Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Cardi-Net: A deep neural network for classification of cardiac disease using phonocardiogram signal

Juwairiya Siraj Khan[a], Manoj Kaushik[a], Anushka Chaurasia[a], Malay Kishore Dutta[a,*], Radim Burget[b]

[a] *Centre for Advanced Studies, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India*

[b] *Department of telecommunications, Faculty of Electrical engineering and communication, Brno University of Technology, Brno, Czech Republic*

ARTICLE INFO

Article history:
Received 26 October 2021
Revised 18 February 2022
Accepted 28 February 2022

Keywords:
Phonocardiogram
Data augmentation
Power spectrogram
Deep learning
Cardiac disorders

ABSTRACT

Background and objectives: The lack of medical facilities in isolated areas makes many patients remain aloof from quick and timely diagnosis of cardiovascular diseases, leading to high mortality rates. A deep learning based method for automatic diagnosis of multiple cardiac diseases from Phonocardiogram (PCG) signals is proposed in this paper.

Methods: The proposed system is a combination of deep learning based convolutional neural network (CNN) and power spectrogram Cardi-Net, which can extract deep discriminating features of PCG signals from the power spectrogram to identify the diseases. The choice of Power Spectral Density (PSD) makes the model extract highly discriminatory features significant for the multi-classification of four common cardiac disorders.

Results: Data augmentation techniques are applied to make the model robust, and the model undergoes 10-fold cross-validation to yield an overall accuracy of 98.879% on the test dataset to diagnose multi heart diseases from PCG signals.

Conclusion: The proposed model is completely automatic, where signal pre-processing and feature engineering are not required. The conversion time of power spectrogram from PCG signals is very low range from 0.10 s to 0.11 s. This reduces the complexity of the model, making it highly reliable and robust for real-time applications. The proposed architecture can be deployed on cloud and a low cost processor, desktop, android app leading to proper access to the dispensaries in remote areas.

1. Introduction

Cardiovascular disease (CVD) is a collective term used for a number of blood vessels or heart-related disorders, which generally includes rheumatic heart disease, cerebrovascular disease, coronary heart disease, etc. The severity of CVDs may be as high as in cases such as heart attacks or strokes leading to premature deaths in four out of five cases. According to World Health Organization, CVDs are ranked first as the cause of death worldwide, with an estimation of about 17.9 million deaths every year. CVDs are held accountable for taking more than 75% of lives in low and middle-income countries [1]. The preliminary diagnosis of heart diseases can be effectively and conveniently made using Phonocardiogram (PCG) signals. Therefore, early diagnosis of cardiac diseases is necessary, mainly in isolated areas with a shortage of primary health facilities and experienced medical examiners.

Millions of people in the world are not aware that they are vulnerable to CVDs. An early-stage diagnosis of patients can avert many strokes and heart attacks. In recent years, attempts have been made to enable semi-automatic or fully automatic diagnosis of CVDs for implementation in primary health care facilities. Heart sound-based cardiac disorders diagnosis is the most common out of many other ways used by medical professionals. Auscultation is a primary method for detecting and diagnosing various heart-related conditions. Many mechanical events have also been noted during each heart cycle, including the production of various cardiac sounds. The heart cycle primarily includes two types of sounds: S1 (first cardiac sound) and S2 (second cardiac sound). There is a short gap between S1 and S2 sounds during which the heart contracts, known as ventricular systole, while the short gap between S2 and S1 is indicative of the relaxation phase of the heart known as ventricular diastole. External sounds named S3 and S4,

[*] Corresponding author.
E-mail addresses: jury.sirajkhan@gmail.com (J.S. Khan), malaykishoredutta@gmail.com (M.K. Dutta).

https://doi.org/10.1016/j.cmpb.2022.106727
0169-2607/© 2022 Elsevier B.V. All rights reserved.

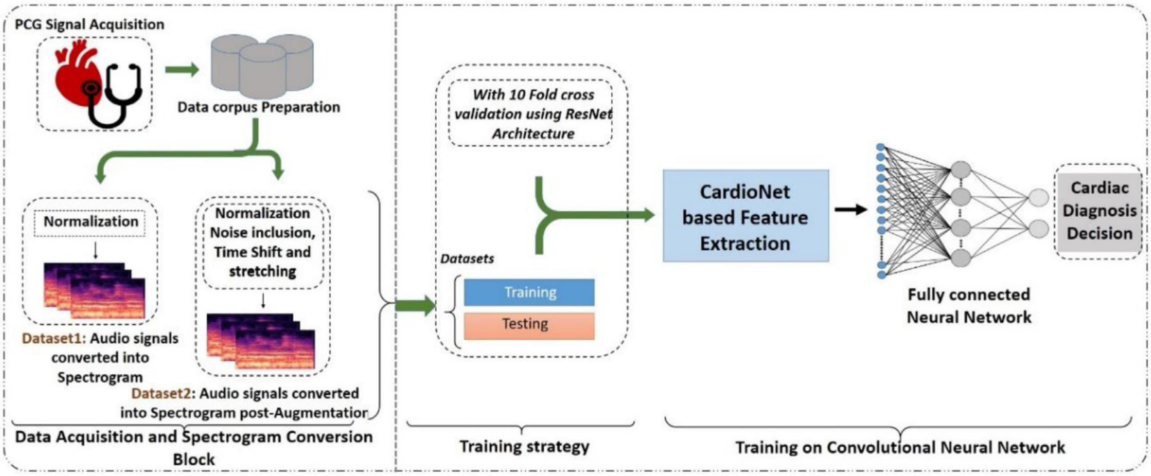
PCG Signal Acquisition
Data corpus Preparation
Normalization
Normalization Noise inclusion, Time Shift and stretching
Dataset1: Audio signals converted into Spectrogram
Dataset2: Audio signals converted into Spectrogram post-Augmentation
Data Acquisition and Spectrogram Conversion Block
With 10 Fold cross validation using ResNet Architecture
Datasets
Training
Testing
Training strategy
CardioNet based Feature Extraction
Fully connected Neural Network
Cardiac Diagnosis Decision
Training on Convolutional Neural Network

Fig. 1. Methodology of the proposed work.

such as snaps, clicks and murmurs, are found in abnormal cardiac sounds. The heart sounds are produced in a periodic pattern for people with no cardiac illness but if it deviates from the normal pattern is an indication of some sort of cardiovascular disease [3]. The conventional approach used by doctors involves listening to the heart sounds using a stethoscope and interpreting the audio signals manually and subjectively. Alternatively, the cardiac audio signals can be recorded using an electronic stethoscope. A phonocardiogram graph can be plotted, later interpreted by medical experts to diagnose various cardiac disorders.

Most machine learning techniques were applied to heart sound classification and CVD detection in the earlier works. Dokur and Olmez present a novel method for classifying different cardiac sounds [2]. Discrete wavelet transform has been applied to windowed one cycle of the heart sound. Vepa [3] investigated features derived from the cepstrum of heart sound and used them to train three classifiers: k-nearest neighbour classifier (k-NN), multilayer perceptron neural network, and support vector machine the classification of heart sounds into normal, systolic murmurs and diastolic murmurs. A hidden Markov model (HMM) has been utilized to classify heart sounds by varying the number of states, number of mixtures and analyzing the frame size of the model [4]. Several machine learning methods have been employed previously for the heart sound feature extraction, and classification, such as empirical wavelet transform [7], combined spectral amplitude and wavelet entropy [8], HMM [9], support vector machine (SVM) [14–16], twin SVM [11], k-NN and random forest [24] and some deep convolutional neural network (CNN) [10]. Spectrograms have been used earlier with wavelets [12] and frequency cepstrum coefficients [13]. The main problem in early-stage detection of CVDs is the time taken during pre-processing of the signals and the requirement of feature engineering, which increases the signal processing time and the system's complexity, making the real-time implementation a challenge. This paper has resolved this using power spectrogram by eradicating the pre-processing and feature engineering of signals, reducing processing time and making a generalized model through PCG augmentation.

The main contribution of this paper lies in the design of an AI-based model for the diagnosis of crucial multi heart disorders by utilizing a deep learning-based residual neural network model with power spectrograms of the PCG audio samples as input to the model. The Cardi-Net model exploited for the classification and diagnosis of major cardiac diseases is suitable for real-time implementation, and it is highly robust, especially in a noisy environment. Additionally, it does not require feature engineering methods and eliminates the pre-processing complexity. The requirement of experienced professionals is not necessary with this model deployed in a device for screening multi heart-related disorders. In the purview of the proposed work, the implementation of an audio data augmentation technique for PCG signals, based on 1D signal-augmentation theory, enables the developing of a robust cardiac disorder detection system and a highly generalized model. The 1D data is converted into 2D images using a power spectrogram that requires a low conversion time, making the complete operation very fast. The mechanism involved in the classification of multi-cardiac disorders achieved high accuracy. The presented Cardi-Net model is suitable for real-time applications in remote areas since it is automatic and has minimal computational demands.

The remaining sections of the paper are organized as follows: Section 2 describes the main method of the proposed approach by detailing the dataset collection that is utilized for training the model, data augmentation techniques and data conversion from 1D signals into spectrogram images Section 3. illustrates the proposed Cardi-Net architecture for the multi-classification of cardiovascular diseases Section 4. discusses the experimental results acquired by training and testing the presented model Section 5. reports the real-time implementation of the model by porting it into a DSP processor and its integration on the cloud. The conclusion and future scope of work are elucidated in Section 6.

2. Method

The prime aim of the proposed work is the development of a deep learning classification model which can enable automatic diagnosis of severe cardiovascular diseases with the help of PCG based heart sounds. This can be achieved by an amalgamation of power spectrogram and Cardi-Net convolutional neural network. The pictorial representation of the proposed model is shown using a block diagram in Fig. 1.

The proposed method is divided into three major blocks. The first block indicates data acquisition and spectrogram conversion. Firstly, audio heart samples of PCG signals are acquired from subjects or dataset followed by data corpus preparation. These blocks also include data augmentation and conversion of audio signals to power spectrogram. Two different spectrogram datasets are being prepared, one without augmentation and the other having the combination of with and without augmentation. The second block focuses on mainly the training strategy. The acquired spectrogram datasets from the first block are divided into two groups in the ratio of 9:1. The former is for the training the model, while the

Table 1
Cardiac audio dataset used in the experiment (raw and augmented).

