mathematics

MDPI

Article

CNN and Bidirectional GRU-Based Heartbeat Sound Classification Architecture for Elderly People

Harshwardhan Yadav [1], Param Shah [1], Neel Gandhi [2], Tarjni Vyas [1], Anuja Nair [1], Shivani Desai [1], Lata Gohil [1], Sudeep Tanwar [1,*], Ravi Sharma [3], Verdes Marina [4,*] and Maria Simona Raboaca [5,6]

[1] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India
[2] Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA
[3] Centre for Inter-Disciplinary Research and Innovation, University of Petroleum and Energy Studies, Dehradun 248001, Uttarakhand, India
[4] Faculty of Civil Engineering and Building Services, Department of Building Services, Technical University of Gheorghe Asachi, 700050 Iasi, Romania
[5] Doctoral School, University Politehnica of Bucharest, Splaiul Independentei Street No. 313, 060042 Bucharest, Romania
[6] National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Vâlcea, Uzinei Street, No. 4, P.O. Box 7, 240050 Râmnicu Vâlcea, Romania
* Correspondence: sudeep.tanwar@nirmauni.ac.in (S.T.); marina.verdes@academic.tuiasi.ro (V.M.)

Abstract: Cardiovascular diseases (CVDs) are a significant cause of death worldwide. CVDs can be prevented by diagnosing heartbeat sounds and other conventional techniques early to reduce the harmful effects caused by CVDs. However, it is still challenging to segment, extract features, and predict heartbeat sounds in elderly people. The inception of deep learning (DL) algorithms has helped detect various types of heartbeat sounds at an early stage. Motivated by this, we proposed an intelligent architecture categorizing heartbeat into normal and murmurs for elderly people. We have used a standard heartbeat dataset with heartbeat class labels, i.e., normal and murmur. Furthermore, it is augmented and preprocessed by normalization and standardization to significantly reduce computational power and time. The proposed convolutional neural network and bi-directional gated recurrent unit (CNN + BiGRU) attention-based architecture for the classification of heartbeat sound achieves an accuracy of 90% compared to the baseline approaches. Hence, the proposed novel CNN + BiGRU attention-based architecture is superior to other DL models for heartbeat sound classification.

Keywords: cardiovascular diseases; heart sound; deep learning; classification; GRU; CNN; attention mechanism

MSC: 62H35

check for updates

Citation: Yadav, H.; Shah, P.; Gandhi, N.; Vyas, T.; Nair, A.; Desai, S.; Gohil, L.; Tanwar, S.; Sharma, R.; Marina, V.; Raboaca, M.S. CNN and Bidirectional GRU-Based Heartbeat Sound Classification Architecture for Elderly People. *Mathematics* **2023**, *11*, 1365. https://doi.org/10.3390/math11061365

Academic Editors: Xujuan Zhou, Lemai Nguyen and Guohun Zhu

Received: 3 February 2023
Revised: 7 March 2023
Accepted: 8 March 2023
Published: 10 March 2023

1. Introduction

Cardiovascular diseases (CVDs), as per the World Health Organisation (WHO), are the number one cause of death worldwide. It causes the death of about 17.9 million lives worldwide every year [1,2]. Moreover, 85% of CVDs are caused by acute myocardial infarction. CVDs are conditions concerning the blood and heart vessels, classified as diseases such as rheumatic heart, cerebra-vascular, coronary heart, etc. People suffering from hyperlipidemia, diabetes, hypertension, etc., are more prone to a higher risk of CVDs [3,4]. With age, gradually, these diseases become a part of our life. CVDs in elderly people may have more harmful effects than in young people as the recovery ratio is smaller in elderly people [5]. Hence, the early detection of indications of cardiac abnormality [6] is an essential step for patient care. In clinical practices, several tools are equipped to diagnose CVDs. Auscultation is a basic diagnostic method in which doctors listen to the heartbeat sound from a patient's chest to make a diagnosis [7,8]. Auscultation can be achieved using a medical instrument known as a "stethoscope", invented by Laennec in 1816. It is widely utilized in the medical field to hear heart sounds to diagnose CVDs [9]. However, by only using a stethoscope, CVDs and heart failures are challenging to diagnose, especially by non-clinical and inexperienced people. Despite accurate auscultation, it requires years of experience and long-term practice to diagnose CVDs, which is challenging to acquire [10]. Even clinical practitioners find it challenging to detect and diagnose heart failure and coronary heart disease at an earlier stage, and it is especially critically in elderly people [11].

The heart is a significant organ that pumps blood throughout the whole body, causing mechanical movements when cardiac valves close per heartbeat, generating vibrations in the myocardial wall, which later is converted to sounds comprehended as heartbeat sounds of phonocardiogram (PCG) [12]. A cardiac cycle has four elements, namely "S1", which happens at the beginning of the systole phase. The second is "S2", which characterizes the systole phase's end and marks the diastole phase's beginning. Thirdly, "S3" corresponds to the end of the fast filling of the ventricle. Lastly, "S4" means the active filling phase of the ventricle [13,14]. The sound of a normal heartbeat has an evident "lub dub, dub lub" pattern. Here, the time taken from dub to lub is more than the time evolved between lub to dub, with a rate of 60–100 beats per minute. In contrast, the sound of a murmur heartbeat is noisy with roaring, whooshing, or rumbling patterns between lub to dub or dub to lub, indicating signs of CVDs. In addition to the above-mentioned heartbeat sounds, two other sound categories, including extra-systole and extra-heart sounds, show some trouble but do not necessarily confirm indicators of CVDs. Researchers use these sound waves to detect/identify the problem associated with the heart. In elderly people, even with the help of a stethoscope, listening to these heart sounds and making an accurate diagnosis is complicated and inefficient. Hence, with the aid of artificial intelligence (AI), finding patterns and predicting heart-related disorders has become easier, thereby enabling better treatment to the patient. Heart sounds can be easily collected using various wearable devices and could benefit people with CVDs.

Many researchers have proposed various solutions to identify CVDs in elderly people; for example, the authors of Ref. [15] used three machine learning (ML)-based algorithms to classify CVDs. They achieved results such as gradient boosting (GB) (87.5% accuracy), random forest (RF) (81.25% accuracy), and support vector machine (SVM) (75% accuracy). While ML-based algorithms can be effective, they have a drawback in requiring manual feature extraction, which can be time-consuming and labor-intensive. With the advent of deep learning (DL)-based algorithms, many practitioners have shifted away from traditional ML methods in favor of these more powerful techniques, which can learn features automatically from raw data [16,17]. Ren et al. [18] attempted to give a deep attention-based model for heart sound classification; however, it used a limited dataset of 845 clips, including training, testing, and validation testing. Mukherjee et al. [19] used different pre-trained models such as ResNet152V2 and MobileNetV2 using transfer learning (TL), but the testing of time and pitch shift in audio after data augmentation was missing. Kui et al. [20] makes a classification model containing convolutional neural network (CNN) and Mel-frequency spectral coefficients (MFCC) features to detect CVDs. Still, it has a drawback of bad performance on the test set.

To overcome the aforesaid shortcomings, this article presents a CNN and bi-directional gated recurrent unit (BiGRU)-based architecture to detect normal efficiently and murmur heartbeats [21]. DL classifiers have proved to be promising in differentiating normal and abnormal heart sounds; we have considered heartbeat sounds from PCG and electrocardiogram (ECG) recordings [22] for training the proposed DL model. The architecture is evaluated using different evaluation metrics, such as statistical measures such as accuracy, precision, recall, etc., validation accuracy with and without data augmentation, and validation accuracy with different optimizers.

1.1. Research Contributions

The following are the major research contributions of the paper:

- To propose a novel attention-based CNN + BiGRU architecture for abnormal heartbeat audio classification;
- To employ Mel-frequency cepstral coefficient (MFCC) for feature extraction, data compression, and downsampling as a pre-processing step to train the AI models efficiently;
- To evaluate the proposed architecture using different evaluation metrics, like statistical measures such as accuracy, precision, recall, etc., validation accuracy with and without data augmentation, and validation accuracy with different optimizers.

1.2. Organization

The rest of the paper is systematized as follows. Section 2 describes various state-of-the-art approaches to heartbeat sound classification and its comparative study. Section 3 depicts problem formulation for the heart sound classification. Section 4 describes the proposed architecture. Section 5 provides results and analysis. Finally, Section 6 provides the conclusion to the work done.

2. Related Work

Several researchers have worked to solve the problem of heartbeat sound classification by applying solutions for every stage of heartbeat sound classification. In Ref. [23], Xiang et al. proposed an architecture based on various models. They used methods such as ML and TL along with feature extraction methods such as logarithmic power spectrogram, log-mel spectrogram, waveform diagram, and the envelope. Keikhosrokiani et al. [24] proposed a swarm intelligence-based model called artificial bee colony–hybrid adaptive neuro-fuzzy inference system (ABC-ANFIS), but it was computationally expensive. In Ref. [25], Ballas et al. proposed a self-supervised CNN based model and used two different sets of data augmentations, which included cut-off filters, rewinding the signal, and inverting. They achieved an accuracy of 78.6%, which is very low. Ren et al. [26] proposed a model based on CNN + attention and LSTM/GRU + attention and was validated using explainable artificial intelligence (XAI) [27]. However, the work needed to be improved in exploring bidirectional models. Moreover, the problems of overfitting and data augmentation still pertain. Tariq et al. [28] proposed an approach based on the feature-based fusion of CNNs, which achieved an accuracy of 97% but had clear class imbalance and overfitting problems.

Table 1. Comparative analysis of state-of-the-art approaches on heartbeat sound classification.

Related Works	Year	Key Contributions	Technology Used	Merits	Demerits
Proposed Architecture	2023	To propose a model to perform heart sound classification on the CirCor DigiScope dataset	CNN+BiGRU and attention	90% accuracy on the testing dataset and performed data augmentation as well	–
[23]	2023	Executed heart sound classification using two-dimensional features	Different ML, DL, and TL models	Used a wide range of models and feature extraction techniques to get better results	MFCC could have been used to further improve the results
[24]	2023	Heartbeat sound classification using a swarm intelligence-based algorithm	ABC-ANFIS	Used a novel approach of artificial bee colony along with ANFIS	The ABC-ANFIS method is quite complex
[29]	2022	A lightweight and robust approach for the detection of automatic heart murmurs using PCG recordings	Lightweight CNN	Uses two data augmentation techniques with low training and inference time	Accuracy of 75.1% is quite low compared to other models with similar architectures

