Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 4, October 2026, pp: 1296-1313 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

STFT-Based Multiclass Heart Sound Classification Using BiLSTM and CNN-BiLSTM Models

Noor S. Ali, Ehab Abdulrazzaq Hussein, and Laith Ali Abdul-Rahaim

Department of Electrical Engineering, University of Babylon, Hilla, Iraq

Corresponding author: Noor S. Ali (e-mail: eng.noor.salman@uobabylon.edu.iq), **Author(s) Email**: Ehab Abdulrazzaq Hussein (e-mail: dr.ehab@uobabylon.edu.iq); Laith Ali Abdul-Rahaim (e-mail: drlaithanzy@uobabylon.edu.iq)

Abstract **Cardiovascular diseases require early and reliable screening because manual auscultation may be affected by noise, subjective interpretation, and inter-observer variability. This study aimed to develop an STFT-based deep learning framework for multiclass phonocardiogram (PCG) classification. The proposed framework was designed to provide a reproducible evaluation procedure by combining standardized preprocessing, time–frequency feature extraction, and deep learning-based classification under the same experimental conditions. Unlike approaches that may evaluate segmented signals without clearly preserving recording-level separation, this study emphasizes a leakage-free splitting strategy to reduce the risk of overestimated performance and to provide a more reliable assessment of model generalization. The Yaseen PCG dataset, consisting of 1000 recordings from five classes (AS, MR, MS, MVP, and Normal), was divided using a leakage-free recording-level split before segmentation and spectrogram generation. After preprocessing, 2-second PCG segments with 50% overlap were converted into 128 × 128 STFT spectrograms and classified using BiLSTM and CNN-BiLSTM models. Both models were trained and tested using the same dataset split, preprocessing pipeline, and evaluation metrics, including accuracy, precision, recall, F1-score, specificity, and confusion matrices. The BiLSTM model achieved 92.36% accuracy in the final independent test run, while the CNN-BiLSTM model achieved 95.83%. Across three repeated runs, BiLSTM achieved 93.85% ± 1.47%, whereas CNN-BiLSTM achieved 95.94% ± 0.71%. These results show that CNN-BiLSTM provides higher and more stable classification performance for five-class PCG classification, while BiLSTM remains a simpler alternative for lightweight implementation. Overall, the proposed STFT-based framework provides a reliable approach for automated heart sound classification and may support future computer-aided cardiac screening applications.**

Keywords **Heart sound classification; PCG; deep learning; CNN; BiLSTM; STFT; spectrogram; multiclass classification**

I. Introduction

Cardiovascular diseases continue to represent a major public health challenge and remain among the leading causes of death worldwide. Early recognition of abnormal cardiac conditions is therefore important for timely diagnosis, clinical follow-up, and reduction of healthcare burden [1] [2]. In this context, heart sound analysis provides a practical screening approach because phonocardiogram (PCG) signals can be acquired using low-cost, non-invasive, and portable devices. PCG signals contain useful acoustic information related to the mechanical activity of the heart valves and cardiac cycles. Compared with more expensive imaging-based procedures, PCG-based analysis may be more suitable for preliminary screening, remote monitoring, and digital stethoscope applications [3] [4]. Nevertheless, conventional auscultation is highly dependent on clinician experience and may be affected by noise, subjective interpretation, and inter-observer variability [5] [6]. These factors make automated PCG classification an important research direction for supporting consistent and reproducible heart sound assessment. Since PCG recordings are non-stationary biomedical signals, their frequency components vary over time. Time–frequency analysis is therefore commonly used to represent heart sound characteristics more clearly. Among different time–frequency methods, the Short-Time Fourier Transform (STFT) provides a direct way to convert PCG signals into spectrograms while preserving both temporal and spectral information [7][8][9]. These spectrograms are suitable for deep learning because they transform one-dimensional acoustic signals into

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

structured two-dimensional representations that can be processed by image-based and sequence-based neural networks. Recent deep learning studies have reported promising performance in heart sound classification. Convolutional neural networks (CNNs) are effective in learning local spatial patterns from spectrogram images, whereas recurrent neural networks, including LSTM and BiLSTM models, are useful for modeling temporal dependencies in sequential biomedical signals [10][11] [12][13]. Hybrid CNN–BiLSTM models can further combine local time–frequency feature extraction with bidirectional temporal modeling, which is useful for capturing both short-term spectral patterns and longer temporal variations in PCG recordings [14][15][16].

Despite these advances, several limitations remain in previous PCG classification studies. Some studies have focused mainly on binary normal–abnormal classification rather than multiclass classification of specific valvular diseases. Other works have used complex transfer learning or transformer-based models, which may increase computational requirements and reduce suitability for lightweight screening systems [17][18][19]. In addition, many studies did not clearly explain whether data splitting was performed before or after segmentation. If segments from the same original PCG recording are distributed across training and testing sets, data leakage may occur and the reported performance may be overestimated. Therefore, a reproducible recording-level evaluation is necessary for a fair assessment of deep learning models in PCG classification.

The novelty of this work is not based on proposing a completely new neural network family, but on providing a leakage-free and reproducible evaluation framework for five-class PCG classification using STFT spectrograms. Unlike studies that mainly evaluate STFT-CNN, CNN-LSTM, transfer learning, or transformer-based approaches under different datasets and validation settings, this study compares BiLSTM and CNN–BiLSTM models under the same recording-level splitting strategy, preprocessing pipeline, STFT representation, and independent test evaluation. This design allows a clearer analysis of the trade-off between a simpler recurrent model and a hybrid spatial–temporal model for multiclass heart sound classification. The CNN–BiLSTM model is expected to improve discriminative feature learning by combining convolutional extraction of local spectrogram patterns with BiLSTM-based temporal modeling, while the BiLSTM model provides a lighter alternative with lower architectural complexity [20][21].

The aim of this study is to develop and evaluate an STFT-based deep learning framework for multiclass PCG classification. The proposed framework classifies five heart sound categories, namely Aortic Stenosis (AS), Mitral Regurgitation (MR), Mitral Stenosis (MS), Mitral Valve Prolapse (MVP), and Normal (N), using BiLSTM and CNN–BiLSTM models. A recording-level split was applied before segmentation and spectrogram extraction to reduce data leakage risk and to provide a more reliable estimation of model performance.

The main contributions of this study are summarized as follows. First, a leakage-free recording-level splitting strategy was applied so that all segments generated from the same original PCG recording remain within only one subset. Second, STFT spectrograms were used to represent PCG signals in the time–frequency domain while preserving information relevant to cardiac sound patterns. Third, BiLSTM and CNN–BiLSTM models were compared under the same experimental conditions for five-class heart sound classification. Fourth, repeated experiments were conducted to report the mean accuracy and standard deviation, rather than relying only on a single run. Finally, class-wise performance was analyzed to identify difficult pathological classes, especially MR and MVP, and to provide a more clinically meaningful interpretation of the classification results.

A. Literature Review

Ismail et al. [4] the authors proposed a heart sound classification system based on spectrogram representations and hybrid classification. They addressed the challenge of spectral overlap between normal heart sounds and other internal and external acoustic sources, such as extra heart sounds, extra systoles, murmurs, respiration sounds, and body motion artifacts. Their proposed approach included signal filtering, time segmentation, spectrogram generation, hybrid classification, and a voting-based decision mechanism. The method performed the analysis at both cycle level and signal level, which improved the reliability of the final classification decision. Evaluation on the public PASCAL 2011 dataset using 5-fold cross-validation achieved precision, recall, and accuracy values greater than 95%. Their results also demonstrated that short PCG recordings of approximately 2–3 seconds can be sufficient for effective heart sound classification. However, this study focused mainly on public benchmark evaluation and did not emphasize a leakage-free multiclass comparison between recurrent and hybrid models.

Tariq et al. [22] highlighted the importance of preprocessing and feature extraction in heart and lung sound classification using deep learning models. Their study transformed biomedical sound signals into informative audio representations, including spectrogram, MFCC, and chromagram features, before feeding them into CNN-based models. They

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

emphasized that converting audio signals into image-based feature representations helps CNNs learn discriminative patterns more effectively. Their work also addressed common challenges in biomedical sound classification, such as noise, limited data, and class imbalance, by applying data augmentation techniques including noise distortion, pitch shifting, and time stretching. The results showed that the fusion-based CNN framework improved classification performance, achieving up to 97% accuracy for augmented heart sound classification. This study supports the importance of proper preprocessing and time-frequency-based feature representation for achieving reliable performance in automatic heart sound classification.

Although this work demonstrated the value of image-based audio representations, it considered multiple biomedical sound types and did not focus specifically on five-class PCG classification. Al-Issa and Alqudah [23], investigated the effectiveness of hybrid deep learning models, particularly CNN-LSTM architectures, for classifying heart sound signals. By integrating Long Short-Term Memory (LSTM) networks with spatial feature extraction using CNNs, their approach enhanced classification performance. The study demonstrated the importance of combining spatial and sequential learning, especially for biomedical signals that exhibit both time-based and frequency-based characteristics. This finding supports the use of hybrid architectures for PCG classification, as CNN layers can extract discriminative spectrogram features while recurrent layers can model temporal variations in heart sound patterns. Such a combination is particularly useful for distinguishing between valvular diseases with overlapping acoustic characteristics. This supports the effectiveness of hybrid CNN–recurrent models in extracting spatial and temporal features from PCG signals.