Heart diseases	Number of samples	Total samples (after augmentation)
Normal	200	400
Aortic Stenosis	200	400
Mitral Regurgitation	200	400
Mitral Stenosis	200	400
Mitral Valve Prolapse	200	400
Total	1000	2000

latter is for the testing the proposed model using 10-fold cross-validation. The last block is for the Cardi-Net CNN, which includes the proposed CNN architecture for multi-classification of cardiac disorders useful for the early diagnosis of four main types of heart-related diseases with the help of the ground truth created by medical professionals.

2.1. Dataset

The dataset used in this work is a publicly available PCG heart sound database [26]. On the basis of PCG signals, the cardiac sound has been categorized into two prime groups, i.e. normal and abnormal groups. The data in these two categories is divided in the ratio of 1:4, with the former containing the recordings from healthy subjects while the latter representing the recordings of subjects suffering from four different types of major valvular diseases which constitutes a total of five different classes. All the recordings are from different subjects. Any cardiovascular condition affecting one or more of the heart's four valves is known as valvular heart disease. A detailed description of the aforementioned dataset has been made in Table 1. The audio signal acquisition was carried out in three main steps, namely: valvular disease category selection, then data collection, and finally signal filtering accompanied with data standardization. The dataset is given in .wav format with 1000 audio samples, with each class having 200 samples and only one channel having 16 bits per sample. It has a bit rate of 128 kbps and a sampling rate of 8000 Hz. The time duration of the recordings varies between 1.4 s and 2.5 s and most of the recordings are close to 1.4 s.

2.2. Data augmentation

The limitation in the number of samples in a dataset may lead to a problem of overfitting when dealing with the training of deep neural networks. This overfitting can be reduced by increasing the size of the dataset with the exploitation of a technique inspired by the method of audio-augmentation, called data augmentation. Data augmentation enables the generation of synthetic data using the existing data. Many augmentation techniques are used for audio signals such as stretching, noise injection, background deformation, time-shifting, changing pitch and changing speed. However, in this work, only three audio augmentation techniques have been used mentioned as follows:

1. Noise injection in data: Noise injection simply involves the addition of some random values into the data. This technique is chosen considering the fact that there may be some reasonable amount of noise during PCG signal recording by medical professionals. Noise injection will enable the working of the proposed system efficiently in real-time applications. Additive white Gaussian noise (AWGN) is a basic noise model used in information theory to mimic the effect of many random processes that occur in nature. AWGN is often used as a channel model in which the only impairment to communication is a linear addition of wideband or white noise with a constant spectral density and a Gaussian distribution of amplitude. The obtained signal-to-noise ratio range is 16.6206 dB to 39.3522 dB. The SNR is calculated as the ratio of PCG signal average power and the noise average power. The average power of a signal is its power spectral density can be calculated by averaged amplitude of its Fast Fourior Transform (FFT) [27]. SNR is defined in dB as in equation 1:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathrm{S_{psd}}}{\mathrm{N_{psd}}}\right) \tag{1}$$

where, $\mathrm{S_{psd}}$ is the power spectral density of the PCG signal and $\mathrm{N_{psd}}$ is the power spectral density of the noise.

2. Time-shifting in data: Shifting time in audio signal typically means a shift of left or right with a random second. In case of left shifting of audio signal, i.e., fast forward, with a random value of x seconds, then initial x seconds will be marked as 0 or silence. Whereas in the case of right shifting of the audio signal, i.e., back forward, with a random value of x seconds, then final x seconds will be marked as 0 or silence.
3. Data stretching: This augmentation technique mainly alters the time duration of the input audio signal. It can enable signal stretching or compression without changing the pitch, as per the desired objective. Data stretching in heart sound used for mimicking the slow heartbeats and also to increase the size of the dataset for augmentation purposes. The aim of PCG signal stretching is to generate somewhat similar data with original slow heartbeat PCG signals so that the dataset size will be increased containing PCG signal variations so that trained deep learning model will be more generalized and robust. To perform the data stretching, all the original heartbeat signal is stretched 0.8 times on the time scale in this study.

Using the aforementioned audio data augmentation techniques, an additional dataset has been proposed for the training and validation set. The injected noise is Additive White Gaussian (AWG) noise. The AWG noise is intrinsic, has uniform power across the frequency band, normal distribution in the time domain and helps to make the audio signal more naturally recorded. The length of noise injected is the same as PCG signal length. As the main features are detectable by the model, its performance remains somewhat the same even without noise. Moreover, the noise inclusion makes the model more robust. This is due to the fact that in real-time applications, there is various background noise while recording the PCG signals. The detailed description of the dataset after the application of audio augmentation techniques has been shown in Table 1 and pictorially represented in Fig. 2. The dataset is in .wav format with 2000 audio samples, with each class having 400 samples in total.

2.3. PSD based power spectrogram conversion

The power spectral density, which distributes the signal power over frequency, constitutes the power spectrogram by considering small windows for a long time duration and then plotting them with respect to time associated with that window. The conversion of raw audio signals into power spectrograms has been accomplished as mentioned in the stepwise Algorithm 1 to convert

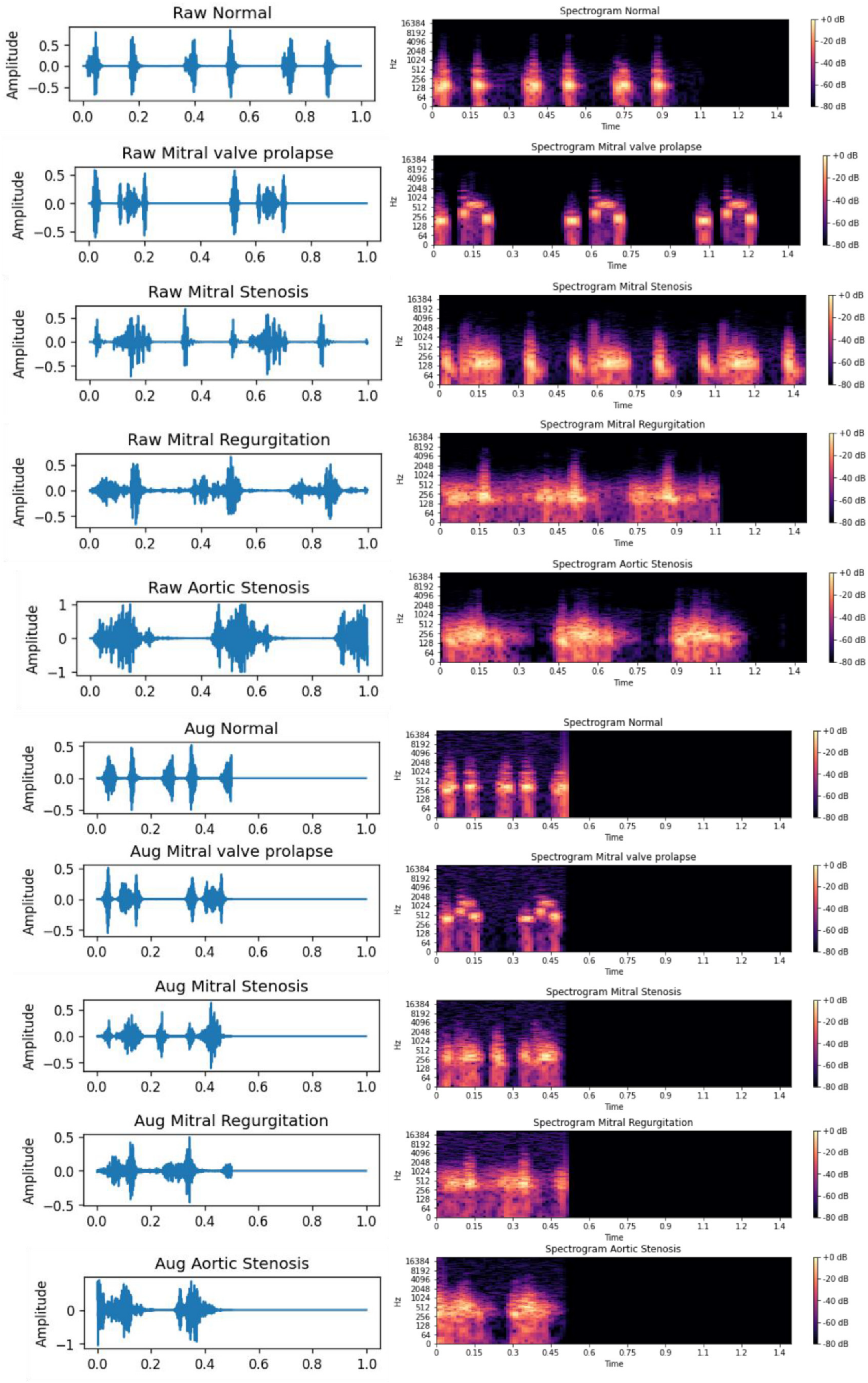

Fig. 2. Graphical representation of the conversion of PCG signals showing amplitude w.r.t time into spectrograms representing amplitude, frequency and time PCG signal graphs and their graphs after applying augmentation for: (a) Normal cardiac PCG signal, (b) Aortic Stenosis cardiac PCG signal (c) Mitral Regurgitation cardiac PCG signal (d) Mitral Stenosis cardiac PCG signal (e) Mitral valve prolapse cardiac PCG signal.

Algorithm 1: Conversion of raw audio signal into a spectrogram.

```
Input: Original PCG audio signal
Output: Power spectrogram

Start Procedure:
I.   Load the raw PCG audio file with a sampling rate of 44100 Hz.
II.  Trim the leading and trailing silence from the audio clip using a
     threshold value 60 decibels.
III. Normalize the trimmed audio clip.
IV.  Fix the normalized array length equivalent to 44100 either by trimming
     or padding
V.   Convert the above fixed normalized array into a complex-valued matrix
     which represents STFT matrix, keeping FFT window size 2048 and hop
     length 812, the equation for discrete STFT is given as:

     $STFT\{x(n)\}(m,\ \omega) = X(m,\ \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n - mR)e^{-j\omega n}$

     Where $x(n)$ is the input signal, $w(n)$ is the window function of length M,
     $X(m,\ \omega)$ is the DTFT of windowed data cantered about time mR and $R$ is
     the size of sample hop between successive DTFTs.
VI.  Convert STFT into dB-scaled STFT
```

the complete dataset and the augmented dataset into a dataset of spectrogram images using python librosa library [17] as shown in Fig. 2.

