DEEP MURMUR: AI DRIVEN PHONOCARDIOGRAM ANALYSIS FOR HEART MURMUR DETECTION

A PROJECT REPORT

Submitted by

SANTHOSH S (2022503042)

RAMANATH PRABHU V B (2022503550)

SURESH KUMAR K (2022503708)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

COURSE CODE: CS6611

COURSE TITLE: CREATIVE AND INNOVATIVE PROJECT

DEPARTMENT OF COMPUTER TECHNOLOGY

ANNA UNIVERSITY, MIT CAMPUS

CHENNAI – 600044

APRIL 2025

DEPARTMENT OF COMPUTER TECHNOLOGY

ANNA UNIVERSITY, MIT CAMPUS

CHROMPET, CHENNAI – 600044

BONAFIDE CERTIFICATE

Certified that this project report "**Deep Murmur: AI driven phonocardiogram analysis for heart murmur detection**" is the work of **Mr. Santhosh S (2022503042), Mr. Ramanath Prabhu V B (2022503550), Mr. Suresh Kumar K (2022503708)** in the Creative and Innovative Project Laboratory subject code CS6611 during the period January to April 2025.

SIGNATURE

Dr. Y. Nancy Jane

SUPERVISOR

Assistant Professor

Department of Computer Technology

Anna University, MIT Campus

Chromepet – 600044.

SIGNATURE

Dr. P. Jayashree

HEAD OF THE DEPARTMENT

Professor

Department of Computer Technology

Anna University, MIT Campus

Chromepet – 600044

ABSTRACT

Accurate detection of heart murmurs is crucial for early diagnosis of cardiovascular diseases and effective treatment planning. Traditional methods for heart murmur detection, such as manual auscultation, are often limited by operator experience and subjectivity, and face challenges due to noise interference and variability in phonocardiogram (PCG) signals. These issues become even more pronounced in real-time monitoring systems, especially those using portable and wearable devices with limited sensor data.

This study proposes a hybrid deep learning framework to address these challenges and improve the accuracy of heart murmur classification from PCG signals. The framework integrates three distinct models—Multi-Head Self-Attention Transformers, PatchTST, and ResNet-18—as base learners, with XGBoost utilized as a meta-learner to combine the predictions of the individual models and achieve enhanced performance. By leveraging ensemble learning, the proposed model captures diverse features from the PCG signals, improving both accuracy and generalization.

The model was trained and evaluated on the Circor Digiscope dataset, achieving a classification accuracy of 96.5%. This demonstrates the model's ability to accurately classify heart murmurs as present or absent, even in the presence of noise and signal variability. The results underscore the potential of this deep learning framework for deployment in real-time clinical settings, such as wearable health monitoring systems. By providing a highly accurate and interpretable solution, this work contributes to the advancement of automated, non-invasive heart murmur detection.

ACKNOWLEDGEMENT

We express our heartfelt gratitude to Dr. K. Ravichandran, Dean, MIT Campus, and Dr. P. Jayashree, Professor & Head, Department of Computer Technology, MIT Campus, for providing the necessary facilities and support.

We are deeply thankful to our project guide, **Dr. Y. Nancy Jane**, Assistant Professor, Department of Computer Technology, MIT, Anna University, whose expert guidance, and constant encouragement helped us realize the successful implementation of our project.

We also extend our thanks to the project panel members—**Dr. Ponsy R K Sathia Bhama**, Associate Professor, Department of Computer Technology, MIT, Anna University, **Dr. P. Pabitha**, Associate Professor, Department of Computer Technology, MIT, Anna University, **Dr. S. Muthurajkumar**, Associate Professor, Department of Computer Technology, MIT, Anna University, **Dr. R. Kathiroli**, Assistant Professor, Department of Computer Technology, MIT, Anna University, and **Dr. T. Sudhakar**, Assistant Professor, Department of Computer Technology, MIT, Anna University, for their insightful feedback and constructive reviews.

We thank all the teaching and non-teaching members of the Department of Computer Technology, MIT, Anna University whose appreciable help, either directly or indirectly helped in the smooth completion of the Creative and Innovative Project in a limited time frame.

Santhosh S (2022503042)

Ramanath Prabhu V B (2022503550)

Suresh Kumar K (2022503708)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

PCG	Phonocardiogram
MFCC	Mel-Frequency Cepstral Coefficients
STFT	Short-Time Fourier Transform
DWT	Discrete Wavelet Transform
SWT	Stationary Wavelet Transform
FFT	Fast Fourier Transform
ResNet	Residual Network
XGBoost	Extreme Gradient Boosting
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Heart murmurs are abnormal sounds heard during heartbeat cycles, often associated with cardiovascular diseases (CVD). These murmurs can indicate serious heart conditions such as valve disorders, congenital defects, and other heart-related complications. Traditional methods for heart murmur detection largely rely on manual auscultation by healthcare professionals, using stethoscopes to listen for these abnormal sounds. However, manual detection is often prone to human error and depends heavily on the clinician's experience, leading to a risk of misdiagnosis.

In recent years, there has been a growing interest in automating the process of heart murmur detection using machine learning and signal processing techniques. Phonocardiogram (PCG) signals, which record heart sounds, are commonly used for this task. The signals contain valuable temporal and frequency-domain features that can help identify the presence or absence of heart murmurs. Despite their usefulness, PCG signals are often subject to noise, interference, and variations between individuals, making classification a challenging task. Thus, there is a need for more robust, accurate, and efficient methods for heart murmur detection, particularly in real-time clinical settings.

1.2 PROBLEM STATEMENT

The conventional diagnostic approach of manually analysing PCG signals using stethoscopes or traditional algorithms faces several limitations, such as operator dependency, subjectivity, and noise sensitivity. Thus, there is a pressing need for automated and reliable heart murmur detection systems that can be deployed in real-time monitoring devices for better diagnosis and patient outcomes.

This research aims to address these limitations by developing a robust deep learning framework for the accurate classification of heart murmurs from PCG signals, focusing on combining multiple advanced machine learning models to improve both accuracy and generalizability.

1.3 MOTIVATION

Advancements in machine learning, particularly in deep learning, have shown great promise in improving the accuracy of medical diagnostics. This study is motivated by the potential to apply these techniques to heart murmur detection, particularly through the use of innovative deep learning models such as Transformers, ResNet, and XGBoost. By leveraging the strengths of these models and combining them in an ensemble approach, the framework aims to achieve highly accurate and interpretable results that can be utilized in clinical practice, including in wearable health monitoring systems.

1.4 OBJECTIVES

The primary objectives of this study are:

- To enhance noise robustness to minimize the impact of background interference.
- To capture long-term dependencies and temporal patterns in PCG signals for improved murmur classification.
- To develop an ensemble learning framework combining Multi head self attention transformer and PatchTST models to enhance classification accuracy and robustness across diverse datasets.
- To improve model generalization across different recording conditions.

CHAPTER 2

LITERATURE SURVEY

This section provides a review of research contributions in the area of heart murmur classification. All the studies in this section employed Phonocardiogram (PCG) signal dataset for their experiments, highlighting the effectiveness of heart sound analysis in diagnosing various cardiovascular conditions.

2.1 DEEP LEARNING APPROACHES FOR MURMUR DETECTION AND GRADING

Elola et al. (2022) developed a deep learning system capable of grading heart murmurs using phonocardiogram (PCG) signals. Their work, as part of the George B. Moody PhysioNet Challenge 2022, involved an ensemble of 15 convolutional residual neural networks with channel-wise attention. The PCG signals were transformed into Mel spectrograms to serve as model input. This framework successfully distinguished among murmur grades—absent, soft, and loud. Their model achieved sensitivities of 90.7% (absent), 75.8% (soft), and 92.3% (loud) during cross-validation. On the hidden test set, it reached an unweighted average sensitivity of 80.4% and an F1-score of 75.8%. The system's ability to automatically classify murmur intensity supports timely clinical decision-making, particularly in settings with limited access to experienced cardiologists.

S. Das and S. Dandapat (2024) presented a novel multi-kernel residual CNN architecture to classify murmur severity from PCG data. The use of kernels with varying receptive fields enabled the model to capture both short-term and long-term patterns across time and frequency domains. Residual connections enhanced gradient flow and allowed for deeper model structures. Their model demonstrated significant improvements over traditional CNNs and support vector machines (SVMs) in classification accuracy and convergence. The ability to learn complex heart sound patterns emphasized deep learning's suitability for murmur detection tasks.

H. K. Alkahtani et al. (2024) developed a deep neural network trained on targeting congenital heart diseases (CHDs) in paediatric. The network architecture, likely combining convolutional and fully connected layers, achieved high sensitivity and specificity, effectively differentiating CHDs from normal heart sounds. This model addresses the challenges of early-stage CHD diagnosis in children a critical factor in preventing long-term complications and demonstrates the growing role of AI in paediatric cardiology.

2.2 HYBRID AND METAHEURISTIC TECHNIQUES IN PCG SIGNAL PROCESSING

A. Q. Aldhahab et al. (2024) proposed a hybrid classification framework that merges Wavelet Scattering Transform (WST) for feature extraction with the Equilibrium Optimization Algorithm (EOA) for feature selection and classification. WST provided robust multi-scale time-frequency representations of heart sounds, preserving key structural features. EOA—a physics-inspired metaheuristic algorithm optimized the feature space to enhance model accuracy and reduce redundancy. The system performed exceptionally well in noisy environments, making it suitable for real-world deployments such as mobile health monitoring.

Navin, K. S. et al. (2023) introduced a filter–wrapper feature selection method for congenital heart failure detection using PCG signals. The filter phase used statistical metrics (e.g., mutual information, correlation) to shortlist features, while the wrapper phase validated them using classifier performance. This two-step process reduced model complexity and training time while improving classification accuracy. Their approach demonstrated computational efficiency and strong predictive performance, making it ideal for clinical decision support systems.

B. Walker et al. (2023) proposed a Dual Bayesian ResNet (DBRes) architecture that integrated uncertainty estimation with murmur detection. The model segmented PCG recordings into overlapping mel spectrogram windows and performed two-stage binary classification. By incorporating Bayesian layers, the model quantified predictive uncertainty—crucial for clinical contexts where interpretability and caution are needed. DBRes achieved a murmur detection weighted accuracy of 0.771 on the hidden test set and improved to 0.820 when combined with demographic features and XGBoost-based post-processing. This architecture demonstrates a balanced blend of performance and reliability.

2.3 ADVANCED NEURAL NETWORK ARCHITECTURES FOR HEART SOUND ANALYSIS

M. Alkhodari et al. (2024) used transformer-based deep learning models to detect congenital valvular murmurs in children. The dataset comprised 942 patients' PCG recordings collected from four auscultation locations. Recordings were processed using discrete wavelet transforms before being input to the transformer model. Attention mechanisms allowed the network to prioritize diagnostically relevant time points in the signal. The model achieved 90.23% average accuracy and 72.41% sensitivity in cross-validation, and 76.10% accuracy on unseen data, proving its potential for non-invasive, scalable cardiac screening in paediatric care.

N. A. Vinay et al. (2024) introduced a hybrid model combining Bi-directional LSTM (BiLSTM) with a Multi-Decision Generative Adversarial Network (GAN). The BiLSTM component captured both past and future dependencies in the heart sound sequence, enhancing temporal modeling. The GAN generated realistic synthetic samples to balance the dataset and improve model generalization. Additionally, a feature optimization module improved computational efficiency. This model design addresses key challenges such as class imbalance and noise—common in real-world PCG data.

S. Das, D. Jyotishi, and S. Dandapat (2023) introduces a hybrid deep learning architecture was developed by combining Stationary Wavelet Transform (SWT) with a hierarchical attention-based LSTM network. SWT decomposed the PCG signals into frequency bands while preserving time resolution, aiding feature representation. The LSTM network, enhanced with attention layers, learned multi-level sequential dependencies, focusing on diagnostically relevant regions of the PCG signal. This architecture improved interpretability and yielded high diagnostic accuracy, particularly for valvular heart diseases.

S. K. Ghosh et al. (2022) presented a time–frequency domain neural network architecture that utilized both spectrogram-based features and raw signal input. This dual-input approach helped the model learn robust representations even in the presence of noise and inter-patient variability. The hybrid design was validated across multiple datasets and showed promising generalization, highlighting its utility for real-time clinical deployment in PCG-based diagnostic systems.

J. Lee et al. (2022) proposed a deep learning model that transformed PCG signals into frequency-time spectrograms to capture both temporal and spectral characteristics. The network architecture included convolutional layers followed by fully connected classifiers. This dual-domain strategy allowed the model to detect murmur presence more effectively than time-domain-only approaches.

Their work emphasizes the importance of comprehensive signal representation for reliable murmur classification.

N. S. Bathe and V. Ingale (2022) analysed the performance of deep learning (CNN) versus traditional machine learning (SVM) models in murmur detection. CNNs, trained end-to-end with automatic feature extraction, significantly outperformed SVMs that relied on handcrafted features. The results highlighted the advantages of deep learning—particularly scalability, accuracy, and reduced need for domain-specific preprocessing—making it a preferred choice in modern diagnostic tools.

2.4 CONCLUSION

The reviewed studies demonstrate significant advancements in the use of AI for heart murmur detection using PCG signals. Deep learning models, including CNNs, RNNs, and transformers, have shown substantial improvements in detecting murmurs and grading their severity. Hybrid models, which combine time-frequency analysis with advanced optimization algorithms, have proven effective in handling the complexities of heart sound data. The literature provides a clear direction for future research, particularly in the areas of hybridization, optimization techniques, and model interpretability. The ensemble-based approach adopted in this thesis aims to build model, further improving the performance and applicability of heart murmur detection systems.

CHAPTER 3

PROPOSED WORK

3.1 DATASET DESCRIPTION

The dataset used in this project is the CirCor DigiScope Phonocardiogram Dataset, released as part of the 2022 George B. Moody PhysioNet Challenge. This dataset contains phonocardiogram (PCG) recordings from paediatric patients in clinical environments, aiming to detect heart murmurs. It is considered one of the largest and most diverse collections of heart sound.

Each patient record consists of four separate auscultation site recordings, capturing heart sounds from different valves: the Aortic Valve (AV), Pulmonary Valve (PV), Tricuspid Valve (TV), and Mitral Valve (MV). These recordings are essential for providing comprehensive coverage of the heart's activity, as each valve can reveal distinct acoustic features that aid in the accurate detection of abnormalities such as heart murmurs.

Each phonocardiogram (PCG) recording in the dataset is stored in .wav format, sampled at a frequency of 4000 Hz, which ensures high-resolution capture of heart sound signals. Alongside each audio recording, there are corresponding diagnostic labels that indicate whether a heart murmur is present (1) or absent (0), facilitating supervised learning tasks. Additionally, each patient record is accompanied by a .tsv metadata file that provides detailed information such as patient demographics, auscultation site descriptions, and clinical annotations, helping to contextualize the recordings. For the purposes of this project, a total of 164 patient records were used. These recordings provide a comprehensive view of heart activity from multiple perspectives. The dataset's rich diversity and structure make it highly suitable for training machine learning and deep learning models to accurately detect and classify heart murmurs in patients.

3.2 SYSTEM DESIGN

The proposed system aims to automate the detection of heart murmurs from Phonocardiogram (PCG) signals using a stacked ensemble deep learning model.

The architecture of the system is illustrated in Figure 3.1, showcasing the flow from raw PCG signals to final murmur prediction via ensemble learning.

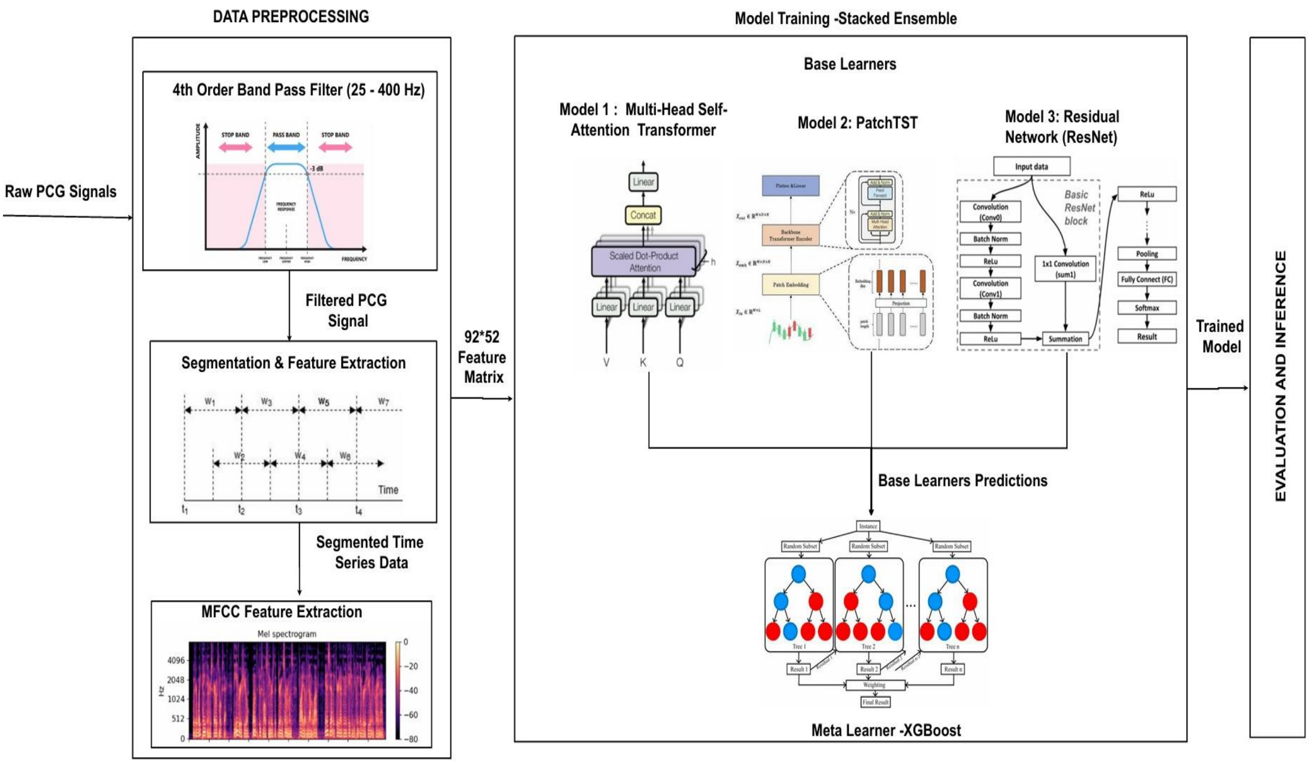


Figure 3.1 System Design

3.2.1 Data Preprocessing

Preprocessing plays a crucial role in transforming raw phonocardiogram (PCG) signals into a structured format suitable for machine learning models. It involves cleaning the signal, removing irrelevant noise, and extracting meaningful features that can be used for training a predictive model. In this phase, several important techniques are applied to ensure the data is properly prepared for further analysis. The three main steps involved in this process are Butterworth Bandpass Filtering, Temporal Sliding Window-based Segmentation, and MFCC Feature Extraction.

3.2.1.1 Butterworth Bandpass Filtering:

Raw PCG signals typically contain noise from various sources, such as background environmental sounds, respiratory noises, or equipment artifacts. These noises can interfere with the analysis of heart murmurs, which are typically within a certain frequency range. To improve the quality of the signal and preserve the essential components of the heart sound, a Butterworth Bandpass Filter is applied.

The Butterworth filter is chosen due to its ability to provide a flat frequency response in the passband, ensuring minimal distortion of the desired signal. The passband for this filter is typically set between 25 Hz and 400 Hz, as the majority of heart murmurs occur within this range. The filter removes unwanted low-frequency noise (such as baseline wander) and high-frequency artifacts (such as electrical noise), thus isolating the relevant frequencies of interest.The filtering process ensures that the signal fed into the model retains only the clinically relevant heart sound frequencies, while suppressing unwanted noise, which could otherwise degrade the performance of machine learning algorithms.

3.2.1.2 Temporal Sliding Window-based Segmentation:

Heart sounds are continuous and vary over time, meaning they may contain transient murmurs that can be challenging to analyse in a single, long signal. To handle this, the filtered signal is segmented into smaller, fixed-duration windows. Each window typically lasts for 1 second, and there is a 50% overlap between consecutive windows. This approach is based on the principle of sliding window segmentation, which divides the signal into smaller overlapping segments to capture both short-duration and long-duration murmurs effectively. The overlapping windows ensure that no important information is lost at the boundaries of the segments, especially for continuous murmurs that may span across the boundary of a single window. Each window represents a snapshot of the signal at a specific moment in time, preserving both the transient and continuous nature of heart sounds.

The windowing technique helps in capturing the temporal dynamics of the heart sounds, allowing for a more granular analysis of the signal. This segmentation method also increases the variability of the data, making it more robust for machine learning models to identify patterns within the heart sounds.

3.2.1.3 MFCC Feature Extraction

Once the signal has been segmented into 1-second windows, the next step is to extract meaningful features. A common technique for extracting relevant features from audio signals is Mel-frequency cepstral coefficients (MFCCs). MFCCs are a representation of the short-term power spectrum of the sound and are widely used in audio processing due to their ability to capture important characteristics of the sound signal.

To compute the MFCCs, each window of the signal undergoes several stages:

Framing and Windowing:

The signal within each window is divided into smaller frames (typically 25 ms long, with 10 ms overlap). This allows the signal to be treated as quasi-stationary, making it easier to analyse its frequency components.

Fourier Transform and Spectral Analysis:

For each frame, the Short-Time Fourier Transform (STFT) is applied to obtain the frequency spectrum of the signal. This step converts the signal from the time domain into the frequency domain, where the different frequency components of the sound can be analysed.

Mel Filter Bank Processing:

The frequency spectrum is then passed through a set of triangular filters that are spaced according to the Mel scale, which is a logarithmic scale that better approximates human hearing. This step emphasizes the frequencies that are more perceptible to human ears while downplaying less relevant frequencies.

Logarithmic Compression:

The energies from the Mel filters are logarithmically compressed, which simulates the way humans perceive sound intensity. This step reduces the dynamic range of the frequencies, making the feature set more suitable for machine learning models.

Discrete Cosine Transform (DCT):

Finally, a Discrete Cosine Transform (DCT) is applied to the log-compressed Mel energy values to decorrelate the features. This step produces a compact representation of the signal, with the first 13 coefficients capturing the most significant information about the frequency content of the heart sound.

The result of this process is a set of 52 MFCC features (13 features per valve) for each window, which provides a detailed representation of the frequency content of the heart sound. For each patient, these MFCC features are organized into a matrix, where each row corresponds to a 1-second window of the signal, and each column corresponds to one of the 52 MFCC features.

3.2.2 Stacked Ensemble

A stacked ensemble model combines multiple models to improve classification accuracy. In this approach, several base learners are trained independently, and their predictions are then combined using a meta-learner. This ensemble method benefits from the diverse strengths of the individual models, allowing for more robust and accurate predictions. By leveraging the complementary learning patterns of different algorithms, the stacked ensemble reduces overfitting and enhances generalization performance. In this project, three distinct deep learning models Multi-Head Self-Attention Transformer, PatchTST, and ResNet18 serve as base learners. Their outputs are then passed to the XGBoost meta-learner, which combines their prediction. Figure 3.2 shows the network diagram of the stacked ensemble architecture used in this project.

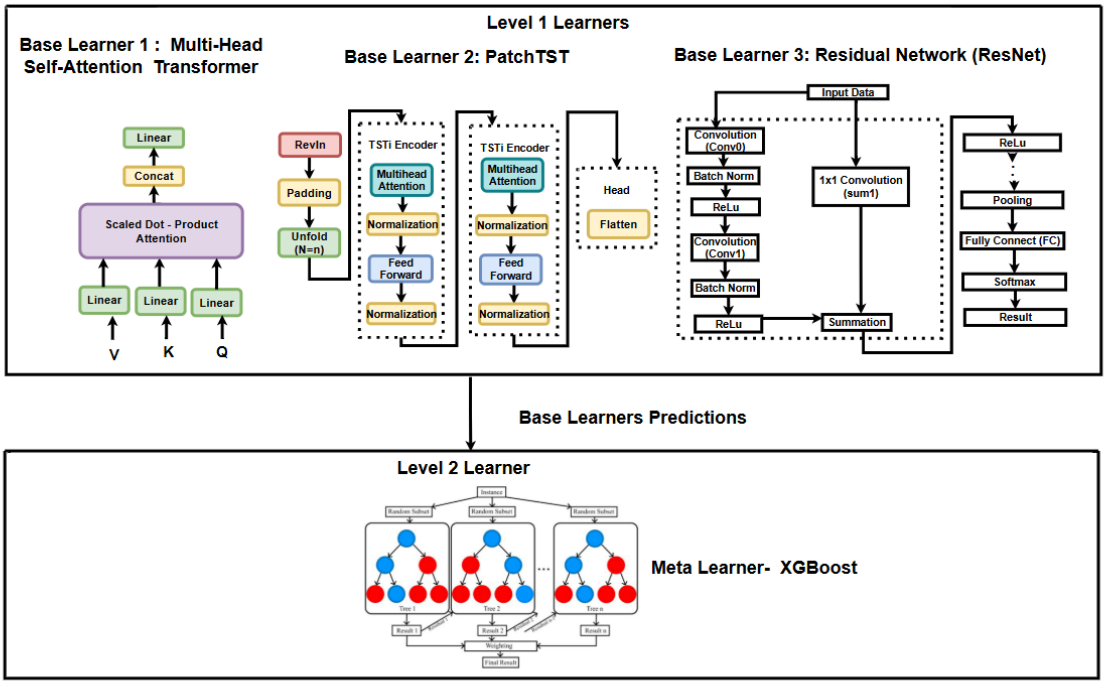


Figure 3.2 Stacked Ensemble Network Diagram

Multi-Head Self-Attention Transformer:

The Multi-Head Self-Attention Transformer is a neural network model that can learn long range dependencies dependencies from time-series data. Each attention head in the model independently learns to focus on different parts of the sequence, allowing the network to understand temporal patterns from multiple perspectives. This multi-headed mechanism is especially powerful in identifying subtle or intermittent murmur signatures spread across long PCG recordings. The model is composed of positional encoding layers, attention layers, feed-forward networks, and residual connections, making it capable of modelling both local and global patterns simultaneously. Compared to recurrent models like LSTMs, transformers offer parallel computation and improved efficiency for longer sequences. The MHSA model in this work helps in capturing rhythm and frequency shifts in heart sounds, which are critical indicators of underlying murmur conditions.

PatchTST (Patch Time Series Transformer):

PatchTST is a modern transformer architecture that treats the time-series input as a sequence of non-overlapping patches. This is inspired by the Vision Transformer (ViT) architecture, where input images are split into patches before being processed. PatchTST adopts the same principle by segmenting the PCG signal into fixed-size time patches, enabling the model to learn relationships between these temporal patches using a self-attention mechanism.This patch-based strategy enhances the model’s capacity to learn meaningful time dependencies without being computationally expensive. The model processes each patch as a vectorized embedding, allowing it to extract both fine-grained local patterns and long-term dependencies. PatchTST is particularly suitable for heart murmur detection as murmurs often span across multiple cardiac cycles. Its ability to capture both recurring and transient phenomena makes it a highly valuable component of the ensemble.

ResNet18:

ResNet18 is a convolutional neural network that employs residual learning to effectively train deep models. Its identity-based skip connections prevent vanishing gradients, ensuring that deep feature hierarchies are learned even in shallow networks. This architecture is composed of 17 convolutional layers followed by a final fully connected layer, making it efficient and effective for time-series classification tasks.The convolutional layers are used for identifying localized features such as pitch, tone, and repetitive patterns in the signal. Murmurs, which often manifest as abnormal vibrations or continuous turbulent flow sounds, can be effectively detected using ResNet’s hierarchical feature extraction.

XGBoost (Extreme Gradient Boosting):

XGBoost is used as the meta-learner in the stacked ensemble, responsible for combining the predictions of the base learners and making the final decision. It is a gradient boosting framework known for its regularization capability, speed, and high predictive accuracy. After the base learners (MHSA, PatchTST, and ResNet18) produce their predictions, XGBoost takes these as features and learns how to optimally weigh and combine them to reduce overall prediction error. XGBoost works by sequentially adding decision trees that focus on correcting the mistakes made by the previous trees. This results in a powerful ensemble of trees capable of capturing nonlinear decision boundaries. In this system, XGBoost enables the model to handle complex interactions among base model predictions and ensures the final ensemble output is highly accurate, even when individual base models may produce conflicting results.

3.2.3 Evaluation and Metrics

The performance of the heart murmur classification models is assessed using standard classification metrics derived from the confusion matrix and the classification report, which are essential for evaluating how well the models distinguish between the two classes: murmur Present and Absent. The confusion matrix offers a comprehensive overview of the prediction outcomes by categorizing them into four groups: True Positives (TP), where murmurs are correctly identified as present; True Negatives (TN), where non-murmur cases are correctly recognized as absent; False Positives (FP), where non-murmur cases are incorrectly classified as murmurs; and False Negatives (FN), where actual murmurs are missed and wrongly labelled as non-murmur. These values form the basis for key performance metrics such as accuracy, precision, recall, and F1-score.

To quantitatively evaluate the model, the following metrics are used:

Accuracy:

Accuracy measures the overall correctness of the model. Equation 3.1 specifies the formula for calculating the model's overall accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

Precision:

Precision evaluates the correctness of positive predictions. Equation 3.2 specifies the formula for determining the precision of murmur detection.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.2}$$

Recall:

Recall measures the model's ability to find all relevant positive instances. Equation 3.3 specifies the formula for evaluating the model's recall in identifying murmurs.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.3}$$

F1-Score:

The F1-score is the harmonic mean of precision and recall, providing a balance between them. Equation 3.4 specifies the formula for calculating the F1-score to balance precision and recall.

$$\text{F1-Score} = 2 \times \frac{PRECISION+RECALL}{PRECISON*RECALL} \tag{3.4}$$

CHAPTER 4

IMPLEMENTATION

This chapter elaborates on the implementation pipeline of the proposed murmur detection framework. The system comprises four major components: signal preprocessing using a Butterworth filter, MFCC-based feature extraction, base learners for learning representations (Transformer, PatchTST, and ResNet18), and a final ensemble layer using XGBoost.

4.1 FILTERING ALGORITHM

Nyquist Frequency:

Nyquist frequency is a fundamental concept used to prevent aliasing when sampling continuous signals. It is defined as half of the sampling rate. Equation 4.1 defines the Nyquist frequency.

$$f\_nyquist = f\_sampling/2 \quad (4.1)$$

where f_nyquist is the Nyquist frequency and f_sampling is the sampling rate of the PCG signal in Hertz.

Z-Score Normalization:

Z Score normalization is normalization technique that transforms the data such that it has a mean of 0 and a standard deviation of 1. Equation 4.2 specifies the standard Z-score normalization applied to the filtered PCG signal.

$$x\_i\_norm = (x\_i\_filtered - \mu) / \sigma \quad (4.2)$$

where x_i_norm is the normalized value of the i^{th} sample, x_i_filtered is the filtered PCG signal of the i^{th} sample, μ is the mean of the filtered PCG signal and σ is the standard deviation of the filtered PCG signal.

PCG_Filtering (X, LC, HC, Order)

Input:

X : Set of PCG patient data samples $\{x_1, x_2, ..., x_n\}$, each with 4 valve recordings, LC : Lower cutoff frequency (25 Hz), HC : Higher cutoff frequency (400 Hz), Order : Order of the Butterworth filter (e.g., 4)

Output:

Filtered and normalized signals for each x_i and each heart valve stored as WAV files.

Algorithm:

1. For each patient sample $x_i \in X$, do:
2. For each valve signal in x_i = [x_i_AV, x_i_MV, x_i_PV, x_i_TV], do:
3. Read the raw signal and sampling rate fs.
4. If the signal is stereo, convert it to mono by averaging channels.
5. Compute Nyquist frequency:

   f_nyquist = fs / 2
6. Compute normalized cutoff frequencies:

   low = LC / f_nyquist

   high = HC / f_nyquist
7. Design a Butterworth bandpass filter using low, high and order
8. Apply zero-phase filtering using filtfilt() to obtain x_i_filtered.

9. Perform Z-score normalization:

$$x_i\_norm = (x_i\_filtered - \mu) / \sigma$$

10. Save the normalized and filtered x_i_norm as a .wav file.

14. End for

15. End for

4.2 FEATURE EXTRACTION ALGORITHM

Step Size:

The step size determines the interval at which consecutive windows are extracted from the signal. It depends on the window size, the amount of overlap between windows, and the sampling rate. Equation 4.3 specifies the step size in terms of time and sampling frequency.

Step_Size = (Window_Size - Overlap) * Sampling_Rate (4.3)

Where, Step_Size indicates how many samples apart each window is when sliding over the signal, Window_Size refers to the duration of each window in seconds, Overlap is the amount of time shared between successive windows and the Sampling_Rate denotes the number of samples taken per second (measured in Hz).

Window Length:

The window length refers to the total number of samples contained in each segment or window of the signal. Equation 4.4 defines the number of samples in one full window.

Window_Length = Window_Size * Sampling_Rate (4.4)

Where, Window_Length represents the number of samples corresponding to one full window, Window_Size again refers to the time duration of each window, and Sampling_Rate denotes the number of samples taken per second (measured in Hz).

Feature_Extractor (X, Sr, W, Overlap, N)

Input:

Y: Set of filtered PCG patient signals $\{y_1, y_2, ..., y_n\}$, each with 4 valves,

Sr: Sampling rate (4000 Hz), W: Duration of sliding window in seconds,

Overlap: Duration of overlap between windows in seconds (0.5),

N: Number of MFCC coefficients to extract (13)

Output:

MFCC feature matrix of size (Number of Windows x Number of MFCC across all valves)

Algorithm:

1. Compute segmentation parameters:

   Step_Size = (W_Size − Overlap) × Sr

   Window_Length = W_Size × Sr

2. For each sample $x_i \in X$, do:
3. For each valve signal x_i_valve ∈ [x_i_AV, x_i_MV, x_i_PV, x_i_TV], do:
4. Segment x_i_valve with sliding window of length Window_Length and stride Step_Size.
5. End For

6. For each segment of size Window_Length, do:

7. Extract MFCC matrix:

   MFCC=librosa.feature.mfcc(y=segment, sr=Sr, n_mfcc=N_MFCC)

8. Compute mean MFCC vector:

   MFCC_Feature_Vector = mean(MFCC, axis=1)

10. End for

11. Align all four valve feature lists to the length of the longest list by padding with NaN vectors.

12.. Concatenate all four valve feature vectors for each time window to form the final MFCC matrix for x_i.

13. Store the MFCC matrix of x_i for model training or analysis.

14. End for

4.3 BASE LEARNER 1 – MULTI-HEAD SELF-ATTENTION TRANSFORMER

HMT(Feature_Matrix, Segments, MFCC_Features, Num_Heads, Num_Layers, Epochs, Batch_Size, Optimizer)

Input:

Feature_Matrix: MFCC feature matrix of shape (Segments x MFCC_Features), Segments = 91 (Number of windowed segments from the PCG signal), MFCC_Features = 52 (Extracted MFCC coefficients across 4 heart valves), Num_Heads = 4 (Number of attention heads),Num_Layers = 6

(Number of Transformer encoder layers), Epochs = 50, Batch_Size = 41, Optimizer = Adam

Output:

Binary classification of murmur (0: Absent, 1: Present).

Algorithm:

1. Project Feature_Matrix into a 128-dimensional space using a dense layer.

2. Add sinusoidal positional encoding for sequence retention.

3. For each of the 6 Transformer encoder layers:

4. Apply a 4-head self-attention, add residual connections, normalize, and pass through a 2-layer feed-forward network.

5. Aggregate features using Global Average Pooling (GAP).

6. Normalize the feature matrix for consistency.

7. Perform data augmentation (if applicable).

8. For each epoch:

9. Optimize using Adam with binary cross-entropy loss.

10. Update weights with a batch size of 41.

11. Implement learning rate scheduling for improved convergence.

12. Implement early stopping based on validation loss.

12.Validate performance after each epoch.

14.Save the best model based on validation performance.

15.Classify using a sigmoid-activated dense layer for binary output.

4.4 BASE LEARNER 2 – PATCH TST

P_TST(Feature_Matrix, config, epochs, batch_size, lr, weight_decay)

Input:

Feature_Matrix (91×52),PatchTST configuration (config),Training parameters: epochs, batch_size, lr, weight_decay

Output:

Binary classification of murmur (0: Absent, 1: Present).

Algorithm:

1. Initialize PatchTST with 52 input channels, 91 context length, 12 patch length, 0.5 dropout, and a CLS token.
2. Normalize Feature_Matrix to ensure all features have zero mean and unit variance.
3. Split the normalized Feature_Matrix into train and test sets, using an 80-20 split.
4. Load the train-test sets into batches with a batch size of 16.
5. Set the learning rate (lr) to 1e-4 and weight decay to 1e-3 for the Adam optimizer.
6. Train the model for 200 epochs using CrossEntropyLoss and the Adam optimizer.
7. For each epoch:
8. Perform a forward pass and compute the loss using CrossEntropyLoss.

9. Backpropagate the error and update the weights of the network using Adam optimizer.
10. Track the average loss across all batches in the epoch.
11. End For
12. Compute the average loss across all batches.
13. Use the trained model to classify murmur as either Present or Absent for each patient in the test set.

4.5 BASE LEARNER 3 – RESNET18 (RESIDUAL NEURAL NETWORK)

Resnet18 (Feature_Matrix, epochs, batch_size, optimizer, loss_function)
Input:

Feature_Matrix: A NumPy array of shape (91, 52) containing MFCC features for 91time segments, each with 52 extracted MFCC coefficients, Epochs=150, Batch_Size = 16,Optimizer = Adam with learning rate of 0.001,Loss Function = CrossEntropyLoss with label smoothing (0.1)

Output:

Binary classification of murmur (0: Absent, 1: Present).

Algorithm:

1. Initialize a custom ResNet18 model adapted for tabular input with 2 output neurons and a sigmoid activation for binary classification.
2. Define the loss function as CrossEntropyLoss with label smoothing set to 0.1.
3. Initialize the optimizer as Adam with learning rate = 0.001 and default beta parameters.

4. Load the 91×52 MFCC Feature_Matrix..
5. Normalize the feature matrix to ensure each feature has zero mean and unit variance.
6. Split the normalized Feature_Matrix into training (80%) and validation (20%).
7. Convert the datasets into PyTorch tensors and wrap them using TensorDataset and DataLoader with batch size 16.
8. For each epoch:
9. Set the model to training mode.
10. Initialize running_loss to 0.
11. For each batch in the training set:
12. Perform the forward pass and compute the predicted outputs.
13. Calculate the loss using CrossEntropyLoss with label smoothing.
14. Perform the backward pass (backpropagation) to compute gradients.
15. Update model weights using the Adam optimizer.
16. Accumulate the loss for the epoch.
17. End For
18. End For
19. Use the trained ResNet18 model to predict murmur presence or absence for new patients.

4.6 META LEARNER – XGBOOST

XG_Boost_Ensemble(Pred1, Pred2, Pred3, y_true, k=5)
Input:

Predictions from three base learners, Ground truth labels (y_true),Number of folds for cross-validation (k=5).

Output:

Final binary murmur classification (0: Absent, 1: Present).

Algorithm:

1. Load predicted outputs from the three base learners (ResNet18, Transformer, PatchTST) stored as .pkl .
2. Combine (stack) the three prediction vectors to form a (N, 3) feature matrix, where N is the number of samples.
3. Initialize XGBoostClassifier with binary:logistic objective, evaluation metric as 'logloss', learning_rate = 0.1, max_depth = 3, and n_estimators = 100.
4. Perform Stratified K-Fold Cross-Validation with k = 5 to maintain class distribution in each fold.

5. For each of the 5 folds:

6. Split the stacked prediction matrix and true labels into training and validation sets.

7. Train the XGBoost model using the training set.

8. Predict murmur classification on the validation set.

9. Calculate accuracy, log loss, precision, recall, and F1-score for the fold.

10. Store predictions and evaluation metrics for averaging later.
11. Record and plot the training loss (log loss) for all boosting rounds.
12. After cross-validation, compute the average accuracy, precision, recall, and F1-score across all folds.
13. Generate a consolidated classification report and plot the confusion matrix to visualize prediction performance.
14. Use the trained XGBoost meta-learner to make final ensemble predictions for unseen patient data.

CHAPTER 5

EXPERIMENTAL RESULTS

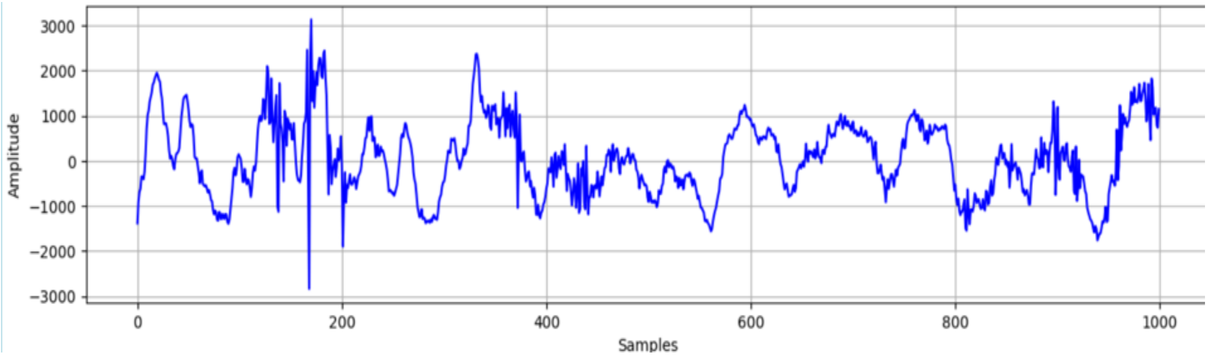
Analysing the performance of machine learning models is a crucial step in understanding their effectiveness. In this section, the outcomes of the heart murmur detection system are evaluated in detail. Each base learner and the final ensemble model are assessed based on classification accuracy using MFCC features extracted from PCG signals. The strengths and comparative performance of the models are discussed with supporting observations.

5.1 DATA PREPROCESSING

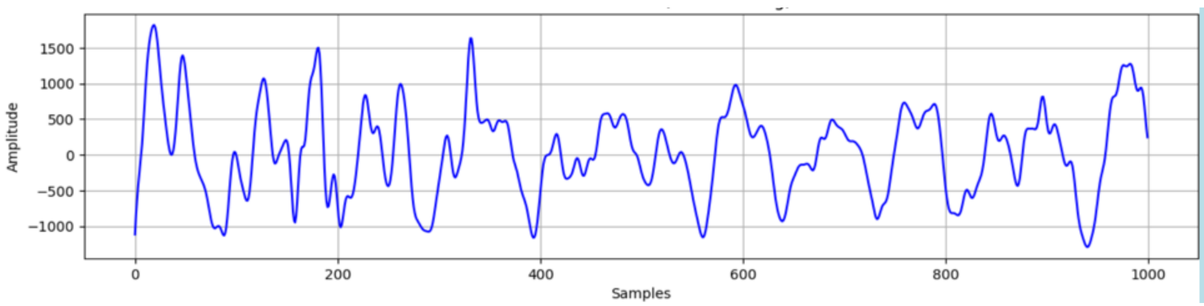
The effectiveness of the preprocessing stage is evident when comparing the raw PCG signal and the filtered signal. Initially, the raw PCG recordings were noisy, with interference from external sources such as ambient noise, respiratory sounds, and equipment artifacts. These unwanted components obscure the key features necessary for accurate murmur detection.

5.1.1 Execution Result of Preprocessing:

The visual comparison between the raw and filtered PCG signals is provided in Figures 5.1(a) and 5.1(b), which demonstrate the transformation achieved through preprocessing.



(a)



(b)

Figure 5.1 (a) Raw PCG Signal (b) Filtered PCG Signal

5.2 PERFORMANCE OF BASE LEARNER 1 – MULTI-HEAD SELF-ATTENTION TRANSFORMER

The Multi-Head Self-Attention Transformer (HMT) model is designed to leverage the power of transformer-based architectures, specifically for time-series classification tasks. The model was trained on the MFCC features extracted from the PCG signals, using a sequence length of 91 segments and 52 MFCC coefficients. The accuracy achieved by the HMT model was **92.07%**.

5.2.1 Key Observations:

The Multi-Head Self-Attention Transformer (HMT) demonstrated strong performance in capturing the temporal dependencies present in phonocardiogram (PCG) signals. This was primarily due to the model's ability to attend to different segments of the signal simultaneously, which enhanced its sensitivity to relevant patterns. The incorporation of multiple attention heads allowed the model to focus on diverse aspects of the time-series data, improving its capacity to extract meaningful features. Overall, the transformer-based architecture provided a significant advantage in learning complex relationships and identifying subtle characteristics associated with heart murmurs.

5.2.2 Execution Result of Multi-Head Self-Attention Transformer:

Figure 5.2 illustrates the training loss curve of the Multi-Head Self-Attention Transformer (HMT) model.

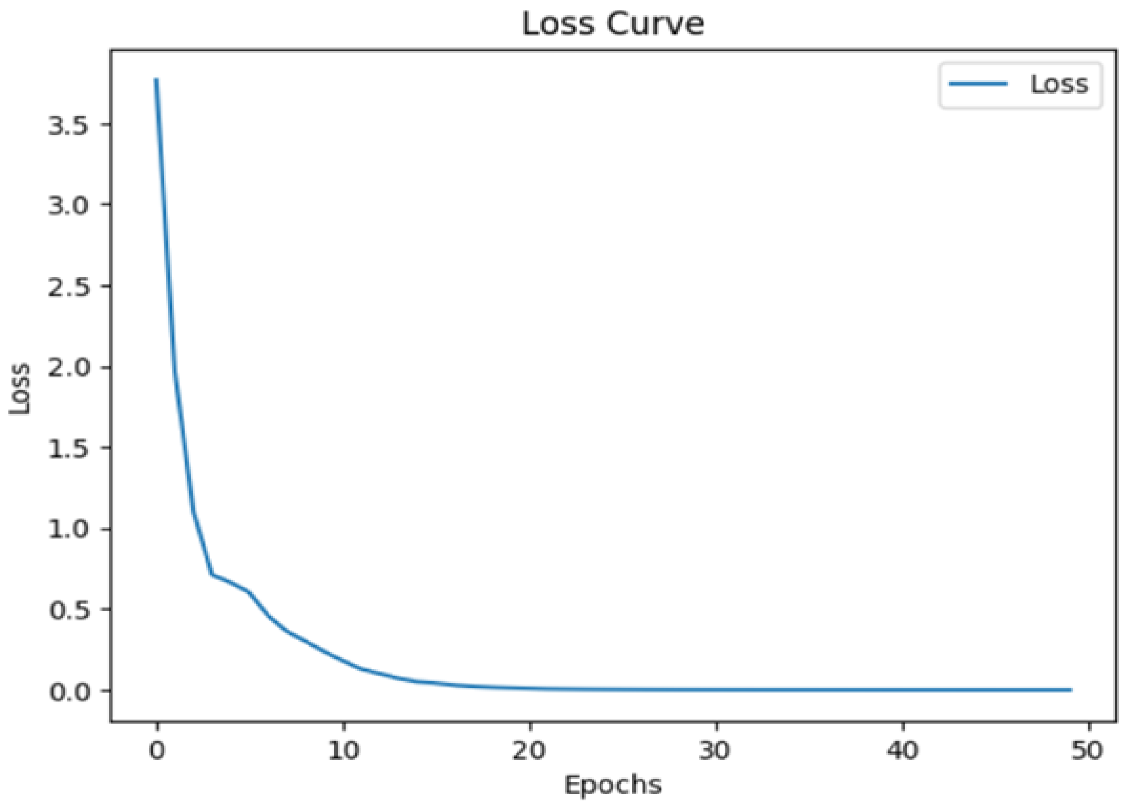


Figure 5.2 Training Loss Curve of Multi Head Self Attention

Table 5.1 illustrates the evaluation metrics of Multi Head Self Attention Transformer.

Table 5.1 Performance of Multi Head Self Attention Transformer

METRICS	PRECISION	RECALL	F1-SCORE
MURMUR ABSENT	0.90	0.95	0.93
MURMUR PRESENT	0.94	0.88	0.91
ACCURACY			0.92
MACRO AVG	0.92	0.92	0.92
WEIGHTED AVG	0.92	0.92	0.92

5.3 PERFORMANCE OF BASE LEARNER 2 – PATCHTST

PatchTST, a transformer variant designed for multivariate time-series data, was trained on the same MFCC features as the HMT model. The model was trained for 200 epochs with a batch size of 16. The PatchTST model achieved an accuracy of **90.85%**.

5.3.1 Key Observations:

PatchTST handled the sequential nature of the PCG signals effectively and achieved strong results, although slightly lower in accuracy compared to the HMT model. Its design, which supports larger context windows and the use of multiple patch lengths, proved beneficial in modeling the temporal structure of MFCC feature sequences. However, the marginally lower performance indicates that additional hyperparameter tuning and architectural adjustments could potentially enhance the accuracy and bring it closer to or surpass the performance of the HMT model.

5.3.2 Execution Result of Patch TST

Figure 5.3 illustrates the training loss curve of the Patch TST model.

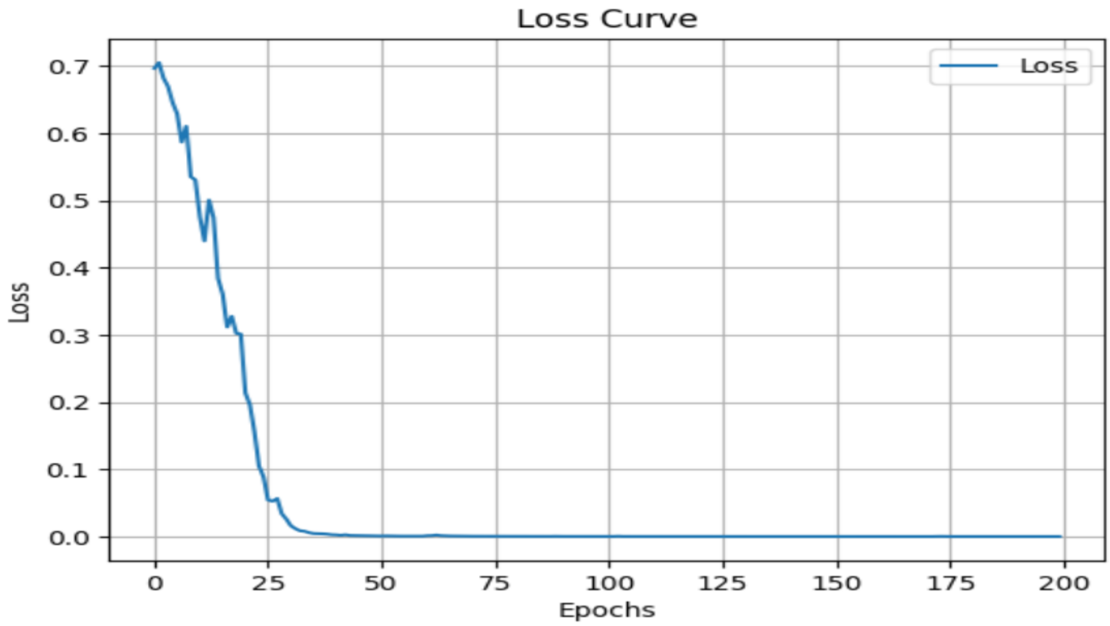


Figure 5.3 Training Loss Curve of Patch TST

Table 5.2 illustrates the evaluation metrics of the Patch TST model.

Table 5.2 Performance of Patch TST

METRICS	PRECISION	RECALL	F1-SCORE
MURMUR ABSENT	0.89	0.94	0.92
MURMUR PRESENT	0.93	0.87	0.90
ACCURACY			0.91
MACRO AVG	0.91	0.91	0.91
WEIGHTED AVG	0.91	0.91	0.91

5.4 PERFORMANCE OF BASE LEARNER 3 – RESNET18

ResNet18, a deep convolutional neural network with residual connections, was also applied to the heart murmur detection task. The ResNet18 model processed the 91×52 MFCC feature matrix and achieved an accuracy of **90.24%**.

5.4.1 Key Observations:

The ResNet18 model, although originally developed for image classification tasks, adapted well to the MFCC feature matrix and delivered performance comparable to PatchTST. The use of residual connections played a crucial role in preventing vanishing gradients during training, enabling the network to learn deeper and more abstract representations of the input data.

5.4.2 Execution Result of Resnet18

Figure 5.4 illustrates the training loss curve of the ResNet18 model.

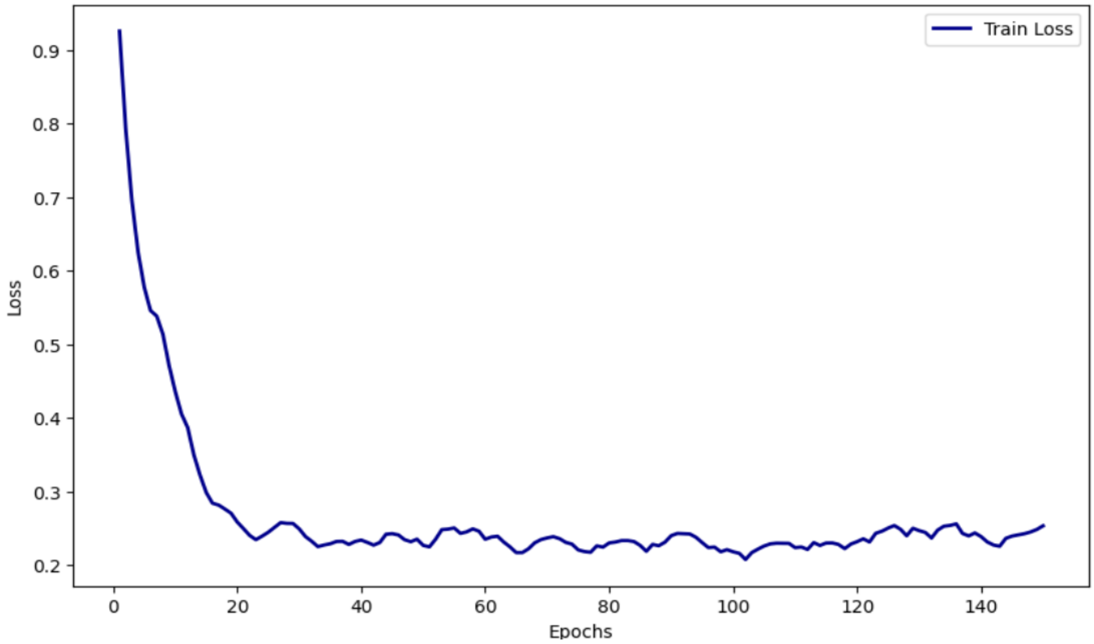


Figure 5.4 Training Loss Curve of Resnet 18

Table 5.3 illustrates the evaluation metrics of the Resnet18 model.

Table 5.3 Performance of Resnet18

METRICS	PRECISION	RECALL	F1-SCORE
MURMUR ABSENT	0.90	0.90	0.90
MURMUR PRESENT	0.89	0.89	0.89
ACCURACY			0.90
MACRO AVG	0.90	0.90	0.90
WEIGHTED AVG	0.90	0.90	0.90

5.5 PERFORMANCE OF THE META LEARNER – XGBOOST

XGBoost, the meta-learner, combined the predictions of all three base learners (HMT, PatchTST, and ResNet18). After stacking the predictions from the base models, XGBoost achieved an overall accuracy of **96.95%**.

5.5.1 Key Observations:

The XGBoost meta-learner outperformed all individual base learners by effectively combining their outputs into a unified prediction model. This ensemble strategy allowed the model to leverage the unique strengths and feature extraction capabilities of each base learner, resulting in more accurate and reliable predictions. The superior performance of XGBoost indicates that stacking the predictions from diverse models is a highly effective approach for heart murmur classification. The ensemble model not only increased overall accuracy but also enhanced generalization.

5.5.2 Execution Result of XGBoost

Figure 5.5 illustrates the training loss curve of the of the XGBoost meta-learner.

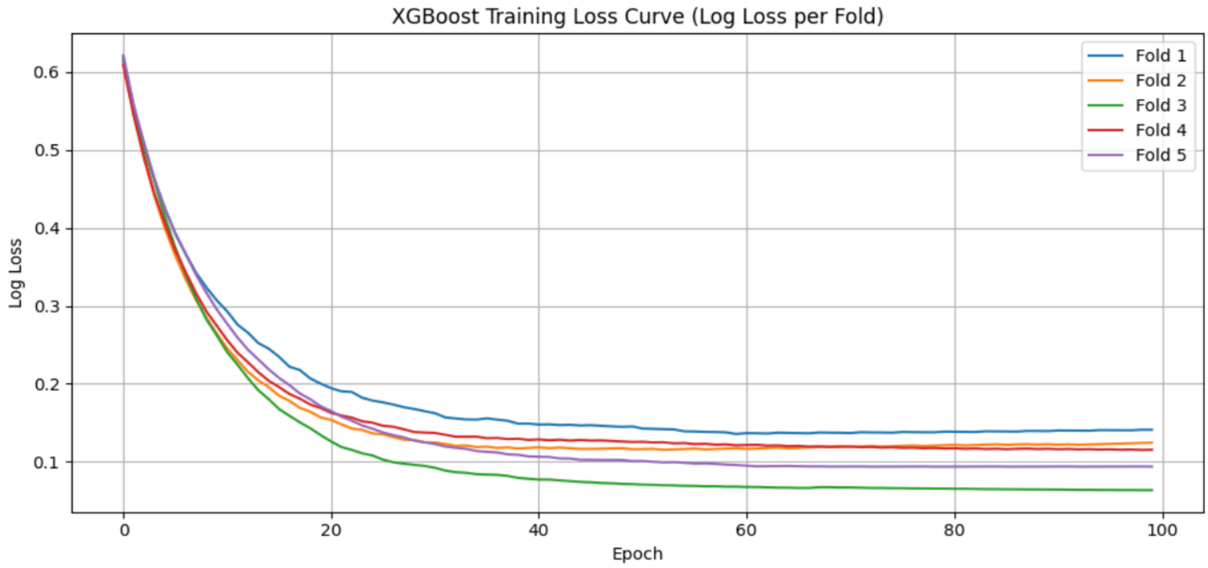


Figure 5.5 Training Loss Curve of XGBoost

Table 5.4 illustrates the evaluation metrics of XGBoost model.

Table 5.4 Performance of XGBoost

METRICS	PRECISION	RECALL	F1-SCORE
MURMUR ABSENT	0.97	0.98	0.97
MURMUR PRESENT	0.97	0.96	0.97
ACCURACY			0.97
MACRO AVG	0.97	0.97	0.97
WEIGHTED AVG	0.97	0.97	0.97

5.6 COMPARATIVE ANALYSIS

To assess the effectiveness of each model in detecting heart murmurs, a comparative analysis was conducted. Evaluating multiple machine learning and deep learning models helps identify the most suitable architecture for robust murmur classification. This comparison not only showcases the performance of individual models but also highlights how ensemble methods can further enhance prediction accuracy by combining their strengths. Both base learners and meta learners were analysed based on their classification accuracy, allowing for a clear understanding of their contributions toward the final prediction outcome.

Table 5.5 presents the standalone accuracy of the individual models.

Table 5.5 Performance of Individual Models

Model	Accuracy (%)
Base Learner 1 – Multi-Head Self-Attention Transformer	92.07
Base Learner 2 – PatchTST	90.85
Base Learner 3 – ResNet18	90.24
Meta Learner – XGBoost	96.95

Table 5.6 evaluates the performance of the ensemble models using different combinations of base learners and XGBoost as meta learners.

Table 5.6 Comparative Study of different base learners with XGBoost as a meta learner

Trials	Base Learners	Accuracy
1	Multi Head Self Attention Transformer Patch TST	**91.52**
2	Multi Head Self Attention Transformer Resnet	**90.91**
3	Patch TST Resnet	**88.43**
4	Multi Head Self Attention Transformer Patch TST Resnet **(Proposed Work)**	**96.95**

CHAPTER 6

CONCLUSION

This project presents a comprehensive approach to the automated detection of heart murmurs using phonocardiogram (PCG) signals. The workflow involved signal preprocessing through a 4th-order Butterworth bandpass filter to isolate the relevant frequency band of 25–400 Hz. The cleaned signals were segmented using 1-second overlapping windows, followed by the extraction of Mel-Frequency Cepstral Coefficients (MFCCs), which captured the spectral features essential for murmur classification.

Three base learners were implemented in this project—Multi-Head Self-Attention Transformer, PatchTST, and ResNet18—each offering different advantages in processing time-series data. The Multi-Head Self-Attention Transformer captured sequential dependencies effectively and achieved an accuracy of 92.07%. PatchTST, with its patch-based representation, offered a lightweight and efficient alternative and achieved 90.85% accuracy. ResNet18, known for its ability to extract deep hierarchical features, reached 90.24% accuracy. To further enhance the classification performance, the outputs of these base learners were combined using an XGBoost meta-learner. This ensemble method significantly boosted the final classification accuracy to 96.95%, outperforming individual models.

The successful implementation and results of this project underline the potential of ensemble learning in biomedical signal classification tasks. It highlights that a well-designed pipeline—starting from proper signal filtering, meaningful feature extraction, and advanced classification techniques—can result in a powerful system capable of assisting in medical diagnosis. With additional real-world testing and deployment strategies, such systems could become valuable diagnostic tools in both clinical and rural settings.

REFERENCES

[1] A. Elola *et al.*, "Beyond Heart Murmur Detection: Automatic Murmur Grading From Phonocardiogram," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 8, pp. 3856–3866, Aug. 2023, doi: 10.1109/JBHI.2023.3275039.

[2] S. Das and S. Dandapat, "Heart Murmur Severity Stages Classification Using Multikernel Residual CNN," *IEEE Sensors Journal*, vol. 24, no. 8, pp. 13019–13027, Apr. 2024, doi: 10.1109/JSEN.2024.3373226.

[3] 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, pp. 76053–76064, 2024, doi: 10.1109/ACCESS.2024.3395389.

[4] A. Q. Aldhahab, M. R. Ismael and H. J. Abd, "Classification of Phonocardiogram Signals Using the Wavelet Scattering Transform and Equilibrium Optimization Approach," *IEEE Access*, vol. 12, pp. 191231–191242, 2024, doi: 10.1109/ACCESS.2024.3518635.

[5] M. Alkhodari, L. J. Hadjileontiadis and A. H. Khandoker, "Identification of Congenital Valvular Murmurs in Young Patients Using Deep Learning-Based Attention Transformers and Phonocardiograms," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 4, pp. 1803–1814, Apr. 2024, doi: 10.1109/JBHI.2024.3357506.

[6] N. A. Vinay, K. N. Vidyasagar, S. Rohith, D. Pruthviraja, S. Supreeth and S. H. Bharathi, "An RNN-Bi LSTM Based Multi Decision GAN Approach for the Recognition of Cardiovascular Disease (CVD) From Heart Beat Sound: A Feature Optimization Process," *IEEE Access*, vol. 12, pp. 65482–65502, 2024, doi: 10.1109/ACCESS.2024.3397574.

[7] S. Das, D. Jyotishi and S. Dandapat, "Automated Detection of Heart Valve Diseases Using Stationary Wavelet Transform and Attention-Based Hierarchical LSTM Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023, Art no. 4005110, doi: 10.1109/TIM.2023.3270974.

[8] S. K. Ghosh, R. N. Ponnalagu, R. K. Tripathy, G. Panda and R. B. Pachori, "Automated Heart Sound Activity Detection From PCG Signal Using Time–Frequency-Domain Deep Neural Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022, Art no. 4006710, doi: 10.1109/TIM.2022.3192257.

[9] J. Lee *et al.*, "Deep Learning Based Heart Murmur Detection Using Frequency-time Domain Features of Heartbeat Sounds," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, 2022, pp. 1–4, doi: 10.22489/CinC.2022.071.

[10] N. S. Bathe and V. Ingale, "Heart Murmur Detection from Phonocardiogram Recordings using Deep Learning Techniques," *2022 International Conference on Futuristic Technologies (INCOFT)*, Belgaum, India, 2022, pp. 1–4, doi: 10.1109/INCOFT55651.2022.10094334.

[11] K. S. Navin *et al.*, "A Classification Framework Using Filter-wrapper Based Feature Selection Approach for the Diagnosis of Congenital Heart Failure," 2023, pp. 6183–6218.

[12] B. Walker, F. Krones, I. Kiskin, G. Parsons, T. Lyons and A. Mahdi, "Dual Bayesian ResNet: A Deep Learning Approach to Heart Murmur Detection," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, 2022, pp. 1–4, doi: 10.22489/CinC.2022.355.

[13] O. R. A. Almanifi, A. F. Ab Nasir, M. A. M. Razman, R. M. Musa and A. P. P. Abdul Majeed, "Heartbeat murmurs detection in phonocardiogram recordings via transfer learning," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 10995–11002, 2022, doi: 10.1016/j.aej.2022.04.031.

[14] M. Alkhodari, S. K. Azman, L. J. Hadjileontiadis and A. H. Khandoker, "Ensemble Transformer-Based Neural Networks Detect Heart Murmur in Phonocardiogram Recordings," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, 2022, pp. 1-4, doi: 10.22489/CinC.2022.035.

[15] S. Parvaneh, Z. Ardalan, J. Song, K. Vyas and C. Potes, "Heart Murmur Detection Using Ensemble of Deep Learning Classifiers for Phonocardiograms Recorded from Multiple Auscultation Locations," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, 2022, pp. 1-4, doi: 10.22489/CinC.2022.241.

[16] A. S. M. Iftekhar, M. M. Rahman, and T. Y. Al-Naffouri, "Automated Heart Murmur Detection using Sound Processing Techniques and Deep Learning," *Procedia Computer Science*, vol. 217, pp. 1–8, 2023.

[17] S. K. Ghosh, R. N. Ponnalagu, R. K. Tripathy, G. Panda, and R. B. Pachori, "CNN-based classification of phonocardiograms using fractal features," *Biomedical Signal Processing and Control*, vol. 85, 2023, Art. no. 104908.

[18] M. S. Islam, M. M. Rahman, and T. Y. Al-Naffouri, "Phonocardiogram signal classification for the detection of heart murmurs using deep learning," *Expert Systems with Applications*, vol. 213, 2023, Art. no. 119157.

[19] A. S. M. Iftekhar, M. M. Rahman, and T. Y. Al-Naffouri, "A deep neural network approach to heart murmur detection using phonocardiogram signals," *Engineering Applications of Artificial Intelligence*, vol. 125, 2024, Art. no. 107314.

[20] A. S. M. Iftekhar, M. M. Rahman, and T. Y. Al-Naffouri, "Exploring finetuned audio-LLM on heart murmur features," *Biomedical Signal Processing and Control*, vol. 85, 2025, Art. no. 104908.

[21] J. Lee et al., "Deep Learning Based Heart Murmur Detection Using Frequency-time Domain Features of Heartbeat Sounds," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, 2022, pp. 1–4, doi: 10.22489/CinC.2022.071.

[22] M. A. Reyna et al., "Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022," PLOS Digital Health, vol. 2, no. 9, pp. e0000324, Sep. 2023, doi: 10.1371/journal.pdig.0000324.

[23] H. Wen and J. Kang, "Searching for Effective Neural Network Architectures for Heart Murmur Detection from Phonocardiogram," arXiv preprint arXiv:2303.02988, Mar. 2023.arXiv+1PMC+1.

[24] A. Patwa, M. M. U. Rahman, and T. Y. Al-Naffouri, "Heart Murmur and Abnormal PCG Detection via Wavelet Scattering Transform & a 1D-CNN," arXiv preprint arXiv:2303.11423, Mar. 2023.arXiv.

[25] B. Walker et al., "Dual Bayesian ResNet: A Deep Learning Approach to Heart Murmur Detection," arXiv preprint arXiv:2305.16691, May 2023.