Khan et al. [24], proposed a deep learning-based approach for automatic PCG signal classification using STFT-based spectrogram representations. Their study focused on distinguishing normal and abnormal heart sounds using two widely available public PCG datasets, namely PhysioNet/CinC 2016 and PASCAL 2011. The authors transformed PCG recordings into time-frequency spectrogram images and evaluated several CNN variants under different experimental settings, including training on PhysioNet, testing on PASCAL, using a combined PhysioNet-PASCAL dataset, and applying transfer learning to improve performance on noisy data. Their results demonstrated that STFT spectrograms can effectively represent the temporal and spectral characteristics of PCG signals. The best CNN model achieved high performance on the PhysioNet dataset with an accuracy of 95.75%, sensitivity of 96.3%, specificity of 94.1%, precision of 97.52%, and F1-score of 96.93%. Moreover, the transfer learning experiment achieved a precision of 96.98% on the noisy PASCAL dataset. Although this work confirmed the effectiveness of STFT spectrograms, the task was mainly normal-versus-abnormal classification, whereas the present study addresses a more challenging five-class PCG classification problem.

From the reviewed studies, it can be observed that several research gaps remain. Many existing approaches focus on binary classification or a limited number of classes, while fewer studies address multiclass PCG classification involving several valvular heart diseases. In addition, some works do not clearly report whether data splitting was performed at the recording level before segmentation, which may increase the risk of data leakage. Furthermore, highly complex transfer learning or transformer-based models may not always be suitable for lightweight clinical screening systems. Therefore, this study provided a leakage-free comparison of BiLSTM and CNN–BiLSTM models using STFT spectrograms for five-class heart sound classification.

II. Method

The proposed framework as shown in Fig.1 consists of five main stages: dataset preparation, leakage-free recording-level splitting, preprocessing and segmentation, STFT spectrogram generation, and deep learning classification using BiLSTM and CNN–BiLSTM models. The overall workflow was designed to reduce data leakage risk and improve reproducibility. In this study, all splitting operations were performed at the original PCG recording level before segmentation and spectrogram extraction. Therefore, segments generated from the same original recording were never shared between the training, validation, and testing subsets.

A. Data Collection

The heart sound recordings used in this study were obtained from the Yaseen PCG dataset reported by Yaseen et al. [25]. The dataset contains 1000 phonocardiogram (PCG) recordings distributed across five classes: Aortic Stenosis (AS), Mitral Regurgitation (MR), Mitral Stenosis (MS), Mitral Valve Prolapse (MVP), and Normal (N). Each class contains approximately 200 recordings, providing a balanced multiclass dataset for evaluating heart sound classification models.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

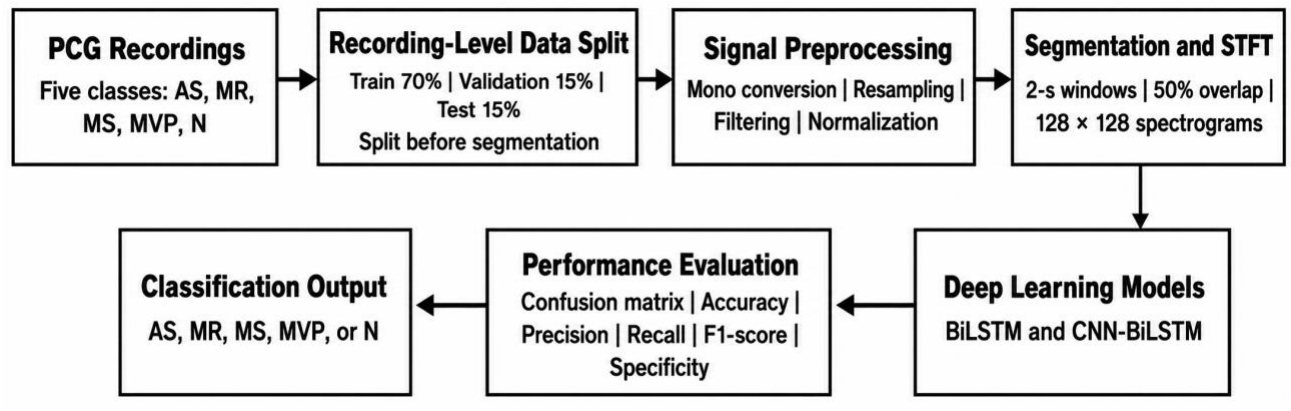


Fig. 1. Overall methodology of the proposed STFT-based heart sound classification framework.

Table 1. Dataset preparation and splitting configuration

Parameter	Description
Dataset	Yaseen PCG dataset [25]
Total recordings	1000 PCG recordings
Number of classes	5 classes
Classes	AS, MR, MS, MVP, and N
Recordings per class	Approximately 200 recordings per class
Split strategy	Recording-level split before segmentation
Training set	70% of original recordings
Validation set	15% of original recordings
Testing set	15% of original recordings
Segmentation	2-second windows
Overlap	50%
Feature representation	STFT spectrogram
Input size	128 × 128
Classification task	Five-class PCG classification

The dataset was divided into training, validation, and testing subsets using a 70%, 15%, and 15% ratio, respectively. Importantly, the split was performed at the original recording level before any segmentation or STFT spectrogram generation. This means that all 2-second segments extracted from one original PCG recording were assigned only to one subset. This strategy was adopted to prevent data leakage and to ensure that the independent test set contained recordings not used during model training or validation. Table 1 summarizes the dataset preparation and splitting configuration used in this study.

B. Data Processing

The preprocessing stage was applied separately within the training, validation, and testing subsets after the recording-level split. This ensured that no information from the test recordings was used during preprocessing or model training. The preprocessing pipeline included mono conversion, resampling, band-pass filtering, amplitude normalization, segmentation, and zero-padding when required. First, all PCG recordings were converted to a single-channel signal when necessary and resampled to 2000 Hz to maintain a consistent sampling frequency across all recordings. A band-pass filter in the range of 20–500 Hz was then applied to preserve the main frequency components of heart sounds while reducing very low-frequency baseline noise and high-frequency interference. After filtering, amplitude normalization was performed to reduce inter-

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

recording amplitude variation and to place all signals on a comparable scale. The filtered and normalized PCG recordings were segmented into fixed 2-second windows with 50% overlap. For recordings shorter than the required window length, zero-padding was applied to produce a consistent segment duration. Segmentation was performed only after the original recordings had been assigned to training, validation, or testing subsets. Therefore, all segments from the same recording remained within the same subset, preventing segment-level leakage between training and testing.

C. Time-Frequency Feature Extraction (STFT)

The Short-Time Fourier Transform (STFT) was used to convert the preprocessed one-dimensional PCG signals into two-dimensional time–frequency spectrograms. STFT is suitable for PCG analysis because heart sound signals are non-stationary, and their spectral characteristics change over time. Compared with purely time-domain representations, STFT spectrograms preserve both temporal and frequency information, which makes them appropriate for CNN-based spatial feature extraction and BiLSTM-based temporal modeling [24], [26], [27] Given a signal x(t), the STFT is defined Eq. (1) [28]:

$$STFT\{x(t)\}(\tau, w) = \int_{-\infty}^{\infty} x(t).w(t-\tau).e^{-jwt}\ dt \qquad (1)$$

where τ is the time shift and w(t-τ) are a sliding window function[29] (such as a Hamming window). In this study, each 2-second PCG segment was transformed into a spectrogram using a Hamming window of 25 ms, 50% overlap, and a 512-point FFT. The resulting magnitude spectrograms were converted into image-like representations and resized to 128 × 128 pixels. These resized STFT spectrograms were used as the input features for the proposed deep learning models. STFT spectrogram generation was performed separately inside each subset after the recording-level split to ensure that spectrograms derived from the same original recording did not appear in more than one subset [28] [29] [30] [31].

D. BiLSTM Architecture

The BiLSTM model was designed to capture temporal dependencies in STFT spectrogram sequences. Each spectrogram was treated as a time–frequency representation, where temporal frames provide sequential information and frequency bins provide spectral features. The model input consisted of resized STFT spectrograms with a size of 128 × 128. The BiLSTM architecture included a sequence input layer, two BiLSTM layers, dropout layers, fully connected layers, and a softmax output layer. The first BiLSTM layer used 64 hidden units with sequence output enabled to learn temporal dependencies across all time steps. A dropout layer with a rate of 0.5 was used after this layer to reduce overfitting. The second BiLSTM layer used 32 hidden units and returned the last time-step representation. This was followed by dense layers with 64 and 32 neurons using ReLU activation. Additional dropout layers with rates of 0.4 and 0.3 were used to improve generalization. The final fully connected layer contained five output neurons corresponding to the five PCG classes, followed by a softmax layer for multiclass classification [31] [30] [31] [32]. Table 2 summarizes the main layers of the BiLSTM model used for STFT-based multiclass PCG classification. The architecture was designed to capture temporal dependencies from the spectrogram sequence while reducing overfitting using dropout regularization.