Table 1. *Cont.*

Related Works	Year	Key Contributions	Technology Used	Merits	Demerits
[25]	2022	To detect a murmur in heartbeat sounds using a self-supervised approach	Self-supervised CNN backbone	Self-supervised model employed along with different data augmentation techniques	The accuracy of 73.7% is quite low and can be improved significantly
[26]	2022	To use XAI in heartbeat sounds with deep attention-based neural networks	CNN + attention and LSTM/GRU + attention	Used attention-based mechanism and validated results using XAI	Data augmentation does not improve the model results and does not tackle overfitting
[28]	2022	To classify lung and heartbeat sounds using feature-based fusion models	Fusion of CNNs	Applied learning in the form of fusion models and achieved an accuracy of 97% in six classes with data augmentation	Does not address overfitting concerns with a very small dataset for six classes and the degree of class imbalance is not explained
[30]	2021	To propose a model identifying abnormalities in a human heart using heart sound analysis	Artificial neural network (ANN) and linear discriminant analysis (LDA)	Achieved 90%, 83.33%, and 93.33% accuracy in the time, frequency, and time-frequency domain, respectively	Integration with DL algorithms might result in more accuracy
[20]	2021	To propose a model for the classification of heartbeat sound using mel-frequency spectral coefficients (MFSC)	CNN and hidden Markov model (HMM)	86.25% accuracy for multi-classification achieved	Poor performance of the test set
[31]	2021	To use MFSC and deep residual learning for reducing the cost and time of hearbeat sound classification	One-dimensional local binary pattern (1D-LBP) and local ternary pattern (1D-LTP), and 1D-CNN	Accuracy of 91.78% and 91.66% achieved with PhysioNet and PASCAL dataset	Use of SVM on these datasets works better
[19]	2021	To propose a model that converts heart sound to visual scale spectrograms	CNN, ResNet152V2, MobileNetV2	To extract features and categorize heart sounds using a TL approach	The time and pitch shift in audio was not tested
[18]	2021	To propose a model heart sound classification in HSS corpus using a Hamming window	CNN, GRU-recurrent neural network (RNN), and LSTM-RNN	Average recall of 51.2% achieved	Data size limitation
[32]	2020	To propose a 1D-CNN architecture for heart sound classification	CNN	Used stacked transition and clique blocks for promising classification performance, with lower consumption of parameters, and discriminative features extracted	Environment noises are not considered
[33]	2020	To propose a model for heartbeat sound classification using PCG signals to identify irregularities and achieving good cardiac diagnosis	CNN	Discrete cosine transform (DCT) achieved classification accuracy	A large number of samples are not taken for experiments

Initially, heartbeat segmentation [34] is done using probabilistic-based methods [35] and amplitude threshold-based methods [36,37]. Further, feature extraction is done by time and frequency [38,39]. Lastly, classification models such as SVM [40,41] and CNN [42] are used to predict normal and abnormal heartbeat sounds. Gomes et al. [43] used J48 and multilayer perceptron (MLP) algorithms to classify heart sounds for the PASCAL challenge into normal, murmur, extra sound, and artifact signals. Sound clips were first preprocessed to identify abnormal heartbeat sounds using the decimate function and band pass filters to remove noise. The average Shannon energy was used to determine the minima and maxima points using the heartbeat sound's peaks. The segmentation process was then performed on the heartbeat sound, although the ECG reference was not used. Raza et al. [44] proposed a framework for heartbeat sound classification based on long short-term memory (LSTM). They used the PASCAL dataset for classification. Moreover, data preprocessing was performed using bandpass filters, and down-sampling was done to acquire meaningful features. However, the study only considered ECG signals, making it unfit for non-clinical use. In the work done by Zheng et al. [45], various features such as the energy fraction of heart murmur, the entropy of first and second heartbeat sounds, max energy fraction of heartbeat sound, and frequency sub-band were extracted to detect abnormalities. Features are normalized and decomposed to wavelet packets. SVM was used to classify normal and murmur heartbeat sounds.

Shi-Wen Deng et al. [41] used SVM to classify normal, murmur, and the extra-systole heartbeat sound. Their solution was developed using auto-correlation features avoiding segmentation combined with discrete wavelet decomposition to feed into the machine learning (ML) model. They utilized a tensor decomposition and scaled spectrogram to extract discriminative features for the classification of heartbeat sound. Later, they used SVM for classification and effectively obtained a precision of 0.74 (74%). Yaseen et al. [46] used PCG signals for heartbeat sound classification. However, integration with DL algorithms could have been beneficial here. Feature extraction techniques were used to extract MFCC and discrete wavelets transform (DWT) for heartbeat sound classification using various ML models such as k-nearest neighbor (KNN), deep neural network (DNN), and SVM. Research has shown CNN to be significant in dealing with large-scale audio data [47]. Kele Xu et al. [48] proposed an ensemble-learning method based on CNN for classification. However, the accuracy could be increased if LSTM or GRU would be used with CNN. To overcome the issues discussed in the state-of-the-art approaches for heartbeat sound classification, we have proposed a DL-based supervised training approach that is quite efficient and straightforward to understand, employing CNN+BiGRU with an attention mechanism, which is superior to the self-supervised training approach. Table 1 depicts the comparative study of state-of-the-art approaches related to heartbeat sound classification.

3. Problem Formulation

In this section, we formulate the problem statement and identify the objective function for the identified problem. The proposed architecture comprises elderly people such as $\{A_1, A_2, \ldots, A_i, \ldots, A_n\} \in A$ where A is $1 \leq i \leq n$; and each has a wearable sensor [49] from a list of sensors such as $\{S_1, S_2, \ldots, S_j, \ldots, S_q\} \in S$ where S is $1 \leq j \leq q$. The sensor can read the heartbeat sound (μ) from a patient and pass it to the DL-based model ($\mathbb{M}$) for prediction (ρ), which helps in determining whether the patient has an abnormality present with the heart sound or not. The process is depicted below, considering the sensor reading as a function of the patient and model prediction as a function dependent on the reading provided by a sensor.

$$\forall\, n\; \exists\, q\; S_q(A_n) \xrightarrow{\text{provide}} \mu \tag{1}$$

$$\mathbb{M}(\mu) \xrightarrow{\text{predict}} \rho \tag{2}$$

If $\rho = 1$, then the wearable sensor needs to notify the user or the caretaker that it suspects something unusual with the heart pumping of the user. Following this, the user should get his heart condition checked by the doctor. This approach can significantly improve the chances of survival of an elderly person in case of an abnormal heart condition. However, for this to be achievable, we need to optimize both of the goals of maximizing the model's prediction accuracy and minimizing the model's loss during training. The model's accuracy employed under medical field applications should be high with no mistakes in prediction; otherwise, the user would be discouraged from relying on such sensor devices where a wrong prediction can lead to severe life-threatening consequences. Moreover, the loss of the model should be low, indicating that the model is making less errors when giving the correct probability for a class. Alternatively, a lower error means higher confidence in the prediction by the model, which is desirable as the model is confident in its diagnosis and is not just giving vague predictions. Thus, the objective of the model training, dealing with both the probability of success ($\mathbb{P}$) and error ($\mathbb{E}$) as a function of samples in validation data (τ) obtained by model prediction, can be given as follows.

$$\max_{\tau}(\sum \mathbb{P}(\eta(\tau))) + \min_{\tau}(\sum \mathbb{E}(\eta(\tau))) \tag{3}$$

where $\max_{\tau}(\sum \mathbb{P}(\eta(\tau)))$ denotes maximizing the probability of success of the model and $\min_{\tau}(\sum \mathbb{E}(\eta(\tau)))$ indicates minimizing the error of the model for validation samples in τ. These both can be combined and written as an objective function ($\mathbb{O}$) to maximize the ratio of ($\mathbb{P}$) and ($\mathbb{E}$), and is depicted as follows:

$$\mathbb{O} = \max_{\tau}(\sum \frac{\mathbb{P}(\eta(\tau))}{\mathbb{E}(\eta(\tau))}) \tag{4}$$

By maximizing $\mathbb{O}$ in Equation (4), the model will have high accuracy and be confident in its predictions. Hence, maximizing $\mathbb{O}$ increases the model's reliability, which is crucial for a delicate healthcare application.

4. Proposed Architecture

The DL pipeline, i.e., the proposed architecture developed for the heartbeat sound classification, comprises five stages, i.e., dataset acquisition, data preprocessing, feature extraction, the proposed classification model, and evaluation. The aim is, that during the evaluation stage, the proposed architecture maximizes $\mathbb{O}$ in Equation (4). The first stage is data acquisition, where data is collected from sources such as stethoscope pro and digiscope (digital stethoscope). Then, feature extraction is done on the preprocessed data using MFCCs in our proposed work of heartbeat sound classification. Data is further preprocessed by filtering, normalization, and downsampling. Later, we pass these features into the proposed classification model in the fourth step. Finally, our proposed classification model based on the attention mechanism is evaluated on whether $\mathbb{O}$ in Equation (4) is maximized based on various performance metrics as mentioned in Section 5.

Figure 1 shows the three-layered proposed architecture, i.e., the application layer, the data layer, and the intelligence layer. The first layer of the proposed architecture is the application layer, which shows the potential application areas where heartbeat sound classification can be utilized to monitor users' health status. The second layer is the data layer, which shows how we can extract data from the patients or people under consideration within an application of the application layer. This layer deals with data augmentation, pre-processing, and feature extraction for providing balanced data to the intelligence layer of the proposed DL model in the third layer. The final layer is the intelligence layer, and is responsible for processing the data acquired by the data layer and providing accurate and timely feedback to the user; our proposed CNN+BiGRU model lies within this layer. A detailed description of the three-layered proposed architecture is shown in the subsequent section.

4.1. Application Layer

The application layer is the first layer in our proposed architecture, as presented in Figure 1. This layer shows the application areas where our proposed DL model can be applied to prevent sudden abnormal heart conditions and make the user aware of the same, so that precautions can be taken. These applications include hospitals, government offices, ambulances, nursing homes, and gymnasiums—but they are not limited to them. Hospitals are the primary field of application as there is a frequent need to measure the heart sounds of patients feeling inconvenienced in terms of health. Patients not in intensive care can be given devices that continuously monitor their heartbeat sound in hospitals. This helps prevent unwanted scenarios where a patient might suffer from a severe heart ailment just because they are not deemed necessary to be in intensive care. Similarly, ambulances and nursing homes that do not have expensive and intensive equipment to measure heart sound, such as those in rural or poor regions, can be provided with the proposed cost-effective sensors where our DL model can alert upon abnormality detection and proper care may be provided. In gymnasiums, for instance, such devices can be given to gym users to prevent sudden heart attacks as the device would alert any peak in heartbeat, denoting the abnormal condition in the user. Hence, the user can stop exercising and reduce putting more strain on their heart by taking a rest. In government organizations, too, such devices can help provide a timely diagnosis to the poor, who generally have to choose government hospitals for their treatment. This might not be a problem in developed nations, but in developing ones, many government-run hospitals need proper facilities.

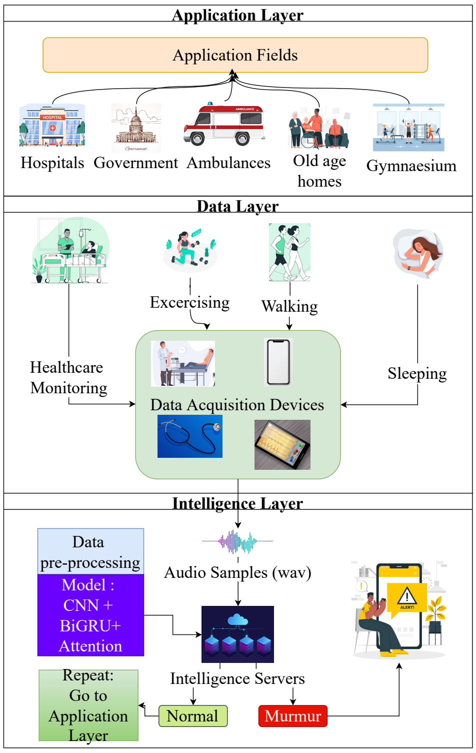


Figure 1. The proposed architecture.