A short-time Fourier transform (STFT) method called Power spectrogram is used for the time-frequency analysis of the audio signals, where mono-audio clips are fed as the batch input in the algorithm. STFT is typically a Fourier related transform utilized to obtain the sinusoidal frequency component as well as the phase component of a local segment in a time-varying signal. The mathematical formulation to determine STFT of a signal when it happens to change over a time period requires the division of an elongated version of the time-varying signal into equally small sized sections by the exploitation of a window function. Later, the Fourier transform is applied to each section. The widely used window functions in STFT are Hann and Hamming window functions. As per the requirement of the proposed work, Hann window is most suited as it has less side lobes and the leakage is less in this windowing technique. Also, the Hann window removes any discontinuity by touching 'zero' at both ends, which was also a plus point. Hann window belongs to the family of generalized cosine window and power-of-sine, which can be expressed mathematically with window length L = T + 1, is mentioned below in Eq. (2):

$$w(t) = 0.5\left(1 - \cos\left(2\pi \frac{t}{T}\right)\right),\ 0 \le t \le T \tag{2}$$

A sampling rate of 44100 Hz is used with a Fast Fourier Transform window size of 2048 and the number of intermediate frames in each successive frame being 812, in order to determine STFT. The difference in energy levels can be easily visualized through power spectrograms for normal as well as cardiac disorder patients and these varying energy levels are the discriminant features in the proposed heart disease classification model using PCG audio signals. A detailed description is given in Algorithm 1. The window size is a trade-off between high resolution in time or frequency. The high resolution in frequency can be obtained if the window size is equal to the sample sequence of the signal. The window size for PCG signal is 2048, which is equal to the sample sequence. The variation of window sizes results in degraded frequency resolution of the final power spectrogram; hence it also affects the model's performance in extracting discriminating features. In the spectrogram, the x-axis has been kept constant for 1.4 s. For some of the recordings larger than 1.4 s, extra data is extracted and for recordings shorter than 1.4 s, zero padding is applied to make the spectrogram image size constant.

3. Proposed Cardi-Net architecture for CVD classification

This paper proposes a two-dimensional Cardi-Net architecture to diagnose various cardiac disorders as shown in Fig. 3. The architecture broadly comprises an Input data, four residual blocks and finally, a fully connected network that outputs the classification results. Power spectrograms are exploited for the input data after conversion from the raw audio PCG signals, depicted using Algorithm 1 in Section 2.3. The dimension of the single-channel spectrogram image is (1025×120×1). The time duration of the recordings varies between 1.4 s and 2.5 s, and most of the recordings are close to 1.4 s. So, a fixed length of the image is selected so that it contains most of the recordings with duration 1.4 s. The rest of the recordings are padded or cropped from the images. The model learns to find local patterns in the image using various feature filters. The feature extraction part of the proposed Cardi-Net architecture is a combination of a convolutional layer, max-pooling layer, batch normalization with appropriate hyperparameters, while the second part is a fully connected neural network that works as a classifier using extracted features as input. The architecture is built using Jupyter notebook, TensorFlow [18] and Keras [19] and librosa APIs in Python 3. The proposed Cardi-Net neural network architecture consists of 43 layers, including batch normalization, activation, convolution, flattening and dense layers.

The proposed architecture is basically a sequence wise, layered amalgamation of two-dimensional convolution, activation and normalization processes. Additionally, it is linked to a densely connected neural network which is accompanied by an activation function as per the requisites of the model. The convolution operation is regarded as the core of a CNN architecture where feature extraction occurs automatically in contrast to the old machine learning manual approaches, thereby saving time in the whole process. A kernel, which is a 2D window with a fixed size, slides over the input spectrogram resulting in a 2D matrix containing high-level features. This operation basically includes an element-by-element product succeeded by tensor addition. If the image formed by the spectrogram (S) has dimensions P×Q and the kernel window (K) has dimensions p×q, then the dimensions of the output image will be: $(P - p + 1) \times (Q - q + 1)$. The extraction of highly refined features out of the image spectrogram can be obtained with repetition of convolution on resulting feature maps with intermediate max-pooling operation. The representation of convoluted output between S and K is given by Eq. (3) and '*' donates the convolution operation.

$$S * K = \mathrm{Output}(i, j) = \sum_{k=1}^{p} \sum_{l=1}^{q} S(i + k - 1,\ j + l - 1)K(k, l) \tag{3}$$

Here, iteration of 'i' if from 1 to $(P - p + 1)$ while for 'j' it is from 1 to $(Q - q + 1)$.

The formation of a convolution layer basically happens by combining linear convolution operation and non-linear activation function. The sequencing of different layers, hyperparameters, and output shape are shown in Table 2. The training and validation of the proposed model is carried out using 10-fold cross validation for multiple epochs. Overfitting of the model is prevented by a technique called early stopping [20], so there is a difference in the number of epochs in each fold, while the model with minimum loss and best score on the basis of performance is saved Table 3. depicts all the tuned parameters of the proposed network architecture where the adjustment of weights supplied to the model takes place for every fold data of the power spectrogram.

It has been evinced in several image processing research works that the convolution neural networks are best recognized for their object recognition, feature extraction and classification accuracy,

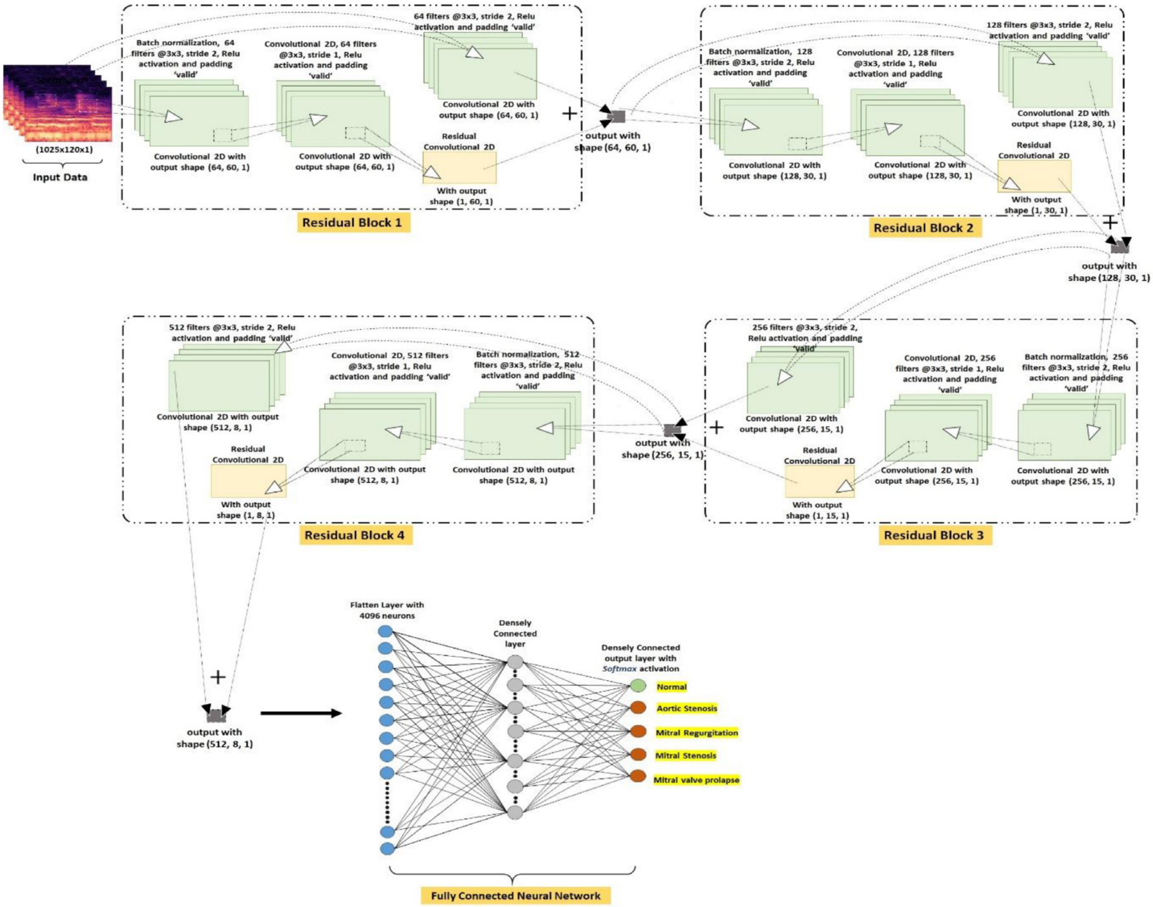
(1025x120x1)
Input Data
Batch normalization, 64 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D, 64 filters @3x3, stride 1, Relu activation and padding 'valid'
64 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D with output shape (64, 60, 1)
Convolutional 2D with output shape (64, 60, 1)
Convolutional 2D with output shape (64, 60, 1)
Residual Convolutional 2D
With output shape (1, 60, 1)
+
output with shape (64, 60, 1)
Residual Block 1
Batch normalization, 128 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D, 128 filters @3x3, stride 1, Relu activation and padding 'valid'
128 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D with output shape (128, 30, 1)
Convolutional 2D with output shape (128, 30, 1)
Convolutional 2D with output shape (128, 30, 1)
Residual Convolutional 2D
With output shape (1, 30, 1)
+
output with shape (128, 30, 1)
Residual Block 2
512 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D, 512 filters @3x3, stride 1, Relu activation and padding 'valid'
Batch normalization, 512 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D with output shape (512, 8, 1)
Convolutional 2D with output shape (512, 8, 1)
Convolutional 2D with output shape (512, 8, 1)
Residual Convolutional 2D
With output shape (1, 8, 1)
Residual Block 4
output with shape (256, 15, 1)
+
256 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D, 256 filters @3x3, stride 1, Relu activation and padding 'valid'
Batch normalization, 256 filters @3x3, stride 2, Relu activation and padding 'valid'
Convolutional 2D with output shape (256, 15, 1)
Convolutional 2D with output shape (256, 15, 1)
Convolutional 2D with output shape (256, 15, 1)
Residual Convolutional 2D
With output shape (1, 15, 1)
Residual Block 3
+
output with shape (512, 8, 1)
Flatten Layer with 4096 neurons
Densely Connected layer
Densely Connected output layer with Softmax activation
Normal
Aortic Stenosis
Mitral Regurgitation
Mitral Stenosis
Mitral valve prolapse
Fully Connected Neural Network

Fig. 3. The layered architecture of the proposed Cardi-Net model containing four residual blocks and one fully connected neural network.