Table 2. BiLSTM model architecture

Layer	Configuration	Output description
Input layer	128 × 128 STFT spectrogram sequence	Time–frequency input
BiLSTM layer 1	64 hidden units, sequence output	Temporal feature sequence
Dropout	Rate = 0.5	Regularization
BiLSTM layer 2	32 hidden units, last output	Final temporal representation
Dropout	Rate = 0.5	Regularization
Dense layer	64 neurons, ReLU	High-level representation
Dropout	Rate = 0.4	Regularization
Dense layer	32 neurons, ReLU	Compact representation
Dropout	Rate = 0.3	Regularization
Output layer	5 neurons, softmax	AS, MR, MS, MVP, N

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

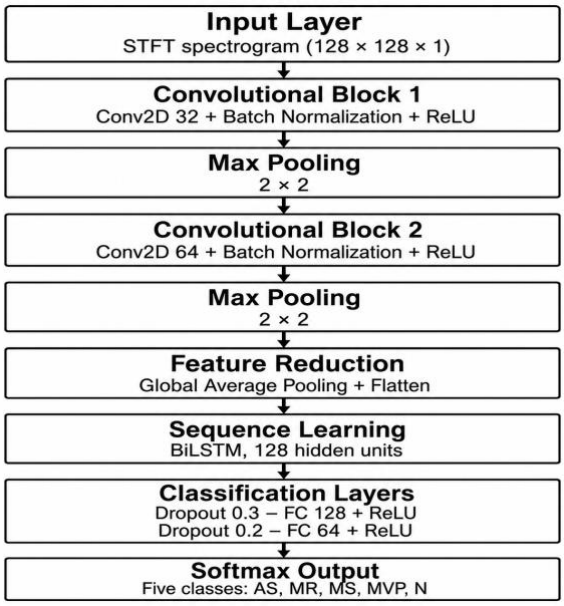


Fig. 2. Proposed CNN-BiLSTM architecture for STFT-based multiclass heart sound classification

E. CNN-BiLSTM Architecture

The CNN–BiLSTM model was designed to combine convolutional spatial feature extraction with bidirectional temporal modeling. The input to the model was a grayscale STFT spectrogram with a fixed size of 128 × 128 × 1. The CNN component extracted local time–frequency patterns from the spectrogram, while the BiLSTM component modeled sequential dependencies in the extracted feature representation [25] [26] .The convolutional part consisted of two main convolutional blocks. The first block used 32 filters to extract low-level time–frequency features, while the second block used 64 filters to learn higher-level representations. Each convolutional block included convolution, batch normalization, ReLU activation, and max-pooling. A global average pooling layer was then used to reduce the dimensionality of the feature maps and improve generalization. The extracted features were reshaped into a sequential format and passed to a BiLSTM layer with 128 hidden units. A dropout layer with a rate of 0.3 was applied after the BiLSTM layer. The classification part included fully connected layers with 128 and 64 neurons, ReLU activation, and dropout. The final output layer used five neurons with soft max activation for multiclass heart sound classification. Fig. 2. illustrates the proposed CNN–BiLSTM architecture used for STFT-based multiclass heart sound classification. The model first extracts local time–frequency features from the STFT spectrogram using two convolutional blocks with batch normalization and max-pooling. The extracted features are then summarized using global average pooling, reshaped into a sequential representation, and passed to a BiLSTM layer with 128 hidden units. Finally, dropout and fully connected layers are used before the softmax output layer for five-class classification

Table 3 illustrates the CNN–BiLSTM architecture used for STFT-based multiclass PCG classification. The convolutional blocks extract local time–frequency features from the spectrograms, while the BiLSTM layer models bidirectional temporal dependencies before final softmax classification.

F. Experimental Setup and Reproducibility

Table 3. CNN-BiLSTM model architecture

Layer	Configuration	Output description
Input layer	128 × 128 × 1 STFT spectrogram	Image-like time–frequency input
Convolution block 1	32 filters, ReLU, batch normalization, max-pooling	Low-level local features
Convolution block 2	64 filters, ReLU, batch normalization, max-pooling	Higher-level local features
Global average pooling	Feature map summarization	Reduced feature representation
Reshape layer	Converts CNN features into sequence form	Sequential feature input
BiLSTM layer	128 hidden units, last output	Bidirectional temporal representation
Dropout	Rate = 0.3	Regularization
Dense layer	128 neurons, ReLU	High-level representation
Dropout	Rate = 0.2	Regularization
Dense layer	64 neurons, ReLU	Compact representation

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

Table 3. (Continued)

Optimizer	**Adam**	**Adam**
Learning rate	0.001	0.0005
Mini-batch size	16	16
Epochs	40	40
Random seed	42	42
Repeated runs	3	3
Output classes	5	5

To improve reproducibility, the main experimental settings are summarized in one subsection. All experiments were conducted using MATLAB R2024b and the Deep Learning Toolbox on a workstation equipped with a single NVIDIA-based GPU and a multi-core CPU. The same dataset split, preprocessing pipeline, and evaluation protocol were used for both BiLSTM and CNN–BiLSTM models. The dataset was split at the original recording level using a 70:15:15 ratio for training, validation, and testing. After splitting, preprocessing, 2-second segmentation with 50% overlap, and STFT spectrogram generation were applied separately within each subset. The STFT parameters included a 25 ms Hamming window, 50% overlap, and a 512-point FFT. All spectrograms were resized to 128 × 128 pixels before being used as model inputs. The BiLSTM model was trained for 40 epochs using the Adam optimizer, an initial learning rate of 0.001, and a mini-batch size of 16. The CNN–BiLSTM model was also trained for 40 epochs using the Adam optimizer, a learning rate of 0.0005, and a mini-batch size of 16. Training and validation accuracy and loss curves were monitored to evaluate convergence behavior and possible overfitting. For reproducible dataset partitioning, the recording-level split was generated using rng(42). To reduce the influence of random initialization and training variability, each experiment was repeated three times, and the mean accuracy and standard deviation were reported. Table 4 summarizes the experimental and training configuration used for both models. The same leakage-free recording-level split, preprocessing pipeline, STFT

Table 4. Experimental and training configuration

Parameter	BiLSTM	CNN–BiLSTM
Software	MATLAB R2024b	MATLAB R2024b
Toolbox	Deep Learning Toolbox	Deep Learning Toolbox
Hardware	Single NVIDIA-based GPU and multi-core CPU	Single NVIDIA-based GPU and multi-core CPU
Dataset split	70% training, 15% validation, 15% testing	70% training, 15% validation, 15% testing
Split level	Original recording level	Original recording level
Segment duration	2 seconds	2 seconds
Segment overlap	50%	50%
Sampling frequency	2000 Hz	2000 Hz
Band-pass range	20–500 Hz	20–500 Hz
STFT window	25 ms Hamming window	25 ms Hamming window
STFT overlap	50%	50%
FFT size	512 points	512 points
Input size	128 × 128	128 × 128 × 1

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

Algorithm 1. Proposed leakage-free STFT-based PCG classification workflow

Input: Original PCG recordings and their class labels

Output: Predicted PCG class and evaluation metrics

1. Loading the original PCG recordings from the Yaseen dataset.
2. Assigning each original recording to one of five classes: AS, MR, MS, MVP, or N.
3. Splitting the original recordings into training, validation, and testing subsets using a 70:15:15 ratio.
4. Ensuring that all segments from the same original recording remain in the same subset.
5. For each subset separately:
   a. Converting recordings to mono format when required.
   b. b. Resampling the PCG signals to 2000 Hz.
   c. Applying band-pass filtering in the range of 20–500 Hz.
   d. Normalizing the signal amplitude.
   e. Segmenting each recording into 2-second windows with 50% overlap.
   f. Applying zero-padding for short recordings when required.
   g. Generating STFT spectrograms using a 25 ms Hamming window, 50% overlap, and 512-point FFT.
   h. Resizing each spectrogram to 128 × 128 pixels.
6. Training the BiLSTM model using the training spectrograms and validate it using the validation set.
7. Training the CNN–BiLSTM model using the same training and validation subsets.
8. Predicting the class labels of the independent test set using each trained model.
9. Computing accuracy, precision, recall, F1-score, specificity, and confusion matrix.
10. Repeatig the experiment three times and report the mean accuracy and standard deviation.

settings, and evaluation protocol were used to ensure a fair comparison between the BiLSTM and CNN–BiLSTM model.

G. Proposed Algorithm

Algorithm 1 describes the main workflow of the proposed STFT-based PCG classification framework. The algorithm includes recording-level splitting, preprocessing, segmentation, STFT spectrogram generation, model training, prediction, and metric calculation.