4.2. Data Layer

The second layer of the proposed architecture is the data layer, which deals with data acquisition in real time. We have discussed methods that can be used to acquire the audio of the heart sounds for each of the applications in the application layer. Our proposed DL model requires data acquisition devices to provide raw audio data. The acquisition devices [50], such as a digital or analog stethoscope, electro-mechanical film (EMFi) transducer, smartphones, etc., can be used to acquire the raw audio data. For example, for applications that are under healthcare monitoring, we can utilize a digital stethoscope to obtain the data. Nevertheless, for applications where a person is not in healthcare monitoring, one can use smartphones with apps that read the data when placed in a pocket close to the heart. Moreover, one may use other wearable devices that specialize in acquiring inputs, such as thin and lightweight EM devices that can cling to t-shirts and read the audio data; such as the EMFi transducer. The EMFi is an elastic cellular polypropylene film sensitive to the dynamic forces usually exerted on its surface as it is thin and only a few dozen micrometers long. As a result, when the force is exerted on the material by heart vibrations, the transducer converts the mechanical vibrations into electrical signals. These mechanical vibrations cause variations in the thickness of the material, which further causes charge creation, which additionally creates voltage at the electrodes. Moreover, the mechanical signals are converted to electrical signals via this charge or voltage creation. These electrical signals are further amplified, so that heart sounds are obtained [51]. After acquiring the data, we can pass it for inference to the proposed model built within the sensor itself, referring to them as intelligent sensors.

4.2.1. Dataset Description

Data is collected from the CirCor DigiScope Phonocardiogram dataset [52], which is publicly available at physionet.org [53]. It has two categories of heartbeat sounds comprising normal and murmur sounds for 5282 recordings collected from 4 central auscultation regions of 1568 patients. The dataset has been annotated for each of the four central regions of interest, such as the aortic valve (AV), tricuspid valve (TV), pulmonic valve (PV), and mitral valve (MV) into murmur or not murmur for each region for each of the patients. This meta-data of the annotation for each patient has been provided in a comma-separated value (CSV) file. Since the data is a part of the competition, only the training data is posted at this project's time. Hence, the data is available for 942 patients only.

The recordings have an audio clip timing duration of 4.8 to 80.4 s, with a mean recording duration of 22.9 s and a standard deviation of 7.4 s. Considering this data distribution, we have considered reading the audio clips for 25 s for our proposed architecture. For many patients, sound clips are unavailable for all four regions, i.e., AV, TV, MV, and PV, and hence, we have not read data for a patient with no clear description of the murmur's presence or absence. This leads to acquiring audio clips with 499 murmur and 2508 normal recordings, with 3007 recordings obtained from 942 patients in total (leaving out those with an unknown presence or absence of murmur). From 3007 recordings, we have considered 489 recordings for the murmur class, as the rest are too long. The reason being training the proposed architecture with the data distribution of 489 recordings for murmur and 2508 for normal recordings leads to a class imbalance problem. To deal with this class imbalance problem, we have applied undersampling. With the undersampling, we used 489 recordings for the murmur class and 489 randomly selected recordings for the normal class. Moreover, these 978 recordings (489 for murmur and normal both) are then divided into training and testing data by a ratio of 70:30 with the help of the sci-kit-learn library, leading to the acquisition of raw data with 684 audio training clips and 294 audio testing clips that have an even distribution of both the classes of data. This distribution of data is represented in Equation (5), where dataset, train, test, Num(normal$_{old}$), Num(normal$_{new}$), Num(murmur), Num(dataset), Num(train), and Num(test) represent the dataset, train and test datasets, number of normal samples before undersampling, number of normal samples after undersampling, the number of murmur samples, number of samples in the dataset, train and test datasets, and number of samples in train and test data, respectively.

$$
\begin{gathered}
\text{Num(murmur)} \leftarrow 489, \\
\text{Num}(\text{normal}_{old}) \leftarrow 2508, \\
\text{Num}(\text{normal}_{old}) \xrightarrow{\text{Undersampling}} \text{Num}(\text{normal}_{new}), \\
\text{Num}(\text{normal}_{new}) \leftarrow 489, \\
\text{Num(dataset)} \leftarrow \text{Num(murmur)} + \text{Num}(\text{normal}_{new}), \\
\text{dataset} \xrightarrow{\text{Split (70:30)}} \text{train, test}, \\
\text{Num(train)} \leftarrow 684, \\
\text{Num(test)} \leftarrow 294
\end{gathered}
\tag{5}
$$

4.2.2. Data Augmentation

Data augmentation helps improve the DL models' performance even in sparse dataset cases. There are many perks to data augmentation, which include dealing with overfitting, handling imbalanced datasets by the generation of synthetic data of the minority class, enhancing model accuracy as the model learns generalization in data points as more availability of data is there, and removing the problem of new data acquisition for the same. Since we only have 684 training samples after the class imbalance problem was solved via undersampling, we employ data augmentation to fight overfitting. We use data augmentation to generate synthetic data from the training data via three approaches, i.e., time stretching, pitch shifting, and audio shifting. First, a random amount of time stretching, pitch, and audio shifting is executed on each original audio clip to generate two new audio clips. This results in the total training data becoming three times the original size of 684 clips. Hence, we apply data augmentation to create an additional 1368 clips, surmounting a total of 2052 clips. This addition can be represented as follows.

$$\begin{gathered}\mathrm{Num}(\mathrm{train}_{old}) \leftarrow 684,\\ \mathrm{train}_{old} \xrightarrow{\text{Data augmentation}} \mathrm{train}_{new},\\ \mathrm{Num}(\mathrm{train}_{new}) \leftarrow \mathrm{Num}(\mathrm{train}_{old}) + 1368\end{gathered} \tag{6}$$

where train_{old}, train_{new}, $\mathrm{Num}(\mathrm{train}_{old})$, and $\mathrm{Num}(\mathrm{train}_{new})$ represent the training dataset before data augmentation, the training dataset after data augmentation, the number of samples in previous data, and the number of samples in new train data. The considerable difference in terms of performance provided by data augmentation is depicted in Section 5. The three types of data augmentation techniques applied are described below.

1. Time stretch—this transformation is used to change the duration or speed of the signal without altering the original pitch. We can change the audio by speeding up the audio signal or slowing it down. If the selected rate is less than 1, the audio is slowed down and if it is greater than 1, it is sped up. We have used the librosa library for applying this transformation with the rate selection done uniformly between 0.8 and 1.2. We have set the probability of application of this transformation to 0.5, which means that the transformation is only applied with half the probability. Figure 2a,b shows the effects of time stretch.
2. Pitch shift—this transformation is used to change the pitch or perform pitch shift by changing the sound pitch up or down without altering the tempo. We can change the audio signal by changing the pitch by reducing or increasing the semitones. We randomly select the semitones in the range of -4 to +4. Again, we set the probability of application of this transformation to 0.5, which means that the transformation is only applied with half the probability. Figure 2a,c shows the effects of pitch shift.
3. Audio shift—this transformation is applied to shift the audio samples forward or backward, with or without any rollover. The shifting is done within a fraction of -0.5 to +0.5 of the audio signal. Furthermore, we set the probability of application of this transformation to 0.5, which means that the transformation is only applied with half the probability. Figure 2a,d shows the effects of audio shifting. As shown by comparison, the audio is the same but it is shifted by some amount.

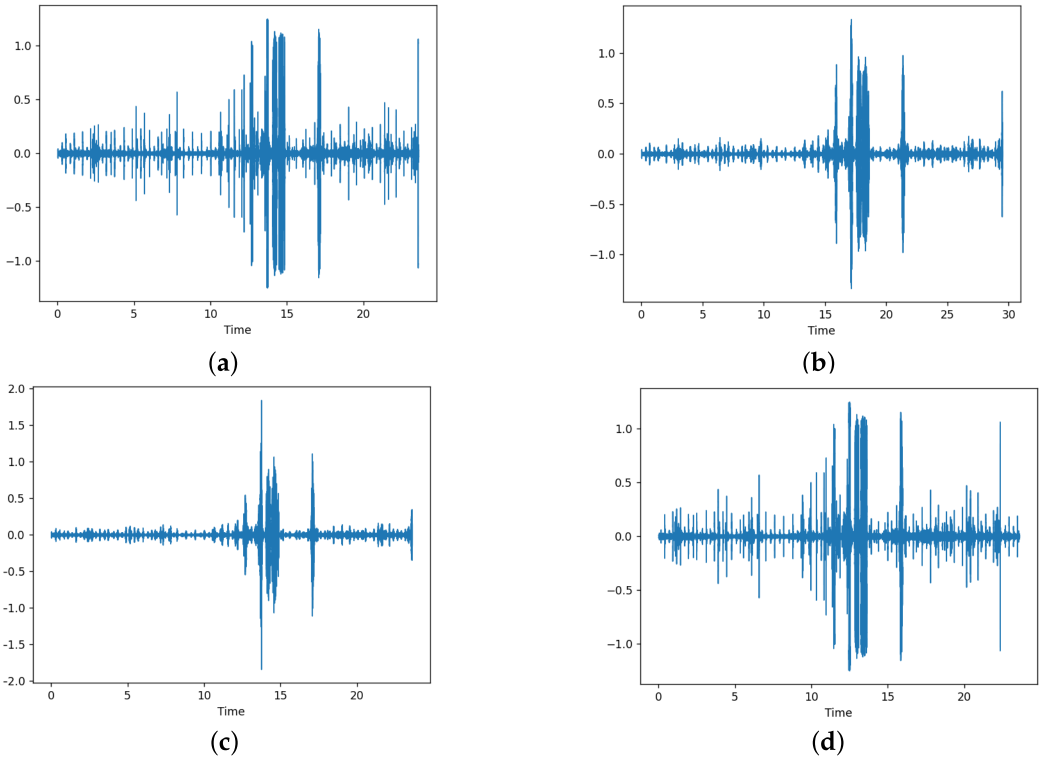


Figure 2. Data augmentation of heartbeat audio sounds. (**a**) Original audio. (**b**) After time stretch. (**c**) After pitch shift. (**d**) After audio shift.

4.2.3. Data Pre-Processing

Data pre-processing involves filtering, normalizing, and downsampling the given audio signal.

Filtering is performed on a given audio dataset to remove noise generated due to various environmental conditions during the recording process.

$$\text{Noisy audio} \xrightarrow{\text{Audio noise removal}} \text{Clear audio} \tag{7}$$

Normalization is done to improve the training process by normalizing every category of heartbeat sound in the range of +1 to −1. Each data point in the audio file is given a new value via the equation mentioned below.

$$\mathrm{dp}_{new} = 2 \times \frac{\mathrm{dp}_{old} - \mathrm{min}_{old}}{\mathrm{max}_{old} - \mathrm{min}_{old}} - 1 \tag{8}$$

where dp_{new}, dp_{old}, max_{old} and min_{old} represents the new value of the datapoint, the old value of the datapoint, a maximum value of all the old values of datapoints, and minimum value of all the old values of data points, respectively.

Down-sampling of the signal is accomplished to a sampling rate of 22,050, along with a bandpass filter having a frequency range of 30 to 1200 Hz.

$$\text{Original}_{44,100} \xrightarrow{\text{downsampling}} \text{Compressed}_{22,050} \tag{9}$$

where $\text{Original}_{44,100}$ and $\text{Compressed}_{22,050}$ represents the original audio at the sampling rate of 44,100 and the compressed audio after halving the sampling rate.

These steps are helpful in the removal of noise and getting the maximum out of the given data before feeding it to the next feature extraction stage.