Table 2
Tuned Hyperparameters of Cardi-Net Architecture.

Hyperparameter	Value
Loss	Categorical cross-entropy
Batch size	32
Epoch	500
Early Stopping	Monitoring the Validation accuracy
Activation function	ReLU
Padding	valid
Optimisation Algorithm	Adadelta
Learning Rate	0.01
Multiprocessing	True

Algorithm 2: Transformation of Batch Normalization, applied to activation x over a mini-batch B.

```
Input: Input activation x over a mini-batch: B= $\{x_{1...n}\}$;
       $\gamma, \beta$ are the parameters to be learned
Output: $y_i = Batch\_Normalization_{\gamma,\beta}(x_i)$

Start Procedure:
Step1: Calculate the mean of mini-batch
Step2: Calculate the variance of mini-batch
Step3: Normalize the mini-batch
Step4: Apply shifting and scaling operation
```

but the problem of vanishing gradient occurs while updating the weights. This can be resolved with the use of skip connections as in ResNet [21] through the exploitation of residual blocks. The Cardi-Net architecture is inspired by the ResNet model, which uses skip connections. In deep neural networks, it allows the gradient to flow through this alternate shortcut path, thereby solving the issue of vanishing gradient. It also enables the higher layers of the model to perform at least on a par with the lower layers. Thus, preventing the model's performance from degrading on addition of extra layers.

Batch normalization technique is used to normalize the output of the first convolution layer and boost up the overall training to make the network fast, light and stable. The shape of the image remains unaltered after batch normalization Algorithm 2. explains the pseudo code of batch normalization. The conversion of 2D data into a single feature vector of 1D, later supplied to the densely connected layer for making a fully-connected neural network, is accomplished with the deployment of a flattening layer. This layer is used as the input to the fully-connected neural network of the proposed Cardi-Net architecture and it basically gives the product of tensor dimension of the preceding layer as its output. Rectified Linear Unit (ReLU) and Softmax activation functions are utilized at various layers in this architecture. ReLU, a non-linear unsaturated activation function, performs better than the linear saturated activation functions such as sigmoid [22] and hyperbolic tangent (tanh). The window size for PCG signal is 2048, which is equal to the sample sequence.

ReLU function implemented in the primary convolution layers improves accuracy and reduces the training time. It is shown in Eq. (4).

$$\mathrm{ReLU}(x) = \begin{cases} x, \ if\ x > 0 \\ 0, \ if\ x \le 0 \end{cases} \tag{4}$$

Table 3
Cardi-Net model architecture with various layers and number of parameters.

No.	Architecture Layers	Output Shape	Kernel Size	Stride	Filters	Parameters
1	Input Layer	1025, 120, 1				0
2	Batch Normalization	1025, 120, 1	-	-	-	4
3	Activation	1025, 120, 1	-	-	-	0
4	Convolution 2D	64, 60, 1	3×3	2	64	590464
5	Batch Normalization	64, 60, 1	-	-	-	4
6	Activation	64, 60, 1	-	-	-	0
7	Convolution 2D	64, 60, 1	3×3	1	64	36928
8	Convolution 2D	64, 60, 1	1×1	1	1	590464
9	Convolution 2D	1, 60, 1	3×3	2	64	65
10	Add(conv2d_2[0][0])	64, 60, 1	-	-	-	0
11	Batch Normalization	64, 60, 1	-	-	-	4
12	Activation	64, 60, 1	-	-	-	0
13	Convolution 2D	128, 30, 1	3×3	2	128	73856
14	Batch Normalization	128, 30, 1	-	-	-	4
15	Activation	128, 30, 1	-	-	-	0
16	Convolution 2D	128, 30, 1	3×3	1	128	147584
17	Convolution 2D	128, 30, 1	1×1	1	1	73856
18	Convolution 2D	1, 30, 1	3×3	2	128	129
19	Add(conv2d_6[0][0])	128, 30, 1	-	-	-	0
20	Batch Normalization	128, 30, 1	-	-	-	4
21	Activation	128, 30, 1	-	-	-	0
22	Convolution 2D	256, 15, 1	3×3	2	256	295168
23	Batch Normalization	256, 15, 1	-	-	-	4
24	Activation	256, 15, 1	-	-	-	0
25	Convolution 2D	256, 15, 1	3×3	1	256	590080
26	Convolution 2D	256, 15, 1	1×1	1	1	295168
27	Convolution 2D	1, 15, 1	3×3	2	256	257
28	Add(conv2d_10[0][0])	256, 15, 1	-	-	-	0
29	Batch Normalization	256, 15, 1	-	-	-	4
30	Activation	256, 15, 1	-	-	-	0
31	Convolution 2D	512, 8, 1	3×3	2	512	1180160
32	Batch Normalization	512, 8, 1	-	-	-	4
33	Activation	512, 8, 1	-	-	-	0
34	Convolution 2D	512, 8, 1	3×3	1	512	2359808
35	Convolution 2D	512, 8, 1	1×1	1	1	1180160
36	Convolution 2D	1, 8, 1	3×3	2	512	513
37	Add(conv2d_14[0][0])	512, 8, 1	-	-	-	0
38	Activation	512, 8, 1	-	-	-	0
39	Flatten	4096	-	-	-	0
40	Dense	512	-	-	-	2097664
41	Activation	512	-	-	-	0
42	Dense	5	-	-	-	2565
43	Activation	5	-	-	-	0
	Trainable parameters			9,514,905		
	Non-trainable parameters			16		
	Total parameters					9,514,921

The last layer of the densely connected network exploits the softmax activation function to enable the normalization of the outputs into probabilities of the envisaged five classes. The softmax function is established on Luce's choice axiom [23] and its mathematical representation is shown in Eq. (5).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{5}$$

The complete description of the proposed Cardi-Net architecture is shown in Table 3. The input layer of the proposed model has a shape of (1025, 120, 1). The first residual block includes a two-dimensional convolution layer having 64 filters having a kernel size of 3×3, which follows a stride of 2 with 'valid' padding, batch normalization with output shape (1025, 120, 1) and activation function being ReLU resulting in an output shape of (64, 60, 1). The next 2D convolution layer takes input from the preceding 2D convolution layer and it has 64 filters having a kernel size of 3×3, which follows a stride of 1 with 'valid' padding, batch normalization with output shape (64, 60, 1) and activation function being ReLU resulting in an output shape of (64, 60, 1). The next 2D convolution layer takes the input from the original input data, as shown in Fig. 3. This prevents any loss of features in the preceding convolution layers. It has 1 filter having a kernel size of 1×1 which follows a stride of 1 with 'valid' padding and activation function being ReLU resulting in an output shape of (64, 60, 1). The residual 2D convolution layer takes input from the previous 2D convolution layer and it has 64 filters having a kernel size of 3×3, which follows a stride of 2 with 'valid' padding and activation function being ReLU resulting in an output shape of (1, 60, 1). The outputs of the preceding 2D convolution layer and the residual 2D convolution layer are concatenated to be fed as the input for the succeeding residual block with an output shape of (64, 60, 1). Similarly, the second, third and last residual blocks are as described in Fig. 3 and Table 3.

The fully connected neural network contains a flatten layer which takes its input from the output of the last residual block and gives an output of (4096) neurons and feds it to the next layer, i.e., the densely connected layer, which is followed by ReLU activation function and results in an output shape of (512). The densely connected output layer is followed by a softmax activation function and results in an output shape of (5) which signifies five classes, one out of which is normal while the rest of the four belongs to

Table 4
Various parameters, hardware and software specifications.

Attribute Name	Parameters Specification
Operating System	64-bit Ubuntu Linux workstation
Processors	2 x Intel Xenon Platinum
RAM	64 GB
Graphics Card	NVIDIA 16x Nvidia V100
GPU Memory	32 GB
Programming Language	Python, version 3.6.9
Development Environment	Jupyter Notebook, Keras, Tensorflow, librosa
Input Data	Power Spectrogram images PCG dataset
Input image dimension	1025×120×1
Batch Size	32
Loss Type	Categorical Cross-Entropy
Activation Function	ReLU and Softmax
No. of Epochs	500
Optimization function	Adadelta
Learning Rate	0.01

the four cardiac disorders, namely: Aortic Stenosis, Mitral Regurgitation, Mitral Stenosis, Mitral valve prolapse.

4. Experimental results

The Cardi-Net model is trained on the cardio dataset with 10-fold cross-validation with tuned hyperparameters. There is total of 1000 audio files in the original dataset distributed in five categories, with each category containing 200 files. Firstly, the model is trained with the original dataset of 1000 files. Then, the same model is trained with augmented dataset having total of 2000 files with each category containing 400 files. All the audio .wav format files are converted into power spectrogram images with the dimension of (1025×120×1). The trained Cardi-Net model can predict various cardiac diseases accurately. The training is performed on Ubuntu OS having NVIDIA DGX-II server (16 Tesla V100 32 GB). The rest of the parameters, software and hardware related information is mentioned in the Table 4.

There are five categories Normal, Aortic Stenosis, Mitral Regurgitation, Mitral Stenosis and Mitral valve prolapse in the PCG cardio dataset. The performance of the trained Cardi-Net model is analyzed using 10-fold cross-validation based on different performance parameters like accuracy, specificity, sensitivity (recall), precision and f-1 score. These performance matrices are obtained using False Positive (FP), True Positive (TP), False Negative (FN) and True Negative (TN), which can be computed using the obtained confusion matrices as shown in Table 5. It can be used for the raw as well for the augmented data Table 5. represents the confusion matrix for all 10-fold having multi-classification predictions and compared to the ground truth labels for both parts of the training (heart sound dataset and augmented data training). In CNNs, more training data provides usually higher accuracies, so the confusion matrix illustrates this aspect as greater accuracy with the larger dataset. TP is the correct prediction of cardiac disease by the Cardi-Net model.