H. Mathematical Formulation of the Proposed Pipeline

This section presents the mathematical formulation of the proposed STFT-based PCG classification pipeline. The formulation covers preprocessing, segmentation, STFT spectrogram generation, CNN feature extraction, BiLSTM temporal modeling, softmax classification, loss function, optimizer update, and evaluation metrics. Let $x[n]$ denote the discrete-time PCG signal, n denote the sample index, and fs denote the sampling frequency. The resampled PCG signal can be expressed in Eq. (2) [30]:

$$x_r[n] = x\left(\frac{n}{f_s}\right) \quad (2)$$

where $x_r[n]$ is the resampled PCG signal and f_s is the unified sampling frequency used in this study. In this work, all PCG recordings were resampled to 2000 Hz to ensure consistency across samples [28] [27]. To reduce irrelevant low- and high-frequency components, band-pass filtering was applied in Eq. (3):

$$x_f[n] = x_r[n] * h[n] \quad (3)$$

where $x_f[n]$ was the filtered PCG signal, $h[n]$ is the impulse response of the band-pass filter, and * denotes convolution. The band-pass range was set to 20–500 Hz to preserve the main heart sound frequency components [27][33][34]. Amplitude normalization was then applied to reduce amplitude variation among different recordings show in Eq. (4):

$$x_{norm}[n] = \frac{x_f[n]}{max|x_f[n]|+\varepsilon} \quad (4)$$

where $x_{norm}[n]$ is the normalized signal and ε is a small constant used to avoid division by zero [34]. The segment length in samples is defined in Eq. (5):

$$L = T_s * f_s \quad (5)$$

where L is the number of samples per segment and T_s is the segment duration. In this study, T_s= 2 s.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

The hop size for 50% overlap is given by Eq. (6):

$$H = L \times (1 - r) \quad (6)$$

where H is the hop size and r are the overlap ratio. For 50% overlap, r = 0.5. The m-th PCG segment is extracted in Eq. (7):

$$s_m[n] = x_{norm}[n + mH], 0 \le n < L \quad (7)$$

where $s_m[n]$ is the m-th segmented PCG frame and m is the segment index. Before STFT computation, each segment was multiplied by a Hamming window as show in Eq. (8):

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right), 0 \le n < L \quad (8)$$

where L is the window length [28][29]. The Short-Time Fourier Transform of the windowed PCG segment is computed in Eq. (9):

$$\mathrm{X(m,k)} = \sum_{n=0}^{N-1} \mathrm{x[n]w[n - mH]e^{-\frac{j2\pi kn}{N}}} \quad (9)$$

where $X(m,k)$ is the complex STFT coefficient, m is the time-frame index, k is the frequency-bin index, and N is the FFT size [28] [29]. The magnitude spectrogram was obtained in Eq. (10):

$$S(m,k) = |X(m,k)| \quad (10)$$

where $S(m,k)$represents the magnitude of the STFT spectrogram. To reduce the dynamic range of spectrogram values, log scaling was applied in Eq. (11):

$$S_{log}(m,k) = \log(1 + S(m,k)) \quad (11)$$

where $S_{log}(m,k)$ is the log-scaled spectrogram representation [22] [24]. To reduce the dynamic range of spectrogram values, log scaling was applied in Eq. (12):

$$S_N(m,k) = \frac{S_{log}(m,k) - \mu_s}{\sigma_s + \varepsilon} \quad (12)$$

where $\mu_s$and σ_s are the mean and standard deviation of the log-scaled spectrogram values, respectively [37]. The normalized spectrogram was resized to a fixed input size show in Eq. (13) [37]:

$$I = Resize(S_N, 128 \times 128) \quad (13)$$

where I is the resized STFT spectrogram used as the input to the deep learning models. For the CNN–BiLSTM model, the two-dimensional convolution operation is defined in Eq. (14) [38]:

$$F_l(i,j,c) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} K_{m,n}^{(l)} I_{i+m,j+n} + b^l \quad (14)$$

where F_l is the output feature map of layer l, K_l is the convolution kernel, b^l is the bias term, c is the output channel, and p, q, and d are kernel and channel indices [39].Batch normalization is applied in Eq. (15):

$$\mathrm{BN(F)} = \gamma\left(\frac{\mathrm{F} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}\right) + \beta \quad (15)$$

where μ_B and σ_B^2 are the mini-batch mean and variance, while γ and β are learnable scale and shift parameters [39]. The ReLU activation function is expressed in Eq. (16):

$$A = \max(0, BN(z)) \quad (16)$$

where A is the activated feature map [39]. Max-pooling reduces the spatial dimension of the feature map in Eq. (17):

$$\mathrm{p(i,j,c)} = \max_{(p,q)\in\Omega} \mathrm{A(i+p, j+q, c)} \quad (17)$$

where Ω denotes the pooling window, and $p(i,j,c)$ is the pooled feature map [39]. Global average pooling summarizes each feature map show in Eq. (18):

$$g_c = \frac{1}{H_f W_f}\sum_{i=1}^{H_f}\sum_{j=1}^{W_f} P(i,j,c) \quad (18)$$

whereg_c is the global average pooled value for channel c, and H and W are the height and width of the feature map [39]. For the BiLSTM part, the LSTM forget gate is calculated in Eq. (19):

$$(f_t) = \sigma\left((W_{hf} \times h_{t-1}) + (W_{xf} \times x_t) + B_f\right) \quad (19)$$

where: f_t represents the forget gate value,σ is the sigmoid function,W_{hf} and W_{xf} are weight matrices for the previous hidden state (h_{t-1})) and input (x_t), respectively and B_f Is the bias vector for the forget gate[30] [31]. The input gate is defined in Eq. (20):

$$(i_t) = \sigma\left((W_i x_t) + (U_i h_{t-1}) + b_i\right) \quad (20)$$

where $i_t$controls the amount of new information added to the memory cell [30] [31]. The candidate cell state is computed in Eq. (21):

$$c_t = \tanh((W_c\, x_t) + (U_c h_{t-1}) + b_i \quad (21)$$

The output gate was computed in Eq. (22):

$$O_t = \sigma\left((W_{oh} \times h_{t-1}) + (W_{ox} \times x_t) + B_o\right) \quad (22)$$

where O_t is the output gate [34]. The hidden state was then calculated in Eq. (23):

$$h_t = o_t \odot \tanh(c_t) \quad (23)$$

where $h_t$is the hidden state at time t [34] [35] The categorical cross-entropy loss is defined in Eq. (24):

$$L_{CE} = -\frac{1}{N_S}\sum_{i=1}^{N_S}\sum_{c=1}^{C} y_{i,c}\log(\widehat{y_{i,c}}) \quad (24)$$

where N_S is the number of samples, $y_{i,c}$ is the true class label, and $y_{i,c}$ is the predicted probability for class c [39]. The Adam optimizer updates each model parameter θ [40] as show in Eq. (25):

$$\theta_{t+1} = \theta_t - \frac{\alpha\,\hat{m}_t}{(\sqrt{\hat{v}_t} + \varepsilon)} \quad (25)$$

where α is the learning rate, $\hat{m}_t$and $\hat{v}_t$are the bias-corrected [40]. First and second moment estimates, and ε is a small constant for numerical stability. The predicted class label was obtained in Eq. (26):

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

$$\hat{y} = \arg max_c (\hat{y}_c) \quad (26)$$

where $\hat{y}$ is the final predicted class. The classification accuracy was calculated in Eq. (27):

$$\text{Accuracy} = \frac{(TP+TN)}{TP+TN+FP+FN} \quad (27)$$

Precision was calculated in Eq. (28):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (28)$$

Recall, also known as sensitivity, was calculated in Eq. (29):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (29)$$

The F1-score was calculated in Eq. (30):

$$F_1\text{-score} = \frac{2\times \text{sensitivity} \times \text{recall}}{\text{sensitivity} + \text{recall}} \quad (30)$$

Specificity was calculated in Eq. (31):

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (31)$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative predictions, respectively [40].

Table 5. BiLSTM performance metrics

Class	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
AS	93.75	100.00	96.77	98.25
MR	100.00	80.00	88.89	100.00
MS	86.21	92.59	89.29	96.58
MVP	82.76	88.89	85.71	95.73
N	100.00	100.00	100.00	100.00
Macro-average	92.54	92.30	92.13	98.11

III. Result

A. Time-Frequency Representation of the Signal

This section presents the training behavior and classification performance of the proposed BiLSTM and CNN–BiLSTM models. The results were obtained using the independent test set generated from the leakage-free recording-level split described in the Methodology section. Training and validation accuracy and loss curves were used to monitor convergence behavior, while confusion matrices and standard evaluation metrics were used to assess the final classification performance on the test set. In Fig. 3. Illustrates the spectrogram of the signal after the initial processing stage. The horizontal axis represents the number of time frames, while the vertical axis represents the frequency bins. The colours indicate the energy intensity at each frequency at a specific time point, with red representing the highest energy values and blue representing the lowest. The figure reveals regions of high energy at low frequencies during specific time intervals, indicating the presence of strong fundamental components in the signal. Clear changes in the spectral distribution over time are also evident, reflecting the unstable nature of the signal under study.