4.2.4. Feature Extraction

The feature extraction stage is crucial because we want to train the proposed architecture on time series data. If we try to train it directly on the raw time series data it would lead to a vanishing gradient problem and a very long training time per epoch. The vanishing gradient problem occurs because there are 551,250-time samples for a 25-s audio clip if sampling is done at the rate of 22,050 samples per second. With so many samples to train, even with GRU [54] (which specializes in remembering the context for long dependencies), the model cannot propagate the error back through time, and a vanishing gradient occurs. Moreover, the training time is very high if we use the DL approach to learn features from the raw data. Due to these two drawbacks, we provide the DL model with the MFCC [55], a compressed version of raw data in the form of coefficients representing the complete data information. Another alternative is to employ a chromagram [56], which is also a feature extraction method like MFCC, but which leads to unstable training, as described in Section 5. Hence, MFCC is the preferred feature extractor. MFCC is one of the essential features for audio classification, compression, and other audio-related processing tasks. MFCCs are effective in audio classification as they are collectively made by Mel-frequency cepstrum (MFC). MFCCs are usually derived from the spectral representation of a particular frequency clip. Standard signal processing techniques cannot be applied to the audio signal due to its non-stationary nature, which is eased by the application of MFCCs. The critical difference between cepstrum and MFC is that the frequency band is equidistant from the mel scale. Moreover, a wrapper uses frequency wrapping to efficiently represent the given audio clip. MFCCs are usually derived using the following steps.

- Divide the raw audio into a set of frames ($\mathbb{F}$).

$$\text{Raw audio} \xrightarrow{\text{windowing}} \mathbb{F} \tag{10}$$

- Take the Fourier transform or fast Fourier transform (FFT) of the signal to get the power spectrogram for each frame.

$$f \xrightarrow{\text{Power spectrum}} PS_f;\ \forall f \in \mathbb{F} \tag{11}$$

  where PS_f represents the power spectrum of frame f.
- Map power of spectrum onto the Mel scale with the use of triangular overlapping windows. Mapping onto the Mel scale is represented below.

$$\text{M(input)} = 1127 \times \log_e\left(\frac{\text{input}}{700} + 1\right) \tag{12}$$

$$\text{PS}_f \xrightarrow{\text{M(PS}_f)} \text{MS}_f;\ \forall f \in \mathbb{F} \tag{13}$$

  where MS_f represents the Mel scale values for each frame.
- Calculate logs of powers for each Mel frequency.

$$\text{MS}_f \xrightarrow{\log|\text{MS}_f|} \text{LMS}_f;\ \forall f \in \mathbb{F} \tag{14}$$

  where $|\text{MS}_f|$ represents the power of the Mel frequencies and LMS_f represents the log of the corresponding Mel scale values.
- Derive discrete cosine transform (DCT) of the list of Mel log powers.

$$\text{LMS}_f \xrightarrow{\text{DCT(LMS}_f)} \text{MFCC}_f;\ \forall f \in \mathbb{F} \tag{15}$$

  where MFCC_f represents the MFCCs obtained for a frame of an audio file.

These MFCCs can be fed directly into the DL model. The model tends to learn meaningful features without facing the problem of vanishing gradient, which significantly reduces the time of reliable training models. Figure 3 represents the MFCC for a audio signal represented in Figure 2a. Visually, it is a heatmap representing tge time and coefficient (color represents the coefficient value) on their x-axis and y-axis, respectively. However, it is to be noted that the shown MFCC has only 15 coefficients (for better understanding), whereas, for our proposed work, we are using 70 coefficients. After the conversion of the raw audio data file to MFCC, we obtain an MFCC from an audio file read for 25 s with a sampling rate of 22,050, having the dimension of (1077, 70) because we are using hop_length = 512 and number of samples per FFT to be 2048 while calculating the MFCC. This compression of data is represented below.

$$(551{,}250,\ 1) \xrightarrow{\text{MFCC}} (1077,\ 70) \tag{16}$$

Now, after the compression via MFCC, we get (1077 × 70), i.e., 75,390 datapoints, which is significantly less than 551,250 datapoints.

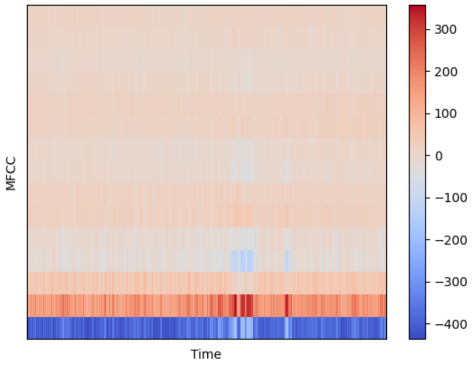


Figure 3. Mel-frequency cepstral coefficients.

4.3. Intelligence Layer

The third layer in the proposed architecture is the intelligence layer, which deals with applying the proposed DL model to provide feedback on the data obtained from the data layer. As shown in Figure 1, the raw audio samples are passed to intelligence servers that include the inference model. This inference model performs the data preprocessing and feature extraction (to get the MFCC) and passes it to the DL model consisting of CNN + BiGRU with an attention model for inference on the audio sample. The result of this inference is evaluated and an alert is sent to the user if there is a detection of murmur or abnormality. If the audio is normal, the inference process starts again with more recent data. Hence, the intelligence layer is the core layer of the proposed architecture. We need to train the model first for the model to undergo inference. Figure 4 describes the training phase. We have first split the raw audio data into training and testing data at a 70:30 ratio. Moreover, we have applied undersampling to counter the class imbalance problem. The reason is that the audio samples of the murmur class are less available. The dataset description mentioned in Section 4.2.1 gives a more detailed process. Later, we apply data augmentation to this training data to counter the overfitting problems. After data augmentation, we extract the MFCC from the obtained audio waves and pass it to the model. Now, the model is trained and validated against the test data to measure its performance against various evaluation metrics explained in the result and analysis. After the model training, finally, the model is stored on a disk for getting equipped with smart sensors.

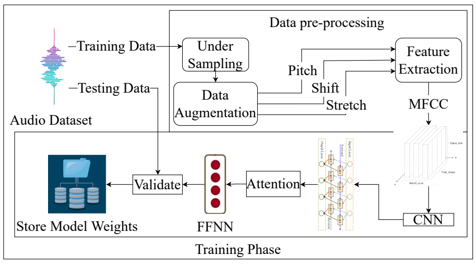


Figure 4. The training phase of the proposed architecture.

The architecture of the proposed DL model consists of a combination of CNN, BiGRU, and attention layers. In the input layer, we provide MFCCs with 70 coefficients. Before passing the GRU units to work upon the time series data, we apply 2D convolution layers with batch normalization and dropout with a probability of 0.3 to prevent overfitting and leaky ReLU with a parameter value of 0.3 as the non-linearity. The CNN layer is expected to extract key MFCC coefficients and provide to the BiGRU layers in the form of time series data [57,58]. From this data, the BiGRU learns important features and passes them to another BiGRU layer via a dropout (to prevent overfitting) to provide final features extracted from MFCCs to a feed-forward neural network (FFNN) to make the prediction. The FFNN's first layer uses tanh activation, whereas the second layer uses sigmoid activation as it is a binary classification problem. The various layers used and their functionality are explained in brief as follows.

1. 2D convolution layer—After adding the bias to an intermediate convolved output, a 2D convolution layer convolves over an input image to produce an output. This creates an output image obtained by the convolution of a learnable kernel matrix on the input image. This is similar to a dense layer but is done on 2D images and has the advantage of parameter sharing. The equation below represents this operation,

$$\mathbb{Y} = \mathbb{F}(\mathbb{X} \bigotimes \mathbb{W}) + b \tag{17}$$

   where $\mathbb{Y}$ represents the output image, $\mathbb{X}$ represents the input image, $\mathbb{W}$ represents the weight matrix (kernel), $\mathbb{F}$ represents the activation function, $\bigotimes$ represents the convolution operation, and b represents the bias.
2. Batch normalization layer—the batch normalization layer is used to normalize the inputs in the input layer such that the output standard deviation is close to 1. In contrast, the mean of the output is maintained to be close to 0. This layer scales the input by a learnable scale factor and offsets it by a learnable offset. This layer is generally used between the neural and output activation functions. This layer helps reduce the network's sensitivity and speeds up the training speed.
3. Dropout layer—the dropout layer is used to counter overfitting. This layer drops or makes the input units null and void in the network by a certain probability p set as a hyperparameter. The dropped units do not participate in the network's forward and backward propagation during the current iteration. This helps in reducing overfitting because the model has exponentially many network architectures to train within the same network. After all, which unit is dropped at each iteration is not fixed. At the time of inference, however, the unit's outputs are multiplied by the same probability p for forward propagation to represent that unit's contribution to inferencing.
4. GRU—they continually capture long-term dependencies in data using memory blocks. Each memory block has two gates, the update, and the reset gate, for performing the operations of updating the current memory cell state and deciding the amount of

previous information to forget, respectively. The equations followed by a GRU unit are as follows.

$$\begin{aligned} \mathbb{C}_t &= \mathbb{T}(\mathbb{W}_{ch} \times (\mathbb{R} * \mathbb{H}_{t-1}) + \mathbb{W}_{cx} \times \mathbb{X}_t + b_c) \\ \mathbb{U} &= \mathbb{S}(\mathbb{W}_{uh} \times \mathbb{H}_{t-1} + \mathbb{W}_{ux} \times \mathbb{X}_t + b_u) \\ \mathbb{R} &= \mathbb{S}(\mathbb{W}_{rh} \times \mathbb{H}_{t-1} + \mathbb{W}_{rx} \times \mathbb{X}_t + b_r) \\ \mathbb{H}_t &= \mathbb{U} * \mathbb{C}_t + (1 - \mathbb{U}) * \mathbb{H}_{t-1} \end{aligned} \tag{18}$$

where the update gate is denoted as $\mathbb{U}$ and the reset gate is denoted with $\mathbb{R}$, respectively. Further, the hidden state is denoted with $\mathbb{H}_t$ for a timestamp t, $\mathbb{C}_t$ means cell state for time t, and input features are denoted with $\mathbb{X}_t$, which are fed to the cell. $\mathbb{W}_{ch}$, $\mathbb{W}_{cx}$, $\mathbb{W}_{uh}$, $\mathbb{W}_{ux}$, $\mathbb{W}_{rh}$, $\mathbb{W}_{rx}$ are the weights, and b_c, b_u, b_r are the biases, which are obtained from the backpropagation algorithm (equipped in the neural network). $\times$ represents matrix multiplication and $*$ denotes element-wise dot product. $\mathbb{T}$ is the tanh and $\mathbb{S}$ is the sigmoid activation functions to show the probability of the neuron being active or inactive, respectively. Figure 5 represents the GRU unit.

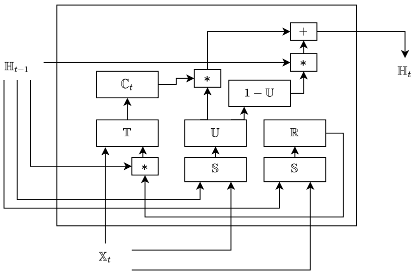


Figure 5. GRU unit.