Similarly, TN is the correct prediction that the PCG signals does not contain any cardiac disease pattern hence it is normal while FP is the wrongly predicted PCG signal as cardiac disease and FN is wrongly predicted PCG signal. High accuracy shows better model performance and FN predictions are highly sensitive for a model because if the person has cardiac disease, but the model wrongly predicts that cardiac signals do not have a disease pattern in their PCG signal. In such a case, it will convey a wrong prediction of contentment to the patients. Otherwise, further diagnosis might save their life.

An exhaustive analysis is performed using the performance matrix of the Cardi-Net trained model in order to make the model robust. The matrix components, on which the performance of the model is evaluated, are mentioned in Eqs. (10)–(17).

$$Accuracy\ of\ ResNet\ model\ (Acc) = \frac{Total\ correct\ predictions}{Total\ number\ of\ subjects}$$

$$= \frac{(TP\ +\ TN)}{TP + TN + FN + FP} \tag{10}$$

$$Average\ Accuracy\ = \frac{\sum_{n=1}^{10} Acc_n}{10} \tag{11}$$

$$Recall\ or\ Sensitivity\ = \frac{(TP)}{(TP\ +\ FN)} \tag{13}$$

$$Specificity = \frac{(TN)}{(TN\ +\ FP)} \tag{14}$$

$$f1\ Score\ = 2 \times \frac{(Precision\ \times\ Recall)}{(Precison +\ Recall)} \tag{15}$$

$$TPR = \frac{TP}{(TP + FN)} \tag{16}$$

$$FPR = \frac{FP}{(FP + TN)} \tag{17}$$

where,

Acc_n is the accuracy of n^{th} fold in 10-fold CV, *TPR* is the True Positive Rate and *FPR* is the False Positive Rate.

4.1. Cross-validation result analysis

For optimization, Adadelta function is used with the learning rate of 0.01. There are a total of 9,514,905 trainable parameters in the proposed model architecture. After converting the cardiac audio signals into power-scaled spectrograms, data shuffling is applied to obtain the final randomly distributed data among five classes. Model is trained with 100 epochs for each fold with standard batch size 32. The obtained results of average training and testing accuracies are 98.678% and 98.680%, respectively, for the original cardiac dataset and the obtained, average training and testing losses are 0.03712 and 0.08981, respectively. In Table 6, original cardiac data set results are shown.

Similarly, for the augmented dataset, each fold contains 1800 training samples and 200 testing samples. The obtained training and validation accuracies for augmented datasets are 99.883% and 98.879% and the losses are 0.0004 and 0.1846 for training and testing, respectively Table 7. shows the augmented dataset results.

The illustrative representation of accuracy, precision, sensitivity, specificity and F_1 score for each fold is shown in Fig. 4. for original dataset in which all the scores of accuracies, precision, sensitivity, specificity and F_1 score are varying between 87% and 100% for original cardiac dataset. Y-axis represents the obtained percentage scores while X-axis represents the number of folds. The pictorial representation of accuracy, precision, sensitivity, specificity and F_1 score for each fold is shown in Fig. 5. for the augmented dataset in which all the scores of accuracies, precision, sensitivity, specificity and F_1 score are varying between 94% and 100% for augmented cardiac dataset Y-axis represents the obtained percentage scores while X-axis represents the number of folds.

Fig. 6 shows the receiver operating characteristics (ROC) curve for each fold and area under the curve (AUC) for the original data, i.e. without augmentation. The proposed CNN architecture is built and trained on both original recorded and augmented data. Data augmentation increases the dataset size which is essential for a deep learning model and moreover augmented data contains PCG signal variations which makes the model more robust and efficient.

An exhaustive analysis is done, where the performance is measured using 10-fold cross validation. In each fold new set of data

Table 5
10-fold confusion matrix for the proposed model (raw and augmented dataset).

Truth Labels	Fold-1	N	21	0	1	0	0	Truth Labels	Fold-6	N	15	0	0	0	0
		AR	0	15	1	0	0			AR	0	16	0	0	7
		MR	0	0	18	0	0			MR	0	0	15	0	1
		MS	0	0	2	23	0			MS	0	0	0	24	0
		MVP	0	0	7	0	12			MVP	0	0	0	0	22
	Fold-2	N	21	0	0	0	0		Fold-7	N	16	0	0	0	0
		AR	1	23	0	0	0			AR	0	18	0	0	0
		MR	0	0	17	0	1			MR	0	0	21	1	0
		MS	0	0	0	20	1			MS	0	0	0	25	0
		MVP	0	0	0	0	16			MVP	0	0	1	0	18
	Fold-3	N	27	0	0	0	0		Fold-8	N	22	0	0	0	0
		AR	0	15	0	0	0			AR	0	25	0	0	0
		MR	0	0	24	0	0			MR	0	0	16	0	0
		MS	0	0	0	20	0			MS	0	0	0	15	0
		MVP	0	1	0	0	13			MVP	0	1	1	0	20
	Fold-4	N	22	0	0	0	0		Fold-9	N	20	0	0	0	0
		AR	0	23	0	0	0			AR	0	23	0	0	0
		MR	0	1	16	0	0			MR	0	0	25	0	0
		MS	0	0	0	18	0			MS	0	0	0	14	1
		MVP	0	0	0	0	20			MVP	0	1	0	0	16
	Fold-5	N	19	0	0	0	0		Fold-10	N	16	0	0	0	0
		AR	0	18	0	0	0			AR	0	15	0	0	0
		MR	0	0	21	0	1			MR	0	0	21	0	1
		MS	0	0	0	20	0			MS	0	0	0	17	0
		MVP	0	0	0	0	21			MVP	0	0	0	1	29
			N	AR	MR	MS	MVP				N	AR	MR	MS	MVP
			Predicted labels			Predicted labels									
Truth Labels	Fold-1	N	42	0	0	0	0	Truth Labels	Fold-6	N	36	0	0	0	0
		AR	0	31	1	0	0			AR	0	41	0	0	1
		MR	0	0	44	0	0			MR	0	0	39	0	0
		MS	0	0	2	39	0			MS	0	0	1	37	1
		MVP	0	0	1	0	40			MVP	0	0	0	0	44
	Fold-2	N	46	0	0	0	0		Fold-7	N	32	0	0	0	0
		AR	0	45	0	0	0			AR	0	51	0	0	0
		MR	0	0	38	0	0			MR	0	0	30	1	3
		MS	0	0	0	35	1			MS	0	0	0	32	0
		MVP	0	0	0	0	35			MVP	0	1	3	2	45
	Fold-3	N	39	0	0	0	0		Fold-8	N	47	0	0	0	0
		AR	0	34	0	0	0			AR	0	47	0	0	3
		MR	0	0	42	0	1			MR	0	0	33	0	2
		MS	0	0	0	34	0			MS	0	0	0	40	1
		MVP	0	0	1	1	48			MVP	0	0	1	1	25
	Fold-4	N	45	0	0	0	0		Fold-9	N	46	0	0	0	0
		AR	0	34	0	0	0			AR	0	30	0	0	0
		MR	0	0	41	1	1			MR	0	1	33	2	3
		MS	0	0	0	39	1			MS	0	1	0	46	0
		MVP	0	1	0	0	37			MVP	0	1	0	0	36
	Fold-5	N	37	0	0	0	0		Fold-10	N	30	0	0	0	0
		AR	0	37	1	0	0			AR	0	40	3	0	1
		MR	0	1	44	0	2			MR	0	0	38	0	0
		MS	0	0	0	37	2			MS	0	0	0	49	1
		MVP	0	1	0	0	38			MVP	0	2	1	0	35
			N	AR	MR	MS	MVP				N	AR	MR	MS	MVP
			Predicted labels			Predicted labels									

Table 6
10-fold CV results on cardiac dataset without augmentation.

	Train	Training	Training	Validation	Validation	Validation		Sensitivity			Total
Fold	Samples	Accuracy	Loss	Samples	Accuracy	Loss	Precision	or Recall	Specificity	f-1 score	Epochs
1	900	91.44	0.2065	100	95.60	0.3057	0.888725	0.924138	0.974012	0.888594	47
2	900	100.0	0.0081	100	98.80	0.0588	0.971032	0.968687	0.992493	0.968736	48
3	900	99.89	0.0116	100	99.60	0.0312	0.985714	0.9875	0.997701	0.986141	50
4	900	100.0	0.0021	100	99.60	0.0136	0.988235	0.991667	0.997619	0.989684	77
5	900	100.0	0.0029	100	99.60	0.0123	0.990909	0.990909	0.997468	0.990698	65
6	900	95.56	0.1095	100	96.80	0.1451	0.92663	0.946667	0.98098	0.926882	58
7	900	100.0	0.0015	100	99.20	0.0669	0.980383	0.983217	0.994997	0.981582	84
8	900	100.0	0.0031	100	99.20	0.0655	0.981818	0.980543	0.995	0.980494	73
9	900	99.89	0.0150	100	99.20	0.1064	0.974902	0.979902	0.995265	0.977083	40
10	900	100.0	0.0109	100	99.20	0.0926	0.984242	0.982222	0.994611	0.982968	55
Total	98.678	0.03712		98.680	0.08981	0.967259	0.973545	0.992015	0.967286		

Table 7
10-fold CV results on cardiac dataset after augmentation.