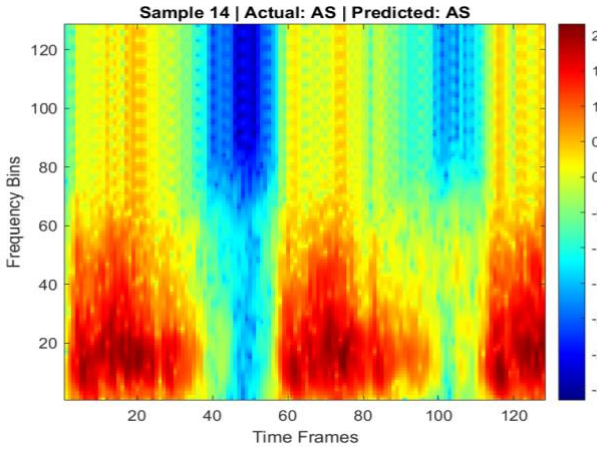


Fig. 3. STFT Spectrogram Representation.

B. Training Behavior of BiLSTM and CNN-BiLSTM Models

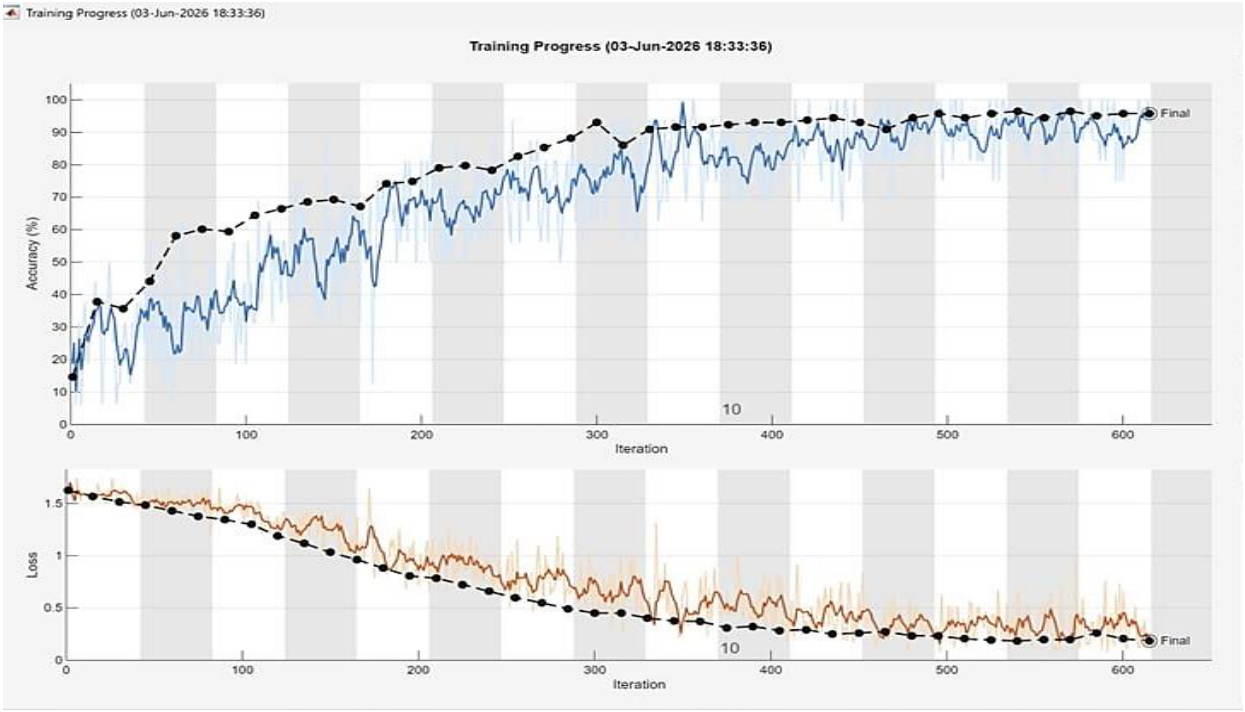
The BiLSTM model was trained for 40 epochs using the Adam optimizer, an initial learning rate of 0.001, and a mini-batch size of 16. As shown in Fig.4. (a) the training accuracy increased gradually during the early epochs and reached stable convergence after approximately 20 epochs. The validation curve remained close to the training curve, indicating stable learning behavior with limited overfitting. The CNN-BiLSTM model was also trained for 40 epochs using the Adam optimizer, a learning rate of 0.0005, and a mini-batch size of 16. As shown in Fig. 4(b), the hybrid model showed faster convergence and stable validation performance. This behavior can be attributed to the convolutional layers, which extract discriminative local time-frequency features before temporal modeling by the BiLSTM layer.

C. Performance Metrics

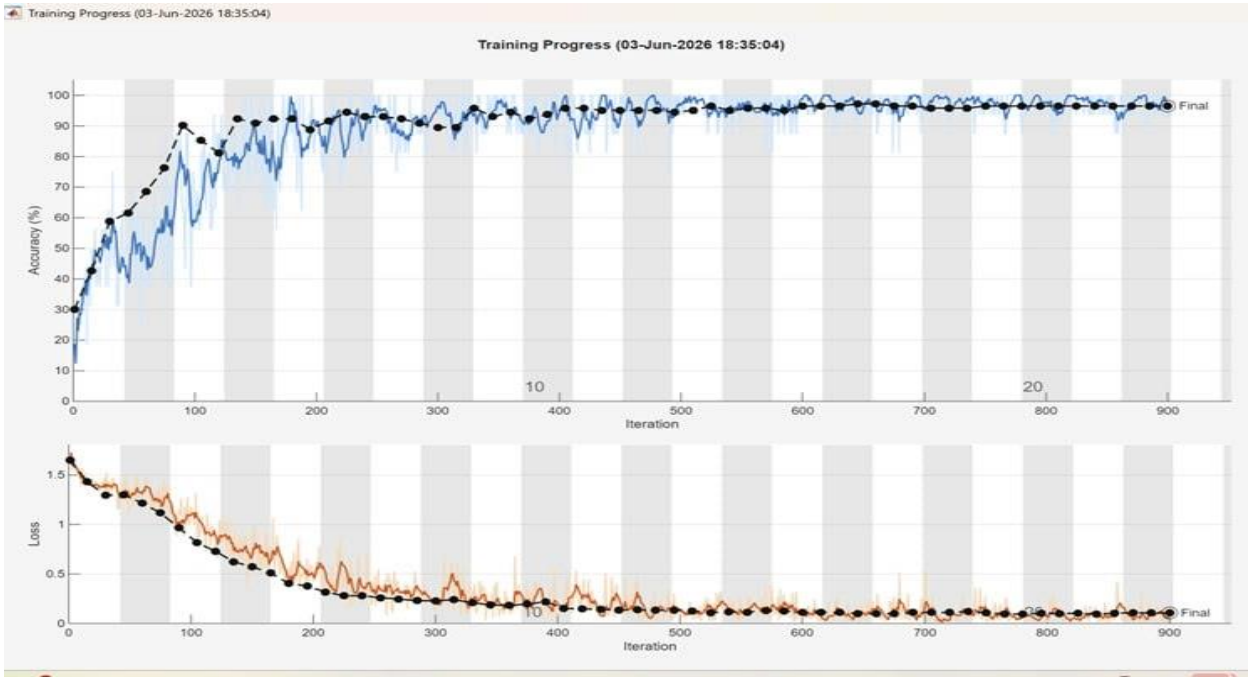
The classification performance of the proposed BiLSTM and CNN–BiLSTM models was evaluated using the independent test set generated from the leakage-free recording-level split. Standard evaluation metrics were used, including accuracy, precision, recall, F1-score, and specificity. These metrics were computed for the five PCG classes: Aortic Stenosis (AS), Mitral Regurgitation (MR), Mitral Stenosis (MS), Mitral Valve Prolapse (MVP), and Normal (N). Confusion matrices were also used to provide a detailed visual analysis of the prediction behavior for each model. Fig. 5. (a) illustrates the confusion matrix

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

of the BiLSTM model. The model correctly classified most samples across the five classes and achieved perfect classification for the Normal class. However, some misclassifications were observed among pathological classes, especially MR, MS, and MVP. This indicates that some abnormal PCG classes have overlapping acoustic characteristics in the STFT spectrogram representation. Fig.5. (b) illustrates the



(a)



(b)

Fig. 4. Training progress of the proposed models: (a) BiLSTM and (b) CNN– BiLSTM.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

