functions-with-derivative-and-python-code-sigmoid-vs-tanh-vs-relu-44d23915c1f4>. Accessed 23 Aug 2020
 45. Lih OS, Jahmunah V, San TR, Ciaccio EJ, Yamakawa T, Tanabe M, Kobayashi M, Faust O, Acharya UR (2020) Comprehensive electrocardiographic diagnosis based on deep learning. *Artif Intell Med* 103:101789. <https://doi.org/10.1016/j.artmed.2019.101789>
 46. Lutins E (2017) Ensemble methods in machine learning: What are they and why use them? | by Evan Lutins | Towards Data Science, Towardsdatascience.Com <https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f>. Accessed 11 Dec 2020
 47. Brownlee J (2016) What is a confusion matrix in machine learning, Machinelearningmastery. <https://machinelearningmastery.com/confusion-matrix-machine-learning/>. Accessed 7 Dec 2020
 48. Leonel J (2019) Hyperparameters in machine/deep learning, Medium.Com <https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981>. Accessed 23 May 2020
 49. Liu W, Huang Q, Chang S, Wang H, He J (2018) Multiple-feature-branch convolutional neural network for myocardial infarction diagnosis using electrocardiogram. *Biomed Signal Process Control* 45:22–32. <https://doi.org/10.1016/j.bspc.2018.05.013>
 50. Sharma LD, Sunkaria RK (2018) Inferior myocardial infarction detection using stationary wavelet transform and machine learning approach. *Signal Image Video Process* 12:199–206. <https://doi.org/10.1007/s11760-017-1146-z>
 51. Sadhukhan D, Pal S, Mitra M (2018) Automated identification of myocardial infarction using harmonic phase distribution pattern of ECG Data. *IEEE Trans Instrum Meas* 67:2303–2313. <https://doi.org/10.1109/TIM.2018.2816458>
 52. Savostin AA, Ritter DV, Savostina GV (2019) Using the K-nearest neighbors algorithm for automated detection of myocardial infarction by electrocardiogram data entries. *Pattern Recognit Image Anal* 29:730–737. <https://doi.org/10.1134/S1054661819040151>
 53. Sharma LD, Sunkaria RK (2020) Myocardial infarction detection and localization using optimal features based lead specific approach. *Irbm* 41:58–70. <https://doi.org/10.1016/j.irbm.2019.09.003>
 54. Alghamdi A, Hammad M, Ugail H, Abdel-Raheem A, Muhammad K, Khalifa HS (2020) A.A. Abd El-Latif, Detection of myocardial infarction based on novel deep transfer learning methods for urban healthcare in smart cities. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-020-08769-x>
 55. Jafarian K, Vahdat V, Salehi S, Mobin M (2020) Automating detection and localization of myocardial infarction using shallow and end-to-end deep neural networks. *Appl Soft Comput J* 93:106383. <https://doi.org/10.1016/j.asoc.2020.106383>
 56. Lin Z, Gao Y, Chen Y, Ge Q, Mahara G, Zhang J (2020) Automated detection of myocardial infarction using robust features extracted from 12-lead ECG. *Signal Image Video Process* 14:857–865. <https://doi.org/10.1007/s11760-019-01617-y>
 57. Heo J, Lee JJ, Kwon S, Kim B, Hwang SO, Yoon YR (2020) A novel method for detecting ST segment elevation myocardial infarction on a 12-lead electrocardiogram with a three-dimensional display. *Biomed Signal Process Control* 56:101700. <https://doi.org/10.1016/j.bspc.2019.101700>
 58. Yildirim O, Talo M, Ciaccio EJ, Tan RS, Acharya UR (2020) Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ECG Records. *Comput Methods Prog Biomed* 197:105740. <https://doi.org/10.1016/j.cmpb.2020.105740>
 59. Sridhar C, Lih OS, Jahmunah V, Koh JEW, Ciaccio EJ, San TR, Arunkumar N, Kadry S (2020) U. Rajendra Acharya, Accurate detection of myocardial infarction using non linear features with ECG signals. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-02536-4>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.