Fold	Train Samples	Training Accuracy	Training Loss	Validation Samples	Validation Accuracy	Validation Loss	Precision	Sensitivity or Recall	Specificity	f-1 score	Total Epochs
1	1800	100.0	0.0027	200	99.20	0.0509	0.979115854	0.983333333	0.995082096	0.980660609	61
2	1800	100.0	0.0003	200	99.80	0.0379	0.994444444	0.994444444	0.998787879	0.994366197	79
3	1800	100.0	0.0043	200	99.40	0.0602	0.987348837	0.985552919	0.996077108	0.98638968	58
4	1800	100.0	0.0026	200	99.20	0.0650	0.980434517	0.979029304	0.994992041	0.979547337	53
5	1800	99.94	0.0030	200	98.60	0.1085	0.966586269	0.966251526	0.991166986	0.965903303	70
6	1800	100.0	0.0017	200	99.40	0.0528	0.984981685	0.986304348	0.996288151	0.985351114	66
7	1800	100.0	0.0002	200	98.00	0.0987	0.952941176	0.948329171	0.987314844	0.950025688	79
8	1800	99.06	0.0212	200	98.40	0.1028	0.956878565	0.950529921	0.990044067	0.952654525	57
9	1800	100.0	0.0004	200	98.39	0.1846	0.959570045	0.958100233	0.990196581	0.956967419	67
10	1800	99.83	0.0058	200	98.40	0.1866	0.962028708	0.960617761	0.989931224	0.960692976	77
Total	99.883	0.00422		98.879	0.0948	0.97243301	0.971249296	0.992988098	0.971255885		

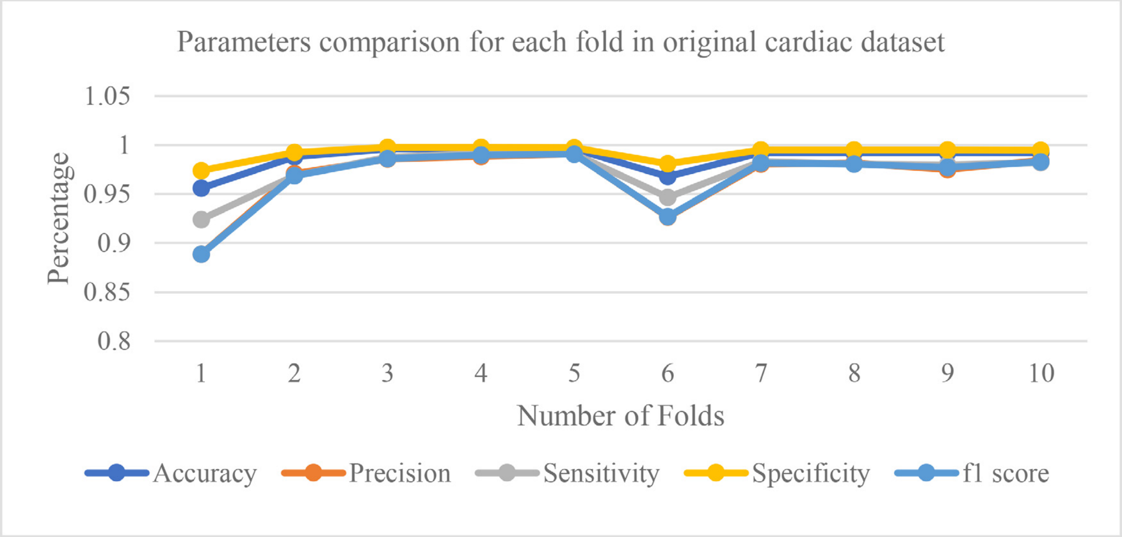

Fig. 4. Visual representation of Accuracy, Precision, Sensitivity, Specificity and F_1 score of original cardiac dataset.

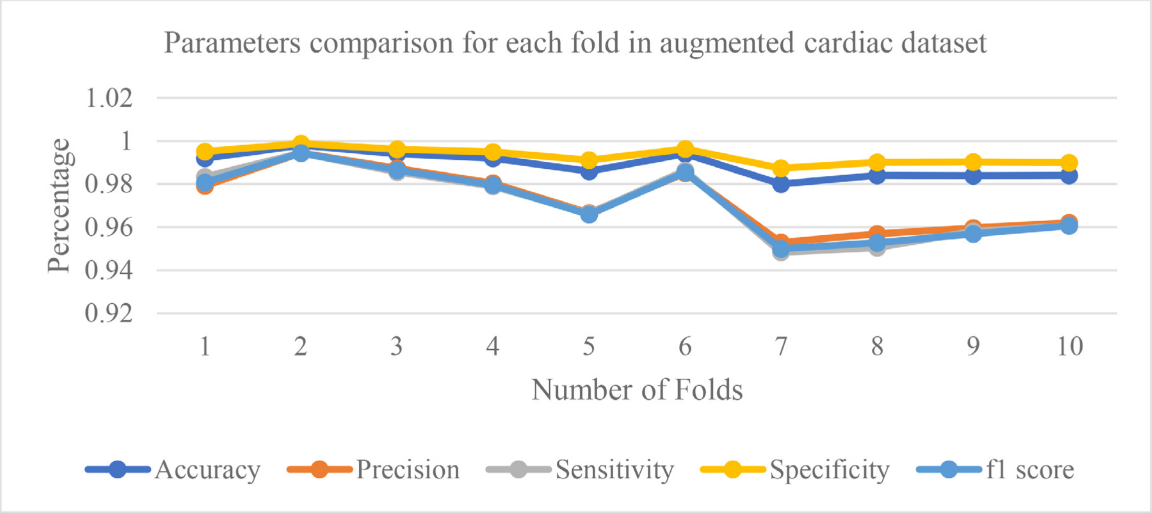

Fig. 5. Visual representation of Accuracy, Preci sion, Sen sitivity, Specificity and F_1 score of augmented cardiac dataset.

is used and receiver operating characteristics (ROC) curve is plotted for validation purpose Fig. 7. shows the ROC for each fold and area under the curve (AUC) for the augmented data. Average AUC value without augmentation is 0.98 and with augmentation is 0.99. The consistent ROC and AUC values shows the proposed model robustness and performance than the previous methods on the same dataset. The average history of model accuracy and loss over the epochs of training and testing of all folds for the power-scaled spectrogram dataset is shown in Fig. 8. In contrast, the average history for model accuracy and loss over the 500 epochs of training and testing of all folds for the power-scaled spectrogram dataset after data augmentation is shown in Fig. 9. The state-of-the-art and its comparison with the proposed technique is illustrated in Table 8. It shows the extracted features, classifiers used, database used and the results. A detailed comparison of the existing approaches for cardiac disorders detection

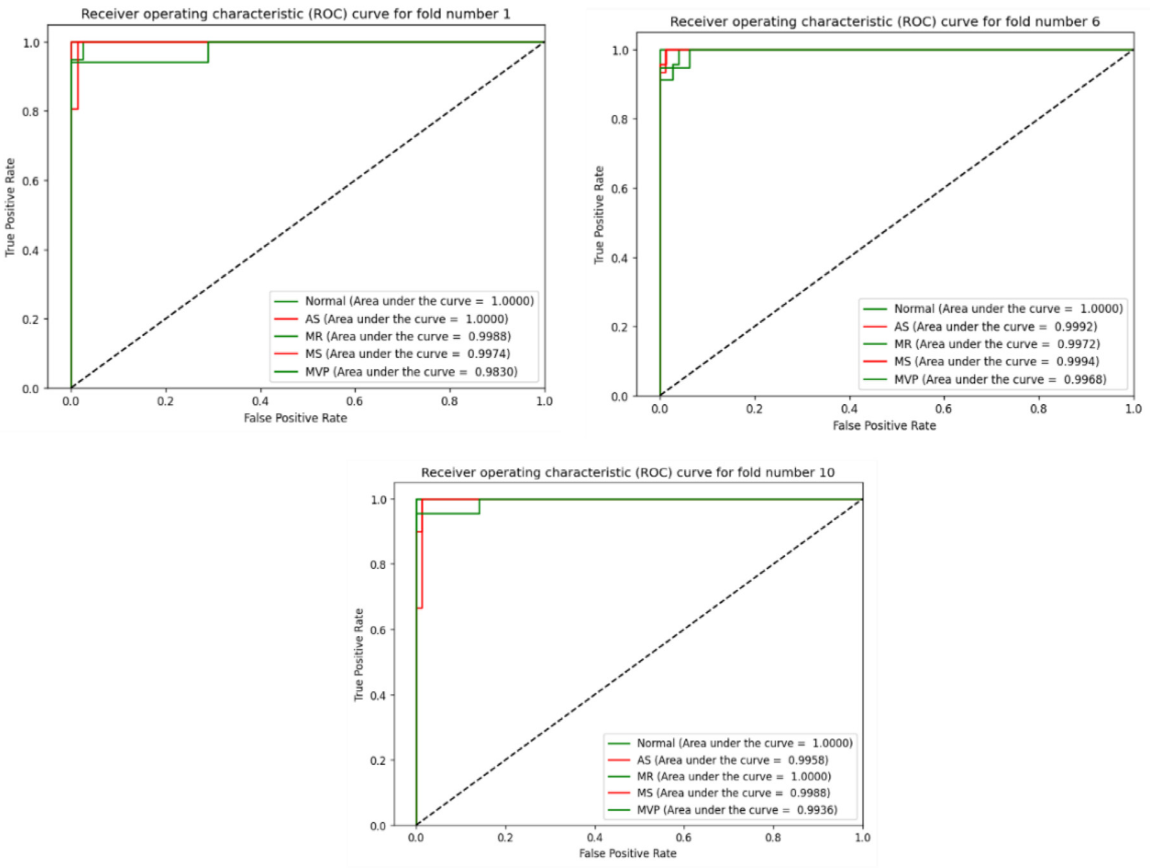

Fig. 6. ROC curve and the AUC for fold-1, 6 and 10 for the proposed architecture (without augmentation).