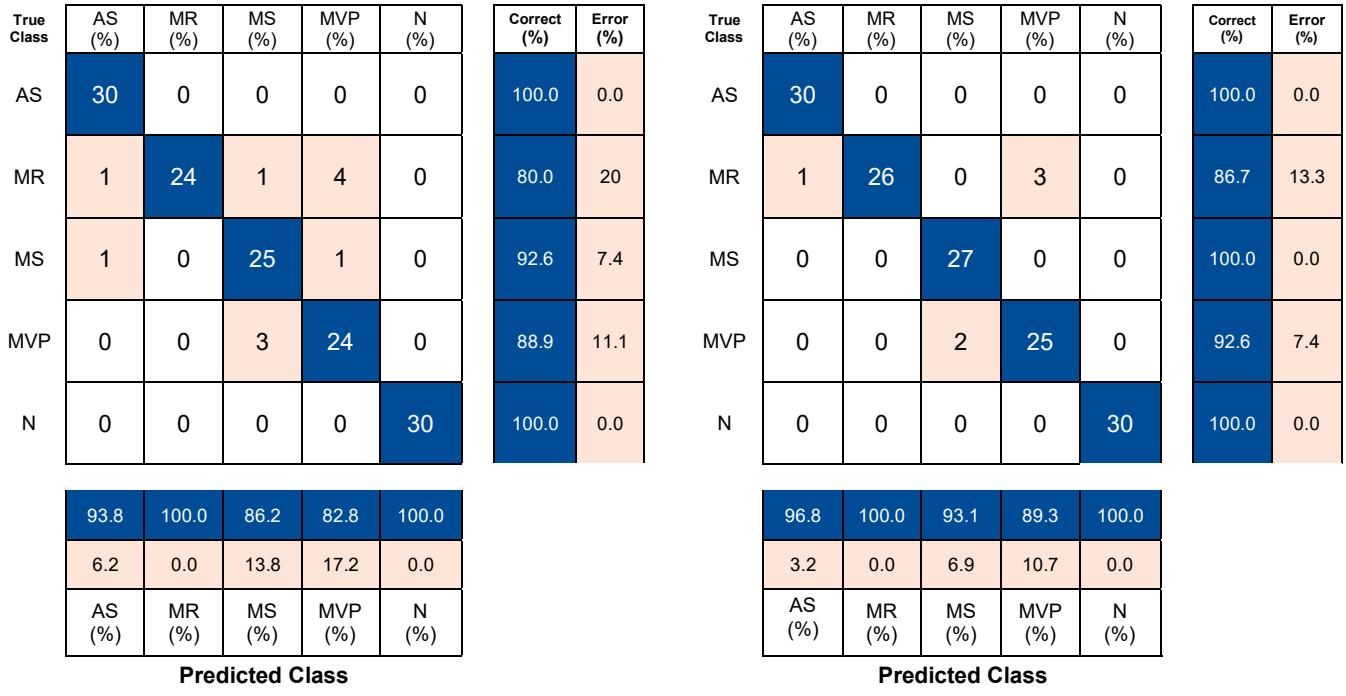


Fig. 5. Confusion matrices of: (a) BiLSTM and(b) CNN-BiLSTM .

Table 6. CNN–BiLSTM performance metrics.

Class	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
AS	96.77	100.00	98.36	99.12
MR	100.00	86.67	92.86	100.00
MS	93.10	100.00	96.43	98.29
MVP	89.29	92.59	90.91	97.44
N	100.00	100.00	100.00	100.00

confusion matrix of the CNN–BiLSTM model. Compared with the BiLSTM model, the CNN–BiLSTM model produced fewer misclassifications and showed better separation among the pathological classes. This improvement can be attributed to the convolutional layers, which extract local time–frequency features from the STFT spectrograms before temporal modeling by the BiLSTM layer. Table 5 illustrates the class-wise performance of the BiLSTM model using the final leakage-free representative run. The model achieved an overall test accuracy of 92.36%, corresponding to an error rate of 7.64%. The macro-averaged precision, recall, F1-score, and specificity were 92.54%, 92.30%, 92.13%, and 98.11%, respectively. The Normal class achieved the best performance, with 100% precision, recall, F1-score, and specificity. The AS class also showed strong performance, with 93.75% precision, 100% recall, and 96.77% F1-score. The weakest recall was observed for the MR class at 80.00%, while the lowest precision and F1-score were observed for the MVP class at 82.76% and 85.71%, respectively. These results suggest that the BiLSTM model can effectively identify normal PCG signals, but some pathological classes, particularly MR and MVP, remain more difficult to distinguish. Table 6 illustrates the class-wise performance of the CNN–BiLSTM model using the final leakage-free representative run. The model achieved an overall test accuracy of 95.83%, corresponding to an error rate of 4.17%. The macro-averaged precision, recall, F1-score, and specificity were 95.83%, 95.85%, 95.71%, and 98.97%, respectively. The Normal class again achieved perfect classification performance, with 100% precision, recall, F1-score, and specificity. The AS and MS classes achieved 100% recall, indicating that all test samples from these classes were correctly detected. The weakest F1-score was observed for the MVP class at 90.91%, while the lowest recall was observed for the MR class at 86.67%. These results indicate that the CNN–BiLSTM model provided more balanced classification performance than the standalone BiLSTM model, although some confusion remained among acoustically similar pathological classes. Based on Tables 5 and Table 6, the CNN–

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

BiLSTM model outperformed the BiLSTM model across all major evaluation metrics. The overall test accuracy improved from 92.36% for BiLSTM to 95.83% for CNN–BiLSTM, representing an improvement of 3.47 percentage points. The macro F1-score also increased from 92.13% to 95.71%, while macro specificity increased from 98.11% to 98.97%. These findings show that adding convolutional feature extraction before BiLSTM temporal modeling improved the classification of STFT-based PCG spectrograms. Fig. 6. compares the overall and class-wise performance of the BiLSTM and CNN–BiLSTM models. The CNN–BiLSTM model achieved higher overall accuracy and better macro-averaged performance, while the BiLSTM model remained a simpler and computationally lighter alternative. Therefore, CNN–BiLSTM is more suitable when classification accuracy and stability are the main priorities, whereas BiLSTM may still be useful for lightweight implementation. To examine model stability, the experiments were repeated three times using the same leakage-free recording-level evaluation protocol. The BiLSTM model achieved an average accuracy of 93.85% ± 1.47%, whereas the CNN–BiLSTM model achieved 95.94% ± 0.71%. The lower standard deviation of the CNN–BiLSTM model indicates more stable performance across repeated runs. Therefore, the CNN–BiLSTM model emonstrated both higher accuracy and better consistency compared with the standalone BiLSTM model.

IV. Discussion

This study evaluated the effectiveness of BiLSTM and CNN–BiLSTM models for multiclass heart sound classification using STFT-based spectrogram representations. The experiments were performed using a leakage-free recording-level splitting strategy, where the original PCG recordings were divided into training, validation, and testing subsets before segmentation and spectrogram extraction. This strategy ensured that segments generated from the same recording were not shared across different subsets. In the final representative run, the CNN–BiLSTM model achieved the highest test accuracy of 95.83%, while the BiLSTM model achieved 92.36%. These findings indicate that the hybrid CNN–BiLSTM architecture provided better classification performance for the five PCG classes, namely AS, MR, MS, MVP, and Normal. The improved performance of CNN–BiLSTM can be attributed to its ability to learn both spatial and temporal characteristics from the STFT spectrograms. The convolutional layers extract local time–frequency patterns from the spectrogram images, while the BiLSTM layer models the sequential dependencies across time frames. In contrast, the standalone BiLSTM model mainly focuses on temporal dependencies extracted from the spectrogram sequence. Although the BiLSTM model achieved lower accuracy than the hybrid architecture, it still provided acceptable performance with a simpler structure and lower computational complexity. The repeated experiments further support the stability of the CNN–

Table 7. Comparative analysis with previous studies

Study	Dataset, classes, and validation protocol	Accuracy and main limitation
Chen et al. [9] (2023)	Multiple heart-sound datasets; 2 classes; cross-dataset evaluation	91.74 ± 3.72%; binary task and variable test performance
Orozco-Reyes et al. [8] (2025)	PhysioNet/CinC, CirCor, and open PCG database; 2 classes; 10-fold cross-validation	Up to 99.99%; binary task with combined time-frequency representations
Barnawi et al. [18] (2023)	PhysioNet; 2 classes; data selection-based evaluation	97.00%; binary normal-abnormal classification
Harimi et al. [16] (2023)	Public PCG datasets; 2 classes; reported in study	94.30%; binary task and transfer learning complexity
Khan et al. [24] (2021)	PhysioNet/CinC and PASCAL; 2 classes; cross-dataset / transfer learning	95.75%; binary normal-abnormal classification
Proposed BiLSTM (2026)	Yaseen PCG; 5 classes; leakage-free recording-level split	92.36%; single-dataset evaluation
Proposed CNN-BiLSTM (2026)	Yaseen PCG; 5 classes; leakage-free recording-level split	95.83%; single-dataset evaluation
Proposed CNN-BiLSTM (2026)	Yaseen PCG; 5 classes; three repeated runs	95.94 ± 0.71%; external validation required

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

BiLSTM model. Across three leakage-free runs, the BiLSTM model achieved an average accuracy of 93.85% ± 1.47%, whereas the CNN–BiLSTM model achieved 95.94% ± 0.71%. The lower standard deviation of CNN–BiLSTM indicates that the hybrid model produced more consistent results across different training runs. This suggests that combining convolutional feature extraction with bidirectional temporal modeling improves both classification accuracy and training stability.

The class-wise results also provide important insight into the behavior of the proposed models. For the BiLSTM model, the Normal class achieved perfect performance with 100% precision, recall, F1-score, and specificity. This indicates that normal heart sounds were clearly distinguishable from abnormal cases in the extracted STFT representations. However, the model showed lower performance for some pathological classes, particularly MR and MVP. The MR class had lower recall, while the MVP class showed lower precision and F1-score, suggesting that these classes may share acoustic characteristics with other abnormal heart sound categories. For the CNN–BiLSTM model, the class-wise performance improved across most categories. The model achieved 100% recall for AS, MS, and Normal classes, while maintaining high precision and specificity. The Normal class again achieved perfect classification performance, confirming the separability of normal PCG recordings from pathological recordings. Some remaining errors were still observed in MR and MVP classes, but their performance values were improved compared with the BiLSTM model. This behavior suggests that the convolutional layers helped the model extract more discriminative local patterns from the spectrograms.