5. Bidirectional GRU—GRUs help address the problem of vanishing and exploding gradient present in the case of RNN and help retain long-term information from past time steps. Bi-GRU is the extension of GRU that works in both directions, incorporating past and future time steps. Bi-GRU is composed of two GRU layers propagating in forward and backward directions. This helps us achieve improved performance in sequential decision-making problems by utilizing the complete context of the problem. Figure 6 illustrates the Bi-GRU's structure.

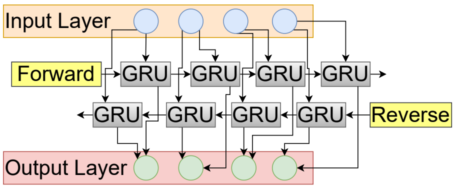


Figure 6. Bi-GRU architectural view.

The BiGRU structure can be individually broken down into two different layers, i.e., the forward propagation layer and the backward propagation layer. Both layers differ only in the direction of the context they witness before making a prediction. The forward layer predicts a timestep considering the context before that timestep, i.e., the previous context. In contrast, the backward layer predicts a timestep considering only the context that occurs after that timestep i.e., future context. Hence, at any timestep, we have both the past and future context; consequently, the model makes better predictions. As seen in Figure 6, the GRU unit used in each layer is the same as in Equation (18). Now, let us consider the hidden output at any timestep 't' for the forward layer's GRU unit and the corresponding backward layer's GRU unit to be $\mathbb{H}_{ft}$ and $\mathbb{H}_{bt}$, respectively. The output at the timestep, say $\mathbb{Y}_t$, can be given as follows.

$$\mathbb{Y}_t = \mathbb{T}(\mathbb{W}_{yhf} \times \mathbb{H}_{ft} + \mathbb{W}_{yhb} \times \mathbb{H}_{bt} + b_y) \tag{19}$$

where $\mathbb{W}_{yhf}$, $\mathbb{W}_{yhb}$, and b_y represent the weight matrix between the forward layer and output, the weight matrix between the backward layer, and bias for the output layer, respectively. $\mathbb{T}$ represents the tanh activation function.

6. Attention—the attention mechanism has applications in word recognition, image captioning, and many other related tasks. With the help of the attention mechanism, BiGRU can decide which part of the sound clip should be "attended". The overall mechanism of attention helps the deep learning model to be decisive in specific time steps or parts of data while ignoring irrelevant parts or time steps of data. The attention mechanism works based on extracting discriminative information, helping to improve the performance of RNN/GRU-based architecture by focusing on certain parts of the data. The attention mechanism captures discriminative information for our problem of sound classification as the complete data does not contribute equally towards representing a particular class of sound clip. The attention mechanism provides aid to traditional BiGRU by significant improvement of the performance of the deep learning model with reduced computational cost. The attention mechanism is helpful in the generation of a dense vector that represents the output produced after proper attention is given to the required timesteps. The equations of the attention mechanism are as follows.

$$\mathbf{h_i} = \mathrm{GRU}(\mathbf{s_i}), i \in [1, L] \tag{20}$$

where $\mathbf{h_i}$ is the hidden state column vector for the given input, L denotes the number of cells in GRU network, and $\mathbf{s_i}$ is the input. Further, an attention mechanism is adopted to capture the hidden states of the network, as shown in Equation (21)

$$\begin{gathered} \mathbf{u}_{\mathrm{i}} = \tanh(\mathbf{W_s h_i} + \mathbf{b_s}) \\ \alpha_{\mathrm{i}} = \frac{\exp(\mathbf{u_i})}{\sum_j \exp(\mathbf{u}_{\mathrm{j}})} \\ \mathbf{v} = \sum_i \alpha_{\mathrm{i}} \mathbf{h_i} \end{gathered} \tag{21}$$

where the attention layer's output is represented by $\mathbf{v}$ and $\mathbf{W_s}$ and $\mathbf{b_s}$ are trainable weights and biases.

7. Dense—the dense layer is one of the most basic neural networks used to change the dimension of a 1D layer by following certain functions for calculation. In the case of heartbeat classification, the dense layer is used as a layer of neurons having input weights following some linear function for generating output. The function followed by a dense layer could be illustrated as,

$$\mathbb{Y} = \mathrm{f}(\mathrm{X} \times \mathrm{w} + \mathrm{b}) \tag{22}$$

where $\mathbb{Y}$ represents the output layer, X represents the input layer, f represents the activation function, w represents the weight matrix, and b represents the bias vector.

Next, we look in brief at the various activation functions used.

1. Leaky ReLU—this activation function is an extension of the ReLU activation function where, if the input is negative, then the output is a negative number scaled down by a parameter instead of zero, as was the case in ReLU activation. The equation below describes the leaky ReLU activation function.

$$\text{output} = \begin{cases} \text{input} & \text{input} \geq 0 \\ \pi \times \text{input} & \text{input} < 0 \end{cases} \tag{23}$$

where π represents the parameter used to scale the output. π is to be set as a hyperparameter.

2. Tanh—used to apply a non-linearity that squeezes the output to a unique value in the range of −1 to +1, corresponding to a unique input value. The equation for the application of this non-linearity can be given below.

$$\text{output} = \frac{2}{1+e^{-2\times \text{input}}} - 1 \tag{24}$$

3. Sigmoid—this activation function is used to apply such a non-linearity that squeezes the output to a unique value in the range of 0 to 1, corresponding to a unique input value. The equation for the application of this non-linearity can be given below.

$$\text{output} = \frac{1}{1+e^{-\text{input}}} \tag{25}$$

   Finally, we propose the usage of the Adam optimizer (the reason for the usage of the Adam optimizer is depicted in the result and analysis section) and discuss the binary cross entropy loss function next.

4. Binary cross-entropy loss—a combination of the sigmoid activation function and cross-entropy loss used for binary classification problems. Hence, the below equation depicts the calculation of the binary cross entropy loss for a sample.

$$\text{Loss} = -\text{t} \times \log(\text{x}) - (1-\text{t}) \times \log(1-\text{x}) \tag{26}$$

   where t is the label value, i.e., either 0 (for normal) or 1 (for murmur), depending upon the label of the current input audio signal, and x is the input obtained from the previous layer after application of the sigmoid activation function.

Algorithm 1 represents the algorithm of the entire training process.

Algorithm 1 : Attention-based Bi-GRU architecture for audio classification

```
hyper parameters ← manual selection
loss_function ← binary cross entropy
optimizer ← Adam
N ← numberOfEpochs
dataset ← loadDataset()
model ← buildModel(hyperparameters)
compileModel(model, loss_function, optimizer)
prevAccuracy ← 0
batch_size ← hyperparameters.get('batch_size')
While N > 0 do
  iterations ← (numOfSamples(dataset)/batch_size)
  While iterations ≠ 0 do
     batch ← createMiniBatch(dataset)
     Foreach input ∈ batch do
       mfcc ← getMFCC(input)
       loss  + = train(model, input)
    endFor
    backpropagateLoss(model, loss)
    iterations ← iterations − 1
  endWhile
  N ← N − 1
  accuracy ← validate(model, getTestData(dataset))
  If accuracy > prevAccuracy do
    storeModelWeights(model)
    prevAccuracy ← accuracy
  endIf
endWhile
```

5. Result and Analysis

This section discusses the implementation details, including experimentation setup, tools, and performance analysis of the proposed architecture with different evaluation metrics.

5.1. Experimentation Setup and Tools

The implementation of the proposed architecture (CNN + BiGRU) along with all the other models, i.e., CNN, LSTM, BiGRU, and CNN + BiLSTM, is done on Google Colaboratory. The specifications of the computing power include a Graphical Processing Unit (GPU)—Tesla T4 16 GB, Disk size—80 GB, Processor—Intel(R) Xeon(R) Central Processing Unit (CPU) @ 2.20 GHz, and Random Access Memory (RAM)—13.3 GB. The proposed architecture is built upon Python (version 3.7.14). Moreover, TensorFlow (version 2.8.2) and Keras (version 2.8.0) architecture aid in building the model architectures. Furthermore, the Matplotlib (version 3.2.2) library is used for all the visualizations to aid data analysis. The pandas library (version 1.3.5) is used for working with CSV files. For all the other mathematical functions such as floating point operations and data manipulations such as reshaping, the NumPy (version 1.21.6) library is used. Finally, for all the audio data manipulations, i.e., audio preprocessing steps such as spectrograms and MFCCs, the librosa library [59] (version 0.8.1) is used. Each model's performance is estimated on various evaluation metrics such as accuracy, recall, F1 score, precision, and the ROC curve. The sci-kit-learn library (version 1.0.2) calculates these metrics and analyzes each model.

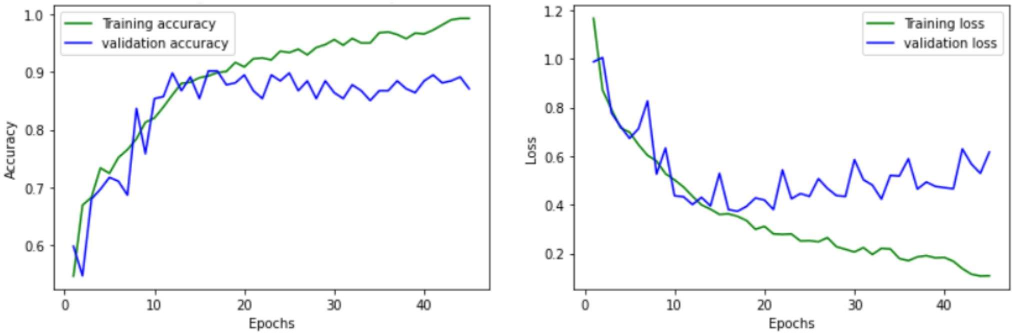
5.2. Simulation Analysis

The proposed architecture and the other models mentioned in Table 2 are trained for 100 epochs and a batch size of 32. Now, the learning rate defines the rate at which the loss or error of the model is back-propagated to update the weights. Hence, if the learning rate is high, the model learns faster, but there is a chance of the model overshooting the minima in the gradient descent. Whereas, with a low learning rate, the model converges very slowly or does not converge due to a vanishing gradient. Hence a lower learning rate is generally used to fine-tune the model weights. With these points in mind, the learning rate used is 0.001 at the beginning of the training (which is not high enough to overshoot the minima in gradient descent). It is reduced later (when a plateau region in the gradient descent is reached) in the hope that the models learn finer details from the MFCCs, i.e., the model's weights are fine-tuned. Three optimizers (methods of back-propagation of the error) are evaluated with the proposed architecture: Adam, SGD (Stochastic Gradient Descent), and RMSprop, with Adam giving optimum results. With the Adam optimizer, the default parameter values are used, i.e., the value of beta1 (exponential decay of the first moment estimates), beta2 (exponential decay of second-moment estimates), and epsilon (a tiny number preventing any division by zero) are 0.9, 0.999, and 10×10^{-8} respectively. With SGD, the momentum is kept at zero.