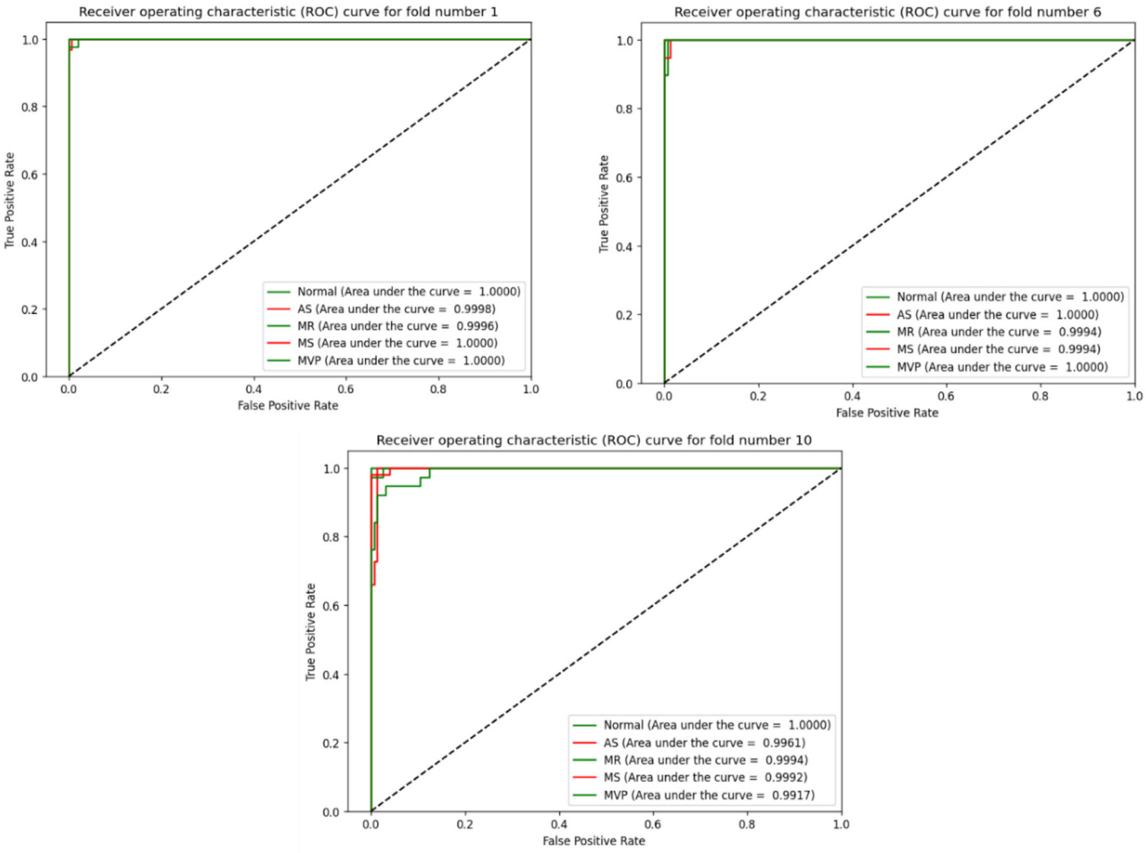

Fig. 7. ROC curve and the AUC for fold-1, 6 and 10 for the proposed architecture (after augmentation).

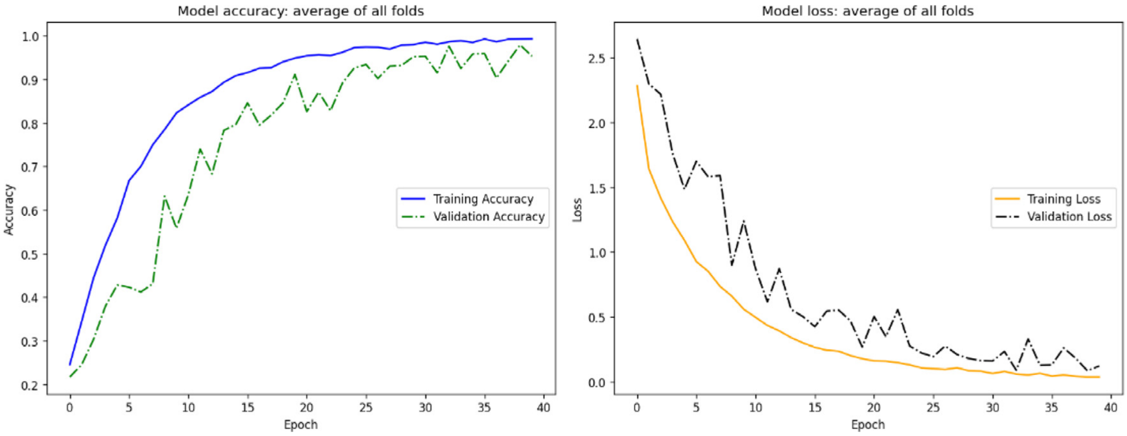

Fig. 8. Average 10-fold CV history. (a) average model accuracy for training and testing and (b) average model loss for training and testing (without augmentation).

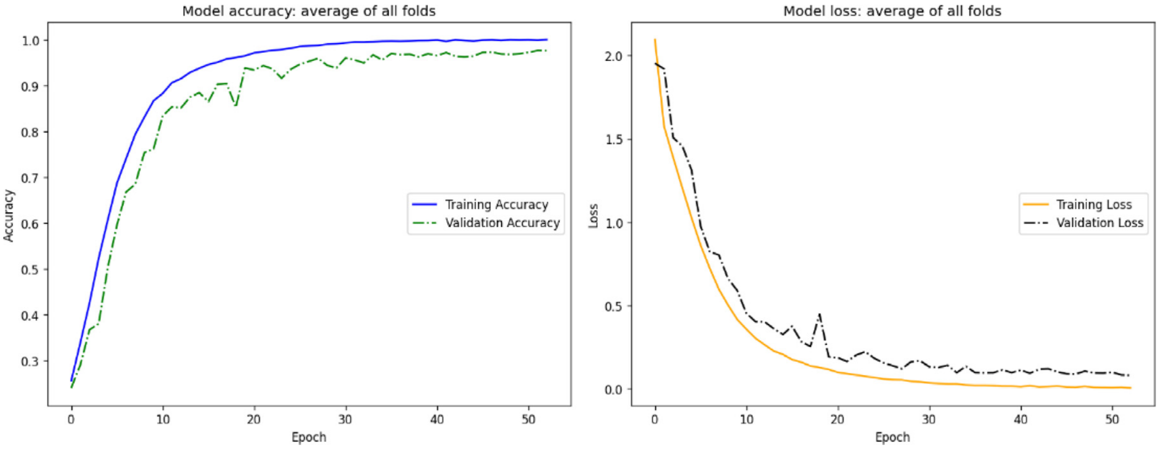

Fig. 9. Average 10-fold CV history. (a) average model accuracy for training and testing and (b) average model loss for training and testing (after augmentation).

Table 8
State-of-the-art of PCG signals classification.

Reference	Features Extracted	Classifier Used	Database	Results
[2]	Wavelet transform based features	incremental self-organizing map	14 heart sounds	95%
[3]	Spectral Features, Wavelet-based Features, Cepstral Features	kNN, MLP, SVM	murmur	95.2%
[4]	MFCC	HMM	10 types of heart disease with 325 total samples (CVDs)	95.08%
[5]	2D mfcc heatmap features	Deep CNN	2 types 3240 PhysioNet	84.8%
[6]	Raw 1D PCG signals	1D Deep Gated RNN	2 types 3153 PhysioNet	89%
[8]	spectral amplitude, wavelet entropy	Combined spectral amplitude and wavelet entropy classifier	PhysioNet/Computing in Cardiology Challenge 2016 (PhysioNet 2016) (heart valve disease and coronary artery disease)	70%
[9]	Murmur likelihood as a temporal feature	HMM, HMM+SVM	i.) normal/abnormal ii.) nine categories murmer	85.6%
[10]	Unsegmented features, Using CNN kernel	CNN, SVM, kNN, Ensemble, LDA	2 types 3240 PhysioNet	90%
[11]	Wavelet, entropy, fractal	twin support vector machine	PhysioNet/CinC Challenge 2016 heart sound database (heart valve disease and coronary artery disease)	90.4%
[13]	spectrograms and Melfrequency cepstrum coefficients	LR, SVM, RF, CNN	Two datasets	88.5%
[25]	MFCC and DWT	SVM, DNN and Centroid displacement-based k nearest neighbor	OPEN Heart Sound Database (valvular disease)	89.30%
[26]	Machine learning	SVM, kNN	OPEN Heart Sound Database (valvular disease)	96.50%
Proposed method	Features are extracted by Cardi-Net	Deep CNN	OPEN Heart Sound Database (valvular disease)	98.879%

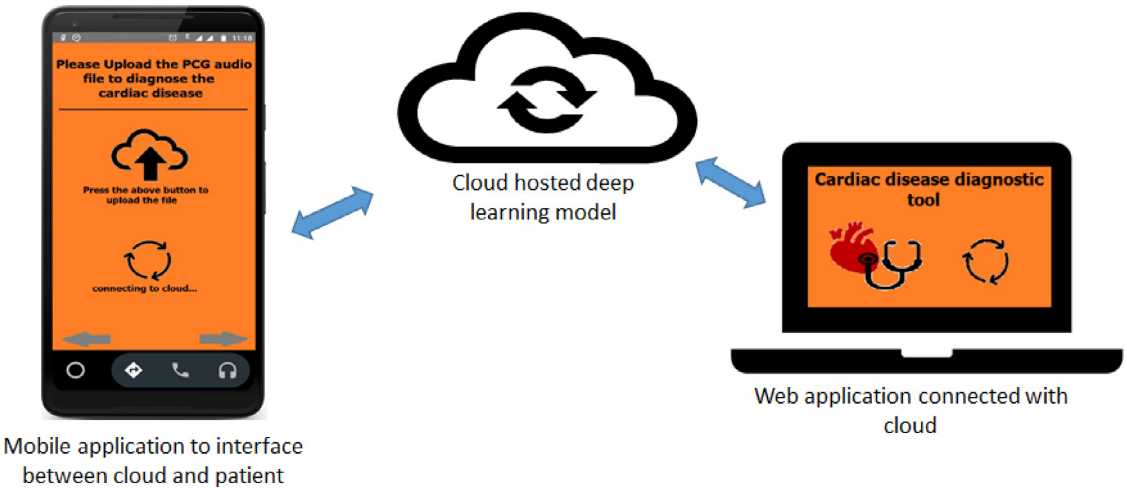
Please Upload the PCG audio file to diagnose the cardiac disease
Press the above button to upload the file
connecting to cloud...
Mobile application to interface between cloud and patient
Cloud hosted deep learning model
Cardiac disease diagnostic tool
Web application connected with cloud

Fig. 10. Cloud deployed model with mobile and desktop clients.