The misclassification patterns observed in MR and MVP can be clinically interpreted by considering the acoustic similarity among valvular heart diseases. MR and MVP may generate murmur-like patterns that overlap in both time and frequency domains. Since STFT represents heart sounds as localized time–frequency components, overlapping spectral patterns may still make some abnormal classes difficult to distinguish. Therefore, the remaining errors are not necessarily random, but may be related to shared acoustic properties among pathological heart sound classes. The learning behavior of both models showed stable convergence during training. The CNN–BiLSTM model achieved higher validation and test performance, indicating that the additional convolutional feature extraction stage improved the representation of PCG spectrograms before temporal modeling. The BiLSTM model, despite its lower accuracy, remained useful as a lighter model because it requires fewer processing stages than the hybrid architecture. This trade-off between accuracy and computational complexity is important when selecting a model for practical implementation.

Compared with Chen et al. [9] ,the present study is similar in using time–frequency representations and deep learning models for heart sound classification. However, Chen et al. [9] evaluated a binary classification task using multiple heart-sound datasets and reported 91.74 ± 3.72% accuracy, whereas the present study addressed a five-class PCG classification task involving AS, MR, MS, MVP, and Normal classes using a leakage-free recording-level split.

Orozco-Reyes et [8] .also used PCG datasets and combined time–frequency representations for automatic heart sound classification. The similarity with the present study is that both works rely on time–frequency information to improve deep learning-based PCG classification. The main difference is that Orozco-Reyes et al. [8] focused on a binary classification task using PhysioNet/CinC and open PCG databases with 10-fold cross-validation and reported accuracy up to 99.99%, while the present study focused on five-class classification using the Yaseen PCG dataset and applied a leakage-free recording-level split before segmentation.

Barnawi et al. [18] proposed a deep learning-based PCG classification method using the PhysioNet dataset. Their work is similar to the present study because both studies aimed to improve automated heart sound classification using deep learning. However, Barnawi et al. [18] focused on binary normal–abnormal classification and data selection-based evaluation, while the present study classified five specific PCG classes and compared BiLSTM and CNN–BiLSTM models under the same preprocessing and evaluation protocol.

Harimi et al. [16] used public PCG datasets and transfer learning for heart sound classification. The similarity with the present study is the use of deep learning for automated PCG analysis. However, Harimi et al. [16] focused on a binary classification task and involved transfer learning complexity, whereas the present study used a simpler STFT-based BiLSTM/CNN–BiLSTM framework for five-class classification with a recording-level leakage-free evaluation strategy.

Khan et al. [24],used PhysioNet/CinC and PASCAL datasets with PCG spectrograms and transfer learning. Their study is similar to the present work because both studies confirm the usefulness of spectrogram-based representations for PCG classification. The main difference is that Khan et al. [24] mainly addressed binary normal–abnormal classification and cross-

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

dataset/transfer learning evaluation, while the present study addressed a more challenging five-class classification task using STFT spectrograms and a leakage-free recording-level split.

As summarized in Table 7, the proposed models were compared with selected previous PCG classification studies in terms of dataset, number of classes, validation protocol, accuracy, and main limitation. The comparison shows that previous studies achieved promising results using deep learning and time–frequency or spectrogram-based representations. However, most of them focused on binary normal–abnormal classification or used different datasets and validation strategies. In contrast, the present study addresses a five-class PCG classification task involving AS, MR, MS, MVP, and Normal classes. In addition, the proposed framework applies a leakage-free recording-level split before segmentation and STFT spectrogram extraction. Therefore, although some previous studies reported higher accuracy, the proposed CNN–BiLSTM model achieved competitive performance under a stricter and more reproducible evaluation protocol.

V. Conclusion

This study developed and evaluated an STFT-based deep learning framework for multiclass PCG classification using BiLSTM and CNN–BiLSTM models. The aim was to classify five heart sound categories, namely AS, MR, MS, MVP, and Normal, while applying a leakage-free recording-level splitting strategy to reduce the risk of overestimated performance. The experimental results showed that the CNN–BiLSTM model achieved the best overall performance. In the final representative leakage-free run, CNN–BiLSTM achieved 95.83% test accuracy, compared with 92.36% for the BiLSTM model. The CNN–BiLSTM model also achieved higher macro-averaged precision, recall, F1-score, and specificity, with values of 95.83%, 95.85%, 95.71%, and 98.97%, respectively. Across three repeated runs, CNN–BiLSTM achieved 95.94% ± 0.71%, while BiLSTM achieved 93.85% ± 1.47%. These results confirm that CNN–BiLSTM provided higher and more stable classification performance, whereas BiLSTM remained a simpler and computationally lighter alternative. The findings suggest that combining convolutional feature extraction with bidirectional temporal modeling can improve the classification of STFT-based heart sound spectrograms. The proposed framework may be useful as a supportive tool for digital stethoscope screening and preliminary PCG analysis. However, clinical claims should be made cautiously because the evaluation was performed on a single public dataset. External validation using independent datasets such as PhysioNet/CinC and PASCAL is required before considering the model for clinical decision-support use.

Future work will focus on external validation, testing under different recording conditions, evaluating computational complexity and inference time, and implementing the proposed framework on embedded or portable digital stethoscope platforms.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Electrical Engineering, University of Babylon, Iraq, for the academic support and research environment provided during the preparation of this study.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

Data Availability

The dataset used in this study is a publicly available phonocardiogram dataset reported by Yaseen et al. [25]. The dataset and related files are available at: https://github.com/yaseen21khan/Classification-of-Heart-Sound-Signal-Using-Multiple-Features-. No new dataset was generated during this study.

Author Contribution

Noor S. Ali contributed to the study design, data preprocessing, STFT spectrogram generation, model implementation, experiments, result analysis, and manuscript writing. Ehab Abdulrazzaq Hussein contributed to supervision, methodology refinement, result interpretation, and manuscript review. Laith Ali Abdul-Rahaim contributed to supervision, technical guidance, result validation, and manuscript review. All authors reviewed and approved the final version of the manuscript.

Declarations

Ethical Approval

This study used a publicly available phonocardiogram dataset. No new human participants were recruited, and no new clinical data were collected by the authors. Therefore, additional ethical approval was not required for this secondary data analysis.

Consent for Publication
Not applicable.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

Competing Interests
The authors declare no competing interests.

Code Availability
The code used in this study is available from the corresponding author upon reasonable request.

References

[1] Y. Zheng, X. Guo, Y. Yang, H. Wang, K. Liao, and J. Qin, "Phonocardiogram transfer learning-based CatBoost model for diastolic dysfunction identification using multiple domain-specific deep feature fusion," *Comput. Biol. Med.*, vol. 156, 2023, doi: 10.1016/j.compbiomed.2023.106707.

[2] E. Partovi, A. Babic, and A. Gharehbaghi, "A review on deep learning methods for heart sound signal analysis," 2024. doi: 10.3389/frai.2024.1434022.

[3] M. Kalimuthu and C. Hemanth, "Preliminary Study on Real-Time Phonocardiogram Signal Acquisition and Analysis Using Machine Learning and IoMT for Digital Stethoscope," *IEEE Access*, vol. 13, 2025, doi: 10.1109/ACCESS.2025.3560763.

[4] S. Ismail, B. Ismail, I. Siddiqi, and U. Akram, "PCG classification through spectrogram using transfer learning," *Biomed. Signal Process. Control*, vol. 79, 2022, doi: 10.1016/j.bspc.2022.104075.

[5] P. Narváez and W. S. Percybrooks, "Synthesis of normal heart sounds using generative adversarial networks and empirical wavelet transform," *Applied Sciences (Switzerland)*, vol. 10, no. 19, 2020, doi: 10.3390/app10197003.

[6] C. Liu *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiol. Meas.*, vol. 37, no. 12, pp. 2181–2213, Nov. 2016, doi: 10.1088/0967-3334/37/12/2181.

[7] T. H. Chowdhury, K. N. Poudel, and Y. Hu, "Time-Frequency Analysis, Denoising, Compression, Segmentation, and Classification of PCG Signals," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3020806.

[8] L. Orozco-Reyes, M. A. Alonso-Arévalo, E. García-Canseco, R. F. Ibarra-Hernández, and R. Conte-Galván, "A Deep-Learning Approach to Heart Sound Classification Based on Combined Time-Frequency Representations," *Technologies (Basel).*, vol. 13, no. 4, 2025, doi: 10.3390/technologies13040147.

[9] W. Chen *et al.*, "Classifying Heart-Sound Signals Based on CNN Trained on MelSpectrum and Log-MelSpectrum Features," *Bioengineering*, vol. 10, no. 6, 2023, doi: 10.3390/bioengineering10060645.

[10] F. Li, H. Tang, S. Shang, K. Mathiak, and F. Cong, "Classification of heart sounds using convolutional neural network," *Applied Sciences (Switzerland)*, vol. 10, no. 11, 2020, doi: 10.3390/app10113956.

[11] F. Noman, S. H. Salleh, C. M. Ting, S. B. Samdin, H. Ombao, and H. Hussain, "A Markov-Switching Model Approach to Heart Sound Segmentation and Classification," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 3, 2020, doi: 10.1109/JBHI.2019.2925036.

[12] T. Jat, P. Bhat, and N. Patil, "Detection of Heart Abnormality with Stethoscope Sounds," *SN Comput. Sci.*, vol. 6, no. 6, 2025, doi: 10.1007/s42979-025-04235-3.

[13] S. Tiwari, A. Jain, A. K. Sharma, and K. Mohamad Almustafa, "Phonocardiogram signal based multi-class cardiac diagnostic decision support system," IEEE Access, vol. 9, pp. 110710–110722, 2021, doi: 10.1109/ACCESS.2021.3103316.

[14] G. Singh, A. Verma, L. Gupta, A. Mehta, and V. Arora, "An automated diagnosis model for classifying cardiac abnormality utilizing deep neural networks," *Multimed. Tools Appl.*, vol. 83, no. 13, 2024, doi: 10.1007/s11042-023-16930-5.

[15] M. Morshed and S. A. Fattah, "A Deep Neural Network for Heart Valve Defect Classification From Synchronously Recorded ECG and PCG," *IEEE Sens. Lett.*, vol. 7, no. 9, 2023, doi: 10.1109/LSENS.2023.3307053.

[16] A. Harimi, M. A. Ameri, S. Sarkar, and M. W. Totaro, "Heart sounds classification: Application of a new CyTex inspired method and deep convolutional neural network with transfer learning," *Smart Health*, vol. 29, 2023, doi: 10.1016/j.smhl.2023.100416.

[17] M. S. Khan, F. A. Khan, K. N. Khan, S. I. Rana, and M. A. A. A. Al-Hashemi, "Advanced Deep Learning for Heart Sounds Classification," in *Studies in Computational Intelligence*, vol. 1124, 2023. doi: 10.1007/978-3-031-46341-9_9.

[18] A. Barnawi, M. Boulares, and R. Somai, "Simple and Powerful PCG Classification Method Based on Selection and Transfer Learning for Precision Medicine Application," *Bioengineering*, vol. 10, no. 3, 2023, doi: 10.3390/bioengineering10030294.

[19] S. A. Singh, N. D. Devi, K. N. Singh, K. Thongam, B. R. D, and S. Majumder, "An ensemble-based transfer learning model for predicting the imbalance heart sound signal using spectrogram

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

images," *Multimed. Tools Appl.*, vol. 83, no. 13, 2024, doi: 10.1007/s11042-023-17186-9.

[20] H. K. Alkahtani, I. U. Haq, Y. Y. Ghadi, N. Innab, M. Alajmi, and M. Nurbapa, "Precision Diagnosis: An Automated Method for Detecting Congenital Heart Diseases in Children From Phonocardiogram Signals Employing Deep Neural Network," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3395389.

[21] J. S. Khan, M. Kaushik, A. Chaurasia, M. K. Dutta, and R. Burget, "Cardi-Net: A deep neural network for classification of cardiac disease using phonocardiogram signal," *Comput. Methods Programs Biomed.*, vol. 219, 2022, doi: 10.1016/j.cmpb.2022.106727.

[22] Z. Tariq, S. K. Shah, and Y. Lee, "Feature-Based Fusion Using CNN for Lung and Heart Sound Classification†," *Sensors*, vol. 22, no. 4, 2022, doi: 10.3390/s22041521.

[23] Y. Al-Issa and A. M. Alqudah, "A lightweight hybrid deep learning system for cardiac valvular disease classification," *Sci. Rep.*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-18293-7.

[24] K. N. Khan *et al.*, "Deep learning based classification of unsegmented phonocardiogram spectrograms leveraging transfer learning," *Physiol. Meas.*, vol. 42, no. 9, 2021, doi: 10.1088/1361-6579/ac1d59.

[25] Yaseen, G.-Y. Son, and S. Kwon, "Classification of heart sound signal using multiple features," Applied Sciences, vol. 8, no. 12, Art. no. 2344, 2018: 10.3390/app8122344.

[26] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, 1977, doi: 10.1109/PROC.1977.10770.

[27] N. Ream, "Discrete-Time Signal Processing," *Electronics and Power*, vol. 23, no. 2, 1977, doi: 10.1049/ep.1977.0078.

[28] M. T. Nguyen, W. W. Lin, and J. H. Huang, "Heart Sound Classification Using Deep Learning Techniques Based on Log-mel Spectrogram," *Circuits Syst. Signal Process.*, vol. 42, no. 1, 2023, doi: 10.1007/s00034-022-02124-1.

[29] I. Ait Ichou, S. Elouaham, B. Nassiri, and J. Isknan, "Deep multimodal learning for heart sound classification using CNN, Transformer, and BiLSTM with attention," Symmetry, vol. 18, no. 4, Art. no. 556, 2026, doi: 10.3390/sym18040556.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997,doi:.org/10.1162/neco.1997.9.8.1735.

[31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5–6, pp. 602–610, 2005, doi: 10.1016/j.neunet.2005.06.042.

[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[33] H. Jiang, S. A. Salehi, M. D. Riedel, and K. K. Parhi, "Discrete-time signal processing with DNA," *ACS Synth. Biol.*, vol. 2, no. 5, 2013, doi: 10.1021/sb300087n.

[34] A. D. Poularikas and Z. M. Ramadan, "Discrete-time signal processing," in *Adaptive Filtering Primer With Matlab®*, 2019. doi: 10.1201/9781315221946-2.

[35] S. Liang, Y. Khoo, and H. Yang, "Drop-Activation: Implicit Parameter Reduction and Harmonious Regularization," *Communications on Applied Mathematics and Computation*, vol. 3, no. 2, 2021, doi: 10.1007/s42967-020-00085-3.

[36] B. Althaph and N. P. Challa, "Explainable attention-based deep learning for classification and interpretation of heart murmurs using phonocardiograms," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-21971-x.

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd International Conference on Machine Learning, ICML, 2015, pp. 448–456. arXiv:1502.03167

[39] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. https://doi.org/10.48550/arXiv.1412.6980.

[40] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009,doi: 10.1016/j.ipm.2009.03.002.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724

Author Biography

Noor S. Ali is with the Department of Electrical Engineering, University of Babylon, Hilla, Iraq. Her research interests include biomedical signal processing, heart sound analysis, machine learning, deep learning, and intelligent medical diagnosis systems. She has been involved in research related to automated phonocardiogram (PCG) classification using time–frequency representations and deep neural network architectures. Her recent work focuses on the development of robust computer-aided diagnostic models based on Short-Time Fourier Transform (STFT) spectrograms, Bidirectional Long Short-Term Memory (BiLSTM) networks, and hybrid CNN–BiLSTM architectures for multiclass heart sound classification. Her academic interests also include pattern recognition, medical data analysis, and the application of artificial intelligence techniques in healthcare and biomedical engineering. Email: eng.noor.salman@uobabylon.edu.iq Orcid: https://orcid.org/0009-0007-6627-8872

Ehab Abdulrazzaq Hussein received the B.Sc. degree in Electrical Engineering from the Faculty of Engineering, University of Babylon, Iraq, in 1997, the M.Sc. degree in Electrical Engineering from the University of Technology, Iraq, in 2000, and the Ph.D. degree in Electrical Engineering from the University of Basrah, Iraq, in 2007. He is currently a Professor in the Department of Electrical Engineering, Faculty of Engineering, University of Babylon, Iraq. His research interests include signal processing, security, information transmission, sensors, and control system analysis.Email: dr.ehab@uobabylon.edu.iq Google Scholar:https://scholar.google.com/citations?view_op=list_works&hl=en&user=r4E_rWMAAAAJ ResearchGate: Scopus: https://www.scopus.com/authid/detail.uri?authorId=57203957823.

Laith Ali Abdul-Rahaim Professor Laith Ali Abdul-Rahaim (Member IEEE) was born in Babylon-1972, Iraq. He received the B.Sc. degree in Electronics and Communications Department from the University of Baghdad (1995)-Iraq, M.Sc. and Ph.D. degrees in Electronics and Communication Engineering from the University of Technology-Iraq in 2001 and 2007 respectively. Since 2003, he has been with the University of Babylon-Iraq, His current position is Dean of faculty of engineering at university of Babylon. His research interests include MC-CDMA, OFDM, MIMO-OFDM, CDMA, Space Time Coding, Modulation Technique, Image processing, Communication Design, Information Science, Computer Security and Reliability Computer Communications (Networks)Communication Engineering, Electronic Engineering, Instrumentation Engineering. His email drlaithanzy@uobabylon.edu.iq, Orcid: https://orcid.org/ 0000-0001-8064-4401.

Manuscript received 6 Mei 2026; Revised 25 June 2026; Accepted 20 July 2026; Available online 3 August 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i4.1724