With RMSprop, the values of momentum, rho (it is the decay factor of the learning rate and reduces the effect of the learning rate eventually to learn finer details from the data and get closer to the minima), epsilon are 0, 0.9, and 1×10^{-7}, respectively. The loss function calculates the loss or error of the model's prediction and is the binary cross entropy loss function. Moreover, early stopping, which stops the training when a sufficient gap is created in the training and validation accuracy, is implemented to prevent the overfitting of the model at the time of training. Figure 7 shows the model's accuracy and loss curve analysis that justifies the early stopping. Figure 7 shows that the accuracy of the proposed architecture on the validation set increases linearly with the training accuracy in the early stages, but, later on, a gap is created due to the overfitting of the model (even after the measures taken to combat overfitting). Due to this reason, we have applied early stopping to prevent the model from learning the noise from the training set and retain its generalization ability on the validation set. The same argument, as is explained for the accuracy curve, goes with the loss curve. After the model is overfitted, the loss of the validation set increases as the model wants to perform better on the training set and hence starts memorizing the training data and unwanted noise peculiar to training data. Thereby, reducing its generalization capabilities on the validation set results in an increase in error, which justifies the use of early stopping to prevent overfitting and unnecessary wastage of computing resources to train the model further, even though the scope of improvement is less. The overall validation accuracy of our proposed CNN+BiGRU with an attention model is superior to the other models, as is evident in Table 2.

Table 2. Performance metrics of different deep learning models for heartbeat sound classification. Precision, Recall, and F1-Scores are macro averages. Columns of "Accuracy", "Precision", "Recall", "F1-score", and "AUC" are calculated on the validation set.

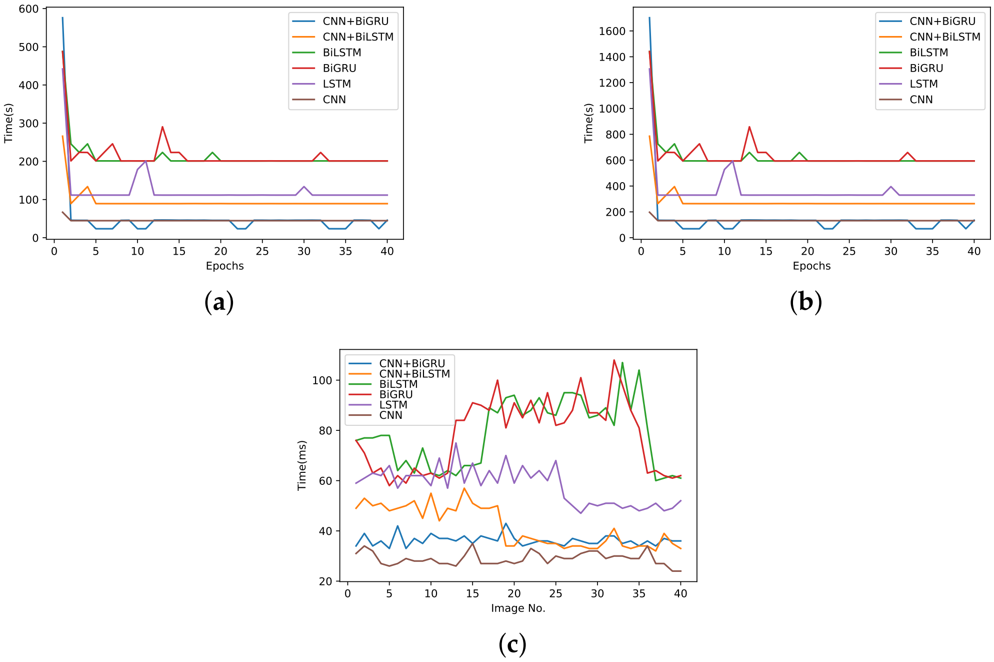
Model	Train Accuracy	Validation Accuracy	Precision	Recall	F1-Score	AUC	Loss	Accuracy/Loss
CNN	87%	84%	86%	85%	85%	0.85	0.43	1.95
LSTM	87%	82%	83%	82%	82%	0.84	0.522	1.61
BiGRU	96%	87%	87%	87%	87%	0.87	0.54	1.61
BiLSTM	95%	85%	86%	85%	85%	0.85	0.5633	1.50
CNN + BiLSTM	99%	85%	85%	85%	85%	0.85	0.84	1.01
[29]	-	75.1%	-	-	-	-	-	-
[25]	78.6%	73.7%	-	-	65.7%	-	-	-
CNN + BiGRU (Proposed model)	**94%**	**90%**	**90%**	**91%**	**90%**	**0.90**	**0.45**	**2**



(**a**) (**b**)

Figure 7. Model evaluation shows that the proposed architecture is able to achieve significant validation accuracy ≈ 90% with very low error. (**a**) Training and validation accuracy. (**b**) Training and validation loss.

Moreover, we have shown the time each model takes to train on the dataset for epochs, with and without data augmentation, along with the computation complexity of the models for giving the inference on MFCC from the test data in Figure 8. As shown, the proposed architecture has much less computational time complexity for training than other models. For inference, the proposed architecture also has much less computational complexity than other models, with tight competitors being CNN and CNN+BiLSTM. CNN is expected to give better inference time complexity because of the parallel nature of the model and hence the superiority. Simple BiGRU and BiLSTM-based recurrent networks are taking more time to give inference because of the sequential nature of the models. Simple LSTM has only one direction to deal with and hence has lesser model complexity overall. However, it is still sequential, thus, with higher time complexity than CNN-based models but lower than the bidirectional models. The introduction of some parallel structure of CNN to BiLSTM and BiGRU makes CNN+BiLSTM and CNN+BiGRU faster than other raw LSTM and GRU-based models. Between CNN+BiLSTM and CNN+BiGRU, these models have almost similar time complexity, and no significant difference exists between them. However, as described in Table 2, the accuracy of CNN+BiGRU outperforms the CNN+BiLSTM and CNN models.



(**a**)

(**b**)

(**c**)

Figure 8. Computation complexity analysis that represents how much time is taken by various models to train for 40 epochs and the time taken to give an inference on a single sample. (**a**) Training Time vs. Epochs (No Augmentation). (**b**) Training Time vs. Epochs (With Augmentation). (**c**) Inference Time vs. Image No. (Id).

5.3. Evaluation Metrics

The evaluation metrics are the metrics against which different models are compared. Compared to other models, these metrics define the effectiveness of a model on the overall performance in solving the classification task at hand. Figure 9 illustrates the confusion matrix for performance evaluation of the binary classification task with our proposed architecture on the validation set. The confusion matrix is the overall summary from which all the other comparison metrics, such as accuracy, precision, recall, and F1 score, as discussed below, are derived.

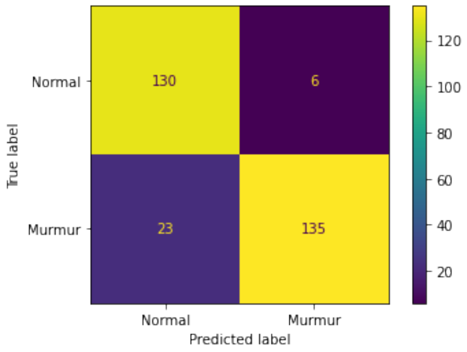


Figure 9. Confusion matrix.

We defined specific terms related to the confusion matrix as below:

1. True Positive (α)—labels that are predicted as murmur by the model are truly murmur (bottom right cell of the confusion matrix).
2. False Positive (β)—labels that are predicted as murmur by the model but are normal (top right cell of the confusion matrix).
3. True Negative (γ)—labels that are predicted as normal by the model are truly normal (top left cell of the confusion matrix).

4. False Negative (δ)—labels that are predicted as normal by the model but are murmur (bottom left cell of the confusion matrix).

Various metrics (derivable from the confusion matrix) are used to select the proposed architecture over the other models.

1. Accuracy—a measure or the ratio of the total number of correct predictions to the total number of predictions performed by a model. A higher accuracy means that a model is more accurate in making predictions. As is evident from Table 2, the validation accuracy of the proposed architecture outperforms others with an acceptable difference in the training accuracy, indicating that the proposed architecture does not overfit as quickly as other models and produces better predictions. Formally, the accuracy is mathematically defined as,

$$\text{Accuracy} = \frac{\alpha + \gamma}{\alpha + \beta + \gamma + \delta} \tag{27}$$

2. Precision—the measure of correctly predicted true positive samples out of all predicted true positive samples. Simply, it measures how many predictions are made by the model belonging to a class from that class. Hence, the precision should be as high as possible. From Table 2, it is evident that the proposed architecture outperforms all the other models in precision with a high margin. Formally, precision is defined as,

$$\text{Precision}(\Gamma) = \frac{\alpha}{\alpha + \beta} \tag{28}$$

3. Recall—the measure of how many samples belonging to a class that the model predicts as belonging to that class, or simply, it is the ratio of the number of samples of a class that a model identifies out of the total number of samples of that class in the dataset. A high and desirable recall means that a model can extract a high number of samples of a class out of all the samples of that class from the dataset. From Table 2, it is evident that the proposed architecture outperforms all the other models in recall with a high margin. Formally, recall is defined as,

$$\text{Recall}(\rho) = \frac{\alpha}{\alpha + \delta} \tag{29}$$

4. F1 score—the relation between precision and recall derived from calculating the harmonic mean of precision and recall. The F1 score considers both precision and recall, and a high F1 score is desirable. For the model's performance, we generally consider the F1 score to be the prime metric of distinction. As is evident from Table 2, the proposed architecture outperforms all the other models in the F1 score with a high margin. Formally, it is defined as,

$$\text{F1 score} = \frac{2 \times \Gamma \times \rho}{\Gamma + \rho} \tag{30}$$

Table 2 illustrates the above-mentioned performance metrics for murmur and regular heartbeat sounds using our proposed CNN+BiGRU with an attention mechanism against other deep learning models. Figure 10 illustrates the receiver operating characteristic (ROC) curve of our proposed architecture along with the area under the curve (AUC) on the validation set to be 0.9, indicating that the model performs exceptionally well on the validation set. This AUC parameter tells us how much the model can distinguish between different classes; hence, the more the area, the better the model. As is evident in Table 2, the AUC of the proposed architecture outperforms the other models.

Figure 11 compares various optimizers used with the proposed architecture during training. We can see that the Adam [60] optimizer produces the best results, as is expected because Adam optimization is an extension of the stochastic gradient descent (SGD) and has the benefits of both the Adagrad and RMSprop optimizers. This results in Adam reaching the optimum results in a short time frame.

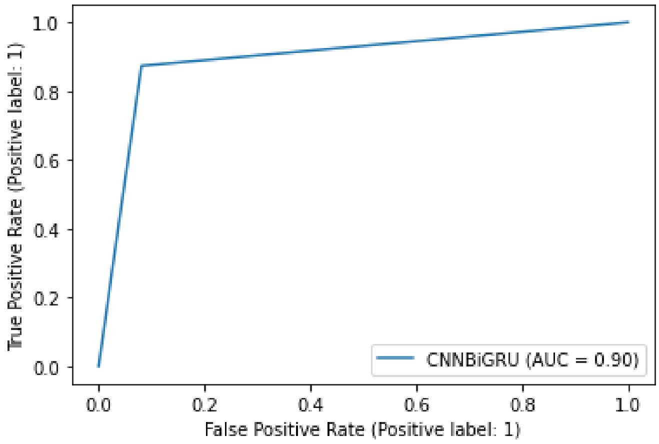


Figure 10. ROC score for CNN+BiGRU.

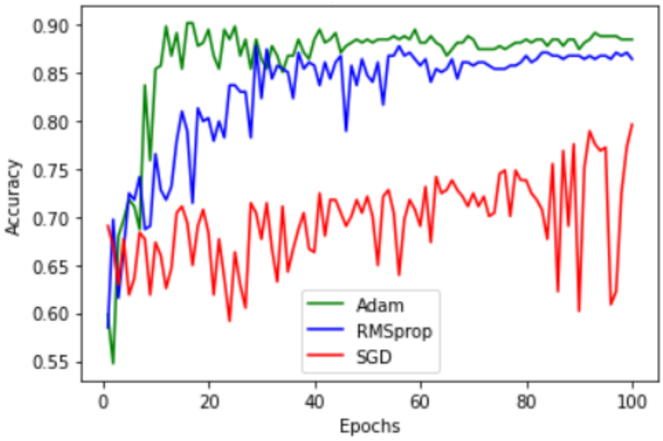


Figure 11. The model's validation accuracy with different optimizers.

Furthermore, we can also see from Figure 12 the reason behind selecting MFCC as a feature selector. It is seen that the training with MFCC is very stable and converges quickly. On the other hand, the training with chromagram as a feature selector gives us very slow and unstable training.

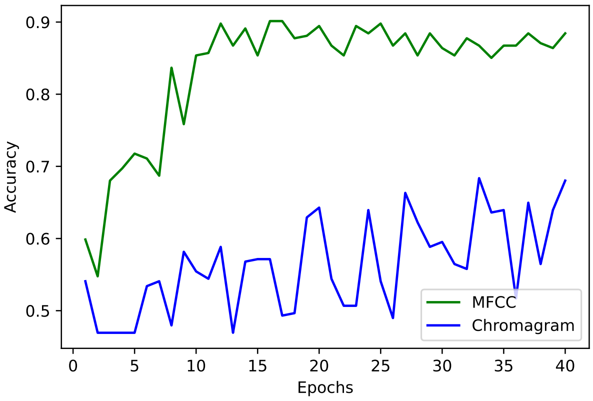


Figure 12. Validation accuracy of the proposed approach with different feature extractors.

Finally, data augmentation improves the validation accuracy to a great extent and helps combat overfitting. Figure 13 shows this positive effect of data augmentation.

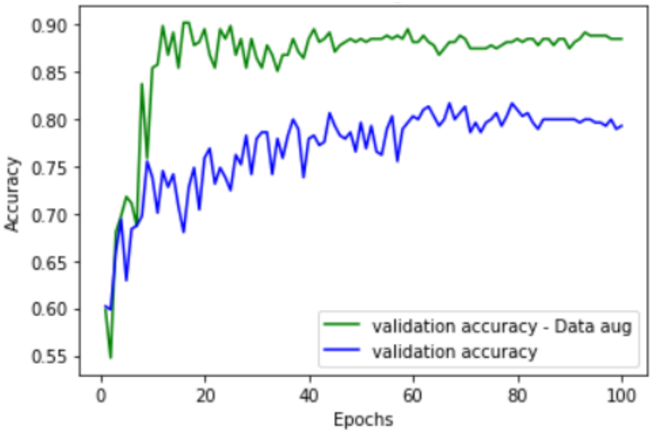


Figure 13. Validation accuracy with and without data augmentation.

This happens due to the model's overfitting earlier due to fewer data. Whereas, with data augmentation, the overfitting concern while training the model is somewhat alleviated. Finally, coming back to the original objective as mentioned in (4) i.e., maximizing $\mathbb{O}$, we can see from the last column in Table 2 that our proposed architecture maximizes the set objective. Moreover, we have also compared our results to specific experiments from others' work, as shown in Table 2. Our model outperforms Ref. [25] by a large margin on the same dataset. The result shown in this work is for the CirCor DigiScope Phonocardiogram Dataset, which is from the Physionet Heart Sound Classification challenge 2022. The authors have applied self-supervised learning considering lower images in the dataset along with 1D-CNN based on the architecture of Refs. [61,62]. They achieved poorer results than our proposed approach because, although the images are lesser, they have applied a very shallow model. Moreover, they have not applied proper data augmentation and preprocessing. As a result, the model cannot learn the feature of the data and hence we see a lower training and validation accuracy. Our proposed model CNN+BiGRU is lightweight but still sufficiently deep enough to learn the appropriate features and give classifications with high accuracy.

Similarly, when comparing with Ref. [29], we can see that their accuracy and F1-score are also lower compared to our proposed approach. They applied proper data augmentation and preprocessing to calculate MFCC before passing the raw audio to the classifier. However, they have only implemented a CNN-based model structure with a perceptron layer to deal with the wide features. However, they have not applied any RNN-based model. Our proposed approach uses a CNN-based model to decrease the input complexity and then the sequential BiGRU layers to get the sequential relationship out of the data. Moreover, our proposed approach is optimized with the attention mechanism, which is neither used in Ref. [25] nor Ref. [29]. Hence, our proposed architecture outperforms other approaches. Therefore, based on all the above metrics and the analysis, we have selected CNN+BiGRU attention as our proposed architecture.

6. Conclusions

The paper proposes a novel CNN+BiGRU attention-based approach for the classification of heartbeat sound to provide the first level of screening to patients for cardiac disorders with the advent of deep learning to classify heart disease. The proposed approach can effectively determine the urgency of treatment in cases of a particular cardiac symptom. We have performed data augmentation with time stretch, pitch shift, and audio shift to improve the model's performance, deal with overfitting, and balance the dataset by generating synthetic data from the minority class. The proposed deep learning model applies filtering, downsampling, feature extraction, i.e., MFCC, and other data processing techniques to effectively utilize the given dataset. The proposed architecture was evaluated using different evaluation metrics, such as statistical measures (e.g., accuracy, precision, recall, etc.), validation accuracy with and without data augmentation, and validation accuracy with different optimizers. Our approach outperforms other state-of-the-art approaches providing a validation accuracy of about 90% on the CirCor DigiScope Phonocardiogram dataset.

For future work, we would like to extend our proposed architecture to other variants of audio recognition problems, including audio clips gathered from various clinical sources, labs, and different acoustic environments for audio classification. Moreover, we would explore transformer-based architectures to try and improve prediction accuracy.

Author Contributions: Conceptualization: P.S., H.Y., N.G., T.V., L.G. and S.T.; Writing—original draft preparation: A.N., S.D., P.S., R.S. and M.S.R.; Methodology: S.T., V.M., T.V., S.D. and L.G.; Writing—review and editing: S.T., H.Y., P.S., N.G. and V.M.; Investigation: M.S.R., S.T., V.M., T.V. and A.N.; Supervision: S.T., L.G., R.S. and V.M.; Visualization; H.Y., P.S., N.G. and M.S.R.; Software: S.D., P.S., S.T. and V.M. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was partially supported by UEFISCDI Romania and MCI through BEIA projects AutoDecS, SOLID-B5G, T4ME2, DISAVIT, PIMEO-AI, AISTOR, MULTI-AI, ADRIATIC, Hydro3D, PREVENTION, DAFCC, EREMI, ADCATER, MUSEION, FinSESCo, iPREMAS, IPSUS, U-GARDEN, CREATE and by European Union's Horizon Europe research and innovation program under grant agreements No. 101037866 (ADMA TranS4MErs). This work is supported by Ministry of Research, Innovation, Digitization from Romania by the National Plan of R & D, Project PN 19 11, Subprogram 1.1. Institutional performance-Projects to finance excellence in RDI, Contract No. 19PFE/30.12.2021 and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN).

Data Availability Statement: No data is associated with this.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cardiovascular Diseases (CVDs)—who.int. Available online: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (accessed on 28 February 2023).
2. Vora, J.; Tanwar, S.; Tyagi, S.; Kumar, N.; Rodrigues, J.J. HRIDaaY: Ballistocardiogram-Based Heart Rate Monitoring Using Fog Computing. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
3. Fuchs, F.D.; Whelton, P.K. High blood pressure and cardiovascular disease. *Hypertension* **2020**, *75*, 285–292. [CrossRef] [PubMed]
4. Nabel, E.G. Cardiovascular Disease. *N. Engl. J. Med.* **2003**, *349*, 60–72. [CrossRef] [PubMed]
5. Ciumărnean, L.; Milaciu, M.V.; Negrean, V.; Orăşan, O.H.; Vesa, S.C.; Sălăgean, O.; Iluţ, S.; Vlaicu, S.I. Cardiovascular Risk Factors and Physical Activity for the Prevention of Cardiovascular Diseases in the Elderly. *Int. J. Environ. Res. Public Health* **2022**, *19*, 207. [CrossRef] [PubMed]
6. Rodgers, J.L.; Jones, J.; Bolleddu, S.I.; Vanthenapalli, S.; Rodgers, L.E.; Shah, K.; Karia, K.; Panguluri, S.K. Cardiovascular Risks Associated with Gender and Aging. *J. Cardiovasc. Dev. Dis.* **2019**, *6*, 19. [CrossRef]
7. Hanna, I.R.; Silverman, M.E. A history of cardiac auscultation and some of its contributors. *Am. J. Cardiol.* **2002**, *90*, 259–267. [CrossRef]
8. Tanwar, S.; Vora, J.; Kaneriya, S.; Tyagi, S.; Kumar, N.; Sharma, V.; You, I. Human Arthritis Analysis in Fog Computing Environment Using Bayesian Network Classifier and Thread Protocol. *IEEE Consum. Electron. Mag.* **2020**, *9*, 88–94. [CrossRef]
9. Vincent, R. I look into the chest: History and evolution of stethoscope. *J. Pract. Cardiovasc. Sci.* **2022**, *8*, 168.
10. Jiang, Z.; Choi, S. A cardiac sound characteristic waveform method for in-home heart disorder monitoring with electric stethoscope. *Expert Syst. Appl.* **2006**, *31*, 286–298. [CrossRef]
11. Kaneriya, S.; Lakhani, D.; Brahmbhatt, H.U.; Tanwar, S.; Tyagi, S.; Kumar, N.; Rodrigues, J.J.P.C. Can Tactile Internet be a Solution for Low Latency Heart Disorientation Measure: An Analysis. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6. [CrossRef]
12. Abbas, Q.; Hussain, A.; Baig, A.R. Automatic Detection and Classification of Cardiovascular Disorders Using Phonocardiogram and Convolutional Vision Transformers. *Diagnostics* **2022**, *12*, 3109. [CrossRef]
13. Babu, K.A.; Ramkumar, B.; Manikandan, M.S. Automatic Identification of S1 and S2 Heart Sounds Using Simultaneous PCG and PPG Recordings. *IEEE Sens. J.* **2018**, *18*, 9430–9440. [CrossRef]
14. Kumar, D.; Carvalho, P.; Antunes, M.; Gil, P.; Henriques, J.; Eugenio, L. A New Algorithm for Detection of S1 and S2 Heart Sounds. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France, 14–19 May 2006; Volume 2, p. II. [CrossRef]
15. Zeinali, Y.; Niaki, S.T.A. Heart sound classification using signal processing and machine learning algorithms. *Mach. Learn. Appl.* **2022**, *7*, 100206. [CrossRef]

16. Chen, W.; Sun, Q.; Chen, X.; Xie, G.; Wu, H.; Xu, C. Deep Learning Methods for Heart Sounds Classification: A Systematic Review. *Entropy* **2021**, *23*, 667. [CrossRef]
17. Chauhan, K.; Jani, S.; Thakkar, D.; Dave, R.; Bhatia, J.; Tanwar, S.; Obaidat, M.S. Automated Machine Learning: The New Wave of Machine Learning. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 205–212. [CrossRef]
18. Ren, Z.; Qian, K.; Dong, F.; Dai, Z.; Yamamoto, Y.; Schuller, B.W. Deep Attention-based Representation Learning for Heart Sound Classification. *arXiv* **2021**, arXiv:2101.04979.
19. Mukherjee, U.; Pancholi, S. A Visual Domain Transfer Learning Approach for Heartbeat Sound Classification. *arXiv* **2021**. [CrossRef]
20. Kui, H.; Pan, J.; Zong, R.; Yang, H.; Wang, W. Heart sound classification based on log Mel-frequency spectral coefficients features and convolutional neural networks. *Biomed. Signal Process. Control* **2021**, *69*, 102893. [CrossRef]
21. Gupta, R.; Patel, M.M.; Shukla, A.; Tanwar, S. Deep learning-based malicious smart contract detection scheme for internet of things environment. *Comput. Electr. Eng.* **2022**, *97*, 107583. [CrossRef]
22. Jamil, S.; Rahman, M. A Novel Deep-Learning-Based Framework for the Classification of Cardiac Arrhythmia. *J. Imaging* **2022**, *8*, 70. [CrossRef] [PubMed]
23. Xiang, M.; Zang, J.; Wang, J.; Wang, H.; Zhou, C.; Bi, R.; Zhang, Z.; Xue, C. Research of heart sound classification using two-dimensional features. *Biomed. Signal Process. Control* **2023**, *79*, 104190. [CrossRef]
24. Keikhosrokiani, P.; Anathan, A.B.N.A.; Fadilah, S.I.; Manickam, S.; Li, Z. Heartbeat sound classification using a hybrid adaptive neuro-fuzzy inferences system (ANFIS) and artificial bee colony. *Digit. Health* **2023**, *9*, 20552076221150741. [CrossRef]
25. Ballas, A.; Papapanagiotou, V.; Delopoulos, A.; Diou, C. Listen2YourHeart: A Self-Supervised Approach for Detecting Murmur in Heart-Beat Sounds. *arXiv* **2022**. [CrossRef]
26. Ren, Z.; Qian, K.; Dong, F.; Dai, Z.; Nejdl, W.; Yamamoto, Y.; Schuller, B.W. Deep attention-based neural networks for explainable heart sound classification. *Mach. Learn. Appl.* **2022**, *9*, 100322. [CrossRef]
27. Saraswat, D.; Bhattacharya, P.; Verma, A.; Prasad, V.K.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Sharma, R. Explainable AI for Healthcare 5.0: Opportunities and Challenges. *IEEE Access* **2022**, *10*, 84486–84517. [CrossRef]
28. Tariq, Z.; Shah, S.K.; Lee, Y. Feature-Based Fusion Using CNN for Lung and Heart Sound Classification. *Sensors* **2022**, *22*, 1521. [CrossRef] [PubMed]
29. Lu, H.; Yip, J.B.; Steigleder, T.; Grießhammer, S.; Sai Jitin Jami, N.; Eskofier, B.; Ostgathe, C.; Koelpin, A. A Lightweight Robust Approach for Automatic Heart Murmurs and Clinical Outcomes Classification from Phonocardiogram Recordings. In Proceedings of the Computing in Cardiology (CinC), Tampere, Finland, 4–7 September 2022; Volume 49.
30. Milani, M.; Abas, P.E.; De Silva, L.C.; Nanayakkara, N.D. Abnormal heart sound classification using phonocardiography signals. *Smart Health* **2021**, *21*, 100194. [CrossRef]
31. Er, M.B. Heart sounds classification using convolutional neural network with 1D-local binary pattern and 1D-local ternary pattern features. *Appl. Acoust.* **2021**, *180*, 108152. [CrossRef]
32. Xiao, B.; Xu, Y.; Bi, X.; Zhang, J.; Ma, X. Heart sounds classification using a novel 1-D convolutional neural network with extremely low parameter consumption. *Neurocomputing* **2020**, *392*, 153–159. [CrossRef]
33. Tiwari, S.; Sapra, V.; Jain, A. Heartbeat sound classification using Mel-frequency cepstral coefficients and deep convolutional neural network. In *Advances in Computational Techniques for Biomedical Image Analysis*; Koundal, D., Gupta, S., Eds.; Academic Press: Cambridge, MA, USA, 2020; pp. 115–131. [CrossRef]
34. Boulares, M.; Alotaibi, R.; AlMansour, A.; Barnawi, A. Cardiovascular Disease Recognition Based on Heartbeat Segmentation and Selection Process. *Int. J. Environ. Res. Public Health* **2021**, *18*, 952. [CrossRef]
35. Schmidt, S.E.; Holst-Hansen, C.; Graff, C.; Toft, E.; Struijk, J.J. Segmentation of heart sound recordings by a duration-dependent hidden Markov model. *Physiol. Meas.* **2010**, *31*, 513. [CrossRef]
36. Chen, T.; Kuan, K.; Celi, L.A.; Clifford, G.D. Intelligent Heartsound Diagnostics on a Cellphone Using a Hands-Free Kit. In Proceedings of the 2010 AAAI Spring Symposium: Artificial Intelligence for Development, Stanford, CA, USA, 22–24 March 2010; Technical Report SS-10-01; AAAI: Palo Alto, CA, USA, 2010.
37. Moukadem, A.; Dieterlen, A.; Hueber, N.; Brandt, C. A robust heart sounds segmentation module based on S-transform. *Biomed. Signal Process. Control* **2013**, *8*, 273–281. [CrossRef]
38. Safara, F.; Doraisamy, S.; Azman, A.; Jantan, A.; Abdullah Ramaiah, A.R. Multi-level basis selection of wavelet packet decomposition tree for heart sound classification. *Comput. Biol. Med.* **2013**, *43*, 1407–1414. [CrossRef]
39. Ari, S.; Hembram, K.; Saha, G. Detection of cardiac abnormality from PCG signal using LMS based least square SVM classifier. *Expert Syst. Appl.* **2010**, *37*, 8019–8026. [CrossRef]
40. Zhang, W.; Han, J.; Deng, S. Heart sound classification based on scaled spectrogram and tensor decomposition. *Expert Syst. Appl.* **2017**, *84*, 220–231. [CrossRef]
41. Deng, S.W.; Han, J.Q. Towards heart sound classification without segmentation via autocorrelation feature and diffusion maps. *Future Gener. Comput. Syst.* **2016**, *60*, 13–21. [CrossRef]
42. Banerjee, M.; Majhi, S. Multi-class Heart Sounds Classification Using 2D-Convolutional Neural Network. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–6. [CrossRef]

43. Gomes, E.; Bentley, P.; Coimbra, M.; Pereira, E.; Deng, Y. Classifying heart sounds: Approaches to the PASCAL challenge. In Proceedings of the HEALTHINF 2013-Proceedings of the International Conference on Health Informatics, Barcelona, Spain, 11–14 February 2013; pp. 337–340.
44. Raza, A.; Mehmood, A.; Ullah, S.; Ahmad, M.; Choi, G.S.; On, B.W. Heartbeat Sound Signal Classification Using Deep Learning. *Sensors* **2019**, *19*, 4819. [CrossRef]
45. Zheng, Y.; Guo, X.; Ding, X. A novel hybrid energy fraction and entropy-based approach for systolic heart murmurs identification. *Expert Syst. Appl.* **2015**, *42*, 2710–2721. [CrossRef]
46. Yaseen.; Son, G.Y.; Kwon, S. Classification of Heart Sound Signal Using Multiple Features. *Appl. Sci.* **2018**, *8*, 2344. [CrossRef]
47. Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M.D. Large-Scale Weakly Supervised Audio Classification Using Gated Convolutional Neural Network. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 121–125. [CrossRef]
48. Xu, K.; Zhu, B.; Kong, Q.; Mi, H.; Ding, B.; Wang, D.; Wang, H. General audio tagging with ensembling convolutional neural network and statistical features. *arXiv* **2018**, arXiv:1810.12832.
49. Chaudhary, S.; Kakkar, R.; Jadav, N.K.; Nair, A.; Gupta, R.; Tanwar, S.; Agrawal, S.; Alshehri, M.D.; Sharma, R.; Sharma, G.; et al. A taxonomy on smart healthcare technologies: Security framework, case study, and future directions. *J. Sens.* **2022**, *2022*, 1863838. [CrossRef]
50. Miller, D.J.; Sargent, C.; Roach, G.D. A Validation of Six Wearable Devices for Estimating Sleep, Heart Rate and Heart Rate Variability in Healthy Adults. *Sensors* **2022**, *22*, 6317. [CrossRef]
51. Karki, S.; Kaariainen, M.; Lekkala, J. Measurement of heart sounds with EMFi transducer. In Proceedings of the 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, 22–26 August 2007; pp. 1683–1686. [CrossRef]
52. Oliveira, J.; Renna, F.; Costa, P.; Nogueira, M.; Oliveira, A.C.; Elola, A.; Ferreira, C.; Jorge, A.; Rad, A.B.; Reyna, M.; et al. *The CirCor DigiScope Phonocardiogram Dataset*, Version 1.0.0; PhysioNet: Cambridge, MA, USA , 2022.
53. Oliveira, J.; Renna, F.; Costa, P.D.; Nogueira, M.; Oliveira, C.; Ferreira, C.; Jorge, A.; Mattos, S.; Hatem, T.; Tavares, T.; et al. The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 2524–2535. [CrossRef]
54. Shah, H.; Shah, D.; Jadav, N.K.; Gupta, R.; Tanwar, S.; Alfarraj, O.; Tolba, A.; Raboaca, M.S.; Marina, V. Deep Learning-Based Malicious Smart Contract and Intrusion Detection System for IoT Environment. *Mathematics* **2023**, *11*, 418. [CrossRef]
55. Gupta, S.; Jaafar, J.; Wan Ahmad, W.F.; Bansal, A. Feature Extraction Using Mfcc. *Signal Image Process. Int. J.* **2013**, *4*, 101–108. [CrossRef]
56. Bartsch, M.; Wakefield, G. Audio thumbnailing of popular music using chroma-based representations. *IEEE Trans. Multimed.* **2005**, *7*, 96–104. [CrossRef]
57. Hathaliya, J.; Parekh, R.; Patel, N.; Gupta, R.; Tanwar, S.; Alqahtani, F.; Elghatwary, M.; Ivanov, O.; Raboaca, M.S.; Neagu, B.C. Convolutional Neural Network-Based Parkinson Disease Classification Using SPECT Imaging Data. *Mathematics* **2022**, *10*, 2566. [CrossRef]
58. Hathaliya, J.J.; Modi, H.; Gupta, R.; Tanwar, S.; Sharma, P.; Sharma, R. Parkinson and essential tremor classification to identify the patient's risk based on tremor severity. *Comput. Electr. Eng.* **2022**, *101*, 107946. [CrossRef]
59. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.; Mcvicar, M.; Battenberg, E.; Nieto, O. librosa: Audio and Music Signal Analysis in Python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 July 2015; pp. 18–24. [CrossRef]
60. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
61. Papapanagiotou, V.; Diou, C.; Delopoulos, A. Chewing detection from an in-ear microphone using convolutional neural networks. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Republic of Korea, 11–15 July 2017; pp. 1258–1261. [CrossRef]
62. Papapanagiotou, V.; Diou, C.; Delopoulos, A. Self-Supervised Feature Learning of 1D Convolutional Neural Networks with Contrastive Loss for Eating Detection Using an In-Ear Microphone. In Proceedings of the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Virtual, 1–5 November 2021; pp. 7186–7189. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.