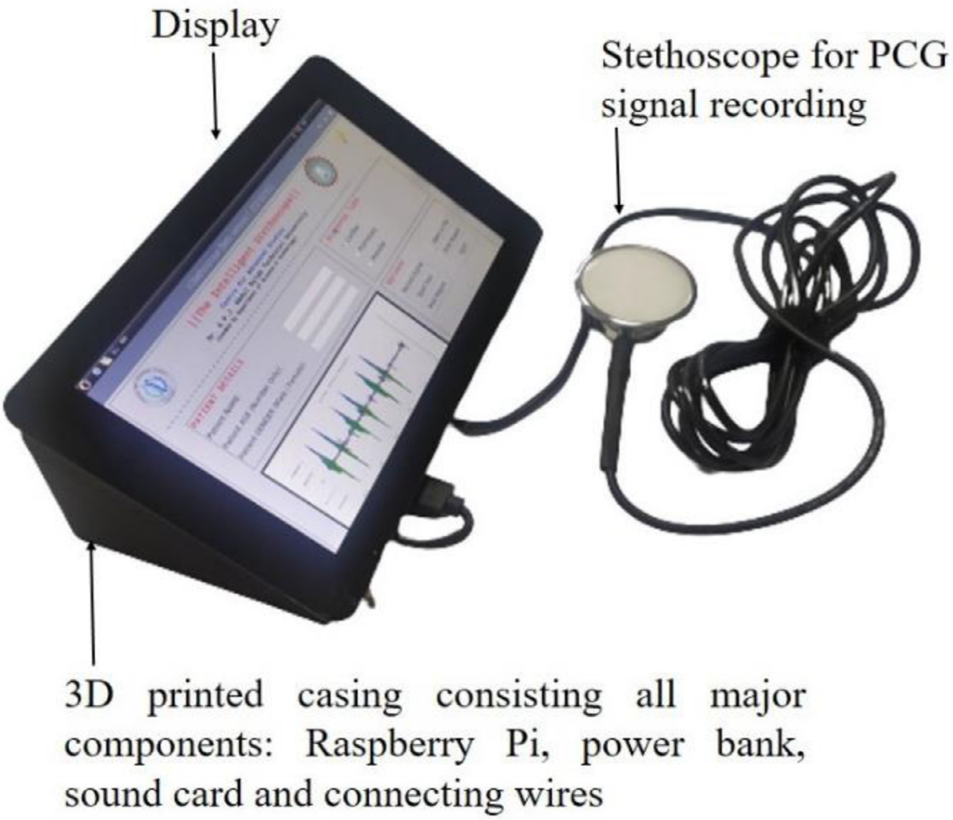
Display
Stethoscope for PCG signal recording
3D printed casing consisting all major components: Raspberry Pi, power bank, sound card and connecting wires

Fig. 11. Laboratory picture of the prototype of the proposed model.

by exploiting various phonocardiogram datasets has been made where the proposed method surpasses them with respect to the accuracy, thereby very smoothly dealing with the misclassification issues.

5. Possible integration of the trained model on cloud and ubiquitous access through the internet

Using the modern web and mobile application technologies, the trained model can be deployed in cloud servers. Mobile and desktop applications can be developed in order to access trained model facilities even in remote areas with limited bandwidth. This type of application can be beneficial for the public who cannot avail of such medical facilities readily. Cardiac sound can be recorded using the digital stethoscope, and the recorded sound can be uploaded to the cloud where the trained deep learning model will predict the cardiac disease. The appropriate report can also be generated within the application.

Fig. 10 is a possible demonstration of deployed deep learning model in the cloud and its accessibility with different applications. The application is lightweight and fast in speed because the recorded cardiac sound occupies less storage than other solutions. Uploading on servers would not take much time and the model can quickly predict possible diseases. The real-time speed is analyzed on low bandwidth and various time parameters as shown in Fig. 10.

Table 9
Time factors for different categories of audio signals of cardiac disease dataset.

Audio file description	Category	Size of the file	Avg. Uploading time (at 150 kbps) in seconds	Power Spectrogram conversion time (in Seconds)	Prediction time on CPU (in Seconds)	Total Time (in Seconds)
Normal	Healthy	33 KB	1.76	0.103088	0.282614	2.145702
Aortic Stenosis	Patient	41 KB	2.18	0.113416	0.203807	2.497223
Mitral Regurgitation	Patient					
33 KB	1.76	0.102975	0.117216	1.980191		
Mitral Stenosis	Patient					
47 KB	2.50	0.121328	0.220056	2.841384		
Mitral valve prolapses	Patient					
44 KB	2.34	0.112257	0.076229	2.528486		

A stand-alone device is also possible in which the model can be integrated with a digital signal processor (DSP) board using a sound card. Cardiac sound can be recorded with an electronic stethoscope, which uses a microphone to record the heart sounds digitally. Other components like a power bank, switch, USB cables, etc. are useful to make the device a complete toolkit. An electronic display is attached to it to display the generated reports based on the model prediction. The diagnosis of cardiac diseases using this toolkit is internet independent Fig. 11. shows a block diagram of possible integration of DSP, electronic display, sound card and power bank. This device is capable of recording, processing, extracting relevant features from the recorded cardiac audio files and classifying them.

In Table 9, information on audio file description, category of audio file, size of each audio file, average uploading time at 150 kbps speed, power spectrogram conversion (pre-processing) time from a recorded audio cardiac file, and prediction time has been included. The time consumed while loading the model from internal storage to RAM is 5.883776 s. These predictions can be made by selecting one of the best models from all 10-fold saved models.

6. Conclusion

The criticality of cardiovascular diseases is a major concern that needs to be diagnosed and cured at the earliest for better living. This has been resolved by exploiting a deep neural network based on cnn architecture to automatically classify four major types of cardiac disorders. Data augmentation is implemented to make the model adaptable to noise. A set of raw and augmented PCG audio samples of certain time duration are converted to the frequency domain as PSD to form power spectrograms which produced highly discriminatory features for automatic classification using the deep learning model. The proposed 2D Pigment model has attained an accuracy of 98.879% with a loss of 0.0948, which clearly demonstrates that it surpasses the prevailing methods. The presented approach is highly robust and is relatively fast due to low PSD conversion time. The possible integration of the trained model on a device makes it suitable for real-time implementation. In future, emphasis will be laid on the further investigation, categorization and diagnosis of various heart-related disorders and clinical device implementation.

Declaration of Competing Interest

None.

Acknowledgements

This work was supported by the Department of Science and Technology, Ministry of Science and Technology (Grant number DST/IDP/MED/36/2016).

References

[1] World Health Organization: Cardiovascular diseases, May 2017, accessed in January 2021 https://www.who.int/health-topics/cardiovascular-diseases.

[2] Z. Dokur, T. Ölmez, Heart sound classification using wavelet transform and incremental self-organising map, Digit. Signal Process. 18 (6) (2008) 951–959.

[3] J. Vepa, Classification of heart murmurs using cepstral features and support vector machines, in: Proceeding of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, 2009, pp. 2539–2542, doi:10.1109/IEMBS.2009.5334810.

[4] H. Wu, S. Kim, K. Bae, Hidden Markov model with heart sound signals for identification of heart diseases, in: Proceedings of 20th International Congress on Acoustics (ICA), Sydney, Australia, 2010, pp. 23–27.

[5] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, K. Sricharan, Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients, in: Proceeding of the Computing in Cardiology Conference (CinC), IEEE, 2016, pp. 813–816.

[6] C. Thomae, A. Dominik, Using deep gated RNN with a convolutional front end for end-to-end classification of heart sound, in: Proceeding of the Computing in Cardiology Conference (CinC), IEEE, 2016, pp. 625–628.

[7] V.N. Varghees, K.I. Ramachandran, Effective heart sound segmentation and murmur classification using empirical wavelet transform and instantaneous phase for electronic stethoscope, IEEE Sens. J. 17 (12) (2017) 3861–3872.

[8] P. Langley, A. Murray, Heart sound classification from unsegmented phonocardiograms, Physiol. Meas. 38 (8) (2017) 1658.

[9] C. Kwak, O.W. Kwon, Cardiac disorder classification by heart sound signals using murmur likelihood and hidden Markov model state likelihood, IET Signal Process. 6 (4) (2012) 326–334.

[10] S.A. Singh, S. Majumder, M. Mishra, Classification of short unsegmented heart sound based on deep learning, in: Proceeding of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, 2019, pp. 1–6.

[11] J. Li, L. Ke, Q. Du, Classification of heart sounds based on the wavelet fractal and twin support vector machine, Entropy 21 (5) (2019) 472.

[12] Y. Deng, P.J. Bentley, A robust heart sound segmentation and classification algorithm using wavelet decomposition and spectrogram, in: Proceeding of the Workshop Classifying Heart Sounds, 2012, pp. 1–6.

[13] T. Nilanon, J. Yao, J. Hao, S. Purushotham, Y. Liu, Normal/abnormal heart sound recordings classification using convolutional neural network, in: Proceeding of the Computing in Cardiology Conference (CinC), IEEE, 2016, pp. 585–588.

[14] W. Zhang, J. Han, S. Deng, Heart sound classification based on scaled spectrogram and partial least squares regression, Biomed. Signal Process. Control 32 (2017) 20–28.

[15] W. Zhang, J. Han, S. Deng, Heart sound classification based on scaled spectrogram and tensor decomposition, Expert Syst. Appl. 84 (2017) 220–231.

[16] F. Demir, A. Şengür, V. Bajaj, K. Polat, Towards the classification of heart sounds based on convolutional deep neural network, Health Inf. Sci. Syst. 7 (1) (2019) 1–9.

[17] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, O. Nieto, Librosa: audio and music signal analysis in python, in: Proceedings of the 14th Python in Science Conference, 8, 2015, pp. 18–25.

[18] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[19] CholletFrançois, keras, 2015.

[20] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, in: Proceedings of the International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 31, 2017 1.

[22] S.B. Jadhav, S.B. Patil, Grading of soybean leaf disease based on segmented image using k-means clustering, Int. J. Adv. Res. Electr. Commun. Eng. 4 (6) (2015) 1816–1822.

[23] Y. Jr, J. I, The relationship between Luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution, J. Math. Psychol. 15 (2) (1977) 109–144.

[24] A. Yadav, A. Singh, Malay Kishore Dutta, M.T. Carlos, Machine learning-based classification of cardiac diseases from PCG recorded heart sounds, Neural Computing and Applications 32 (24) (2020) 17843–17856.

[25] G.Y. Son, S. Kwon, Classification of heart sound signal using multiple features, Appl. Sci. 8 (12) (2018) 2344.

[26] P. Upretee, M.E. Yüksel, Accurate classification of heart sounds for disease diagnosis by using spectral analysis and deep learning methods, in: Data Analytics in Biomedical Engineering and Healthcare, Academic Press, 2021, pp. 215–232.

[27] M. Cerna, A.F. Harvey, Application note 041 – the fundamentals of FFT-based signal analysis and measurement, Natl. Instrum. (2000) July https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf .