People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Echahid Cheickh Larbi Tebessi University
Department of Computer Science

Master Thesis

Submitted by:

Bouatouata Chakib

Option:

Information Systems

AI-Driven Cardiac Diagnostics: Leveraging Vision Transformers for PCG Signal Classification

Supervised by:
Dr. Issam BENDDIB

Jury Members:

President:	**Pr.Bendjenna hakim**
Co-President:	**Dr.Zaghari Ahmed**
Framed:	**Dr.Issam BENDDIB**
Supervisor Assistant:	**Dr.Khalifa Boujemaa**

Academic Year: 2024-2025

Signatures

Host Company Supervisor: Mrs./Mr.FisrtName LastName



ENIT Supervisor: Mrs./Mr.FisrtName LastName



Dedication

We would like to dedicate our project to ...

Acknowledgments

We would like to express our special regards...

We would like to also thank ...

Abstract

work presents a comprehensive approach to classifying cardiac sound (PCG) signals into normal and abnormal categories using hybrid deep learning and machine learning techniques. Our goal is to develop a solution based on artificial intelligence techniques. The study leverages the PhysioNet/CinC Challenge 2016 dataset, which consists of recordings from multiple environments and patient cases, introducing significant variability in signal quality, duration, and class balance. A robust preprocessing pipeline was implemented to normalize and segment the signals by patient. Time-domain and frequency-domain features were extracted to improve representation. n our work, we proposed a hybrid architecture that integrates multiple models, long short-term memory (LSTM) networks to model temporal context, and an attention mechanism to weight the most informative heartbeats. The most relevant segments were selected using attention weights, forming the final input to a random forest classifier. Experimental results demonstrate that the proposed system achieves promising classification performance, proving its effectiveness in handling PCG variability and imbalanced real-world data. This work highlights the potential of hybrid deep learning methods in automated cardiac listening systems

Keywords : Heart sound , Phonocardiogram (PCG)„ Feature ,machine learning, Deep learning, CNN, LSTM, Attention mechanism, Random Forest, Signal segmentation, PhysioNet/CinC 2016,SVM,padding,boosting

Résumé

Notre travail présente une approche complète pour classer les signaux du son cardiaque (PCG) en catégories normales et anormales en utilisant des techniques hybrides d'apprentissage profond et d'apprentissage automatique. Notre objectif est de développer une solution basée sur des techniques d'intelligence artificielle. L'étude exploite le jeu de données PhysioNet/CinC Challenge 2016, qui se compose d'enregistrements provenant de plusieurs environnements et cas de patients, introduisant une variabilité significative dans la qualité du signal, la durée et l'équilibre des classes. Un pipeline de prétraitement robuste a été mis en œuvre pour normaliser et segmenter les signaux par patient. Les caractéristiques du domaine temporel et du domaine fréquentiel ont été extraites pour améliorer la représentation. Dans notre travail, nous avons proposé une architecture hybride qui intègre plusieurs modèles, des réseaux de mémoire à long terme (LSTM) pour modéliser le contexte temporel, et un mécanisme d'attention pour pondérer les battements de cœur les plus informatifs. Les segments les plus pertinents ont été sélectionnés à l'aide de poids d'attention, formant l'entrée finale d'un classificateur de forêt aléatoire. Les résultats expérimentaux démontrent que le système proposé atteint des performances de classification prometteuses, prouvant son efficacité dans la gestion de la variabilité du PCG et des données déséquilibrées du monde réel. Ce travail met en évidence le potentiel des méthodes hybrides d'apprentissage profond dans les systèmes d'écoute cardiaque automatisés
xtbfMots clés : Traitement Automatique du Langage Naturel, Intelligence Artificielle, Mot clé, Mot c

Contents

List of Figures **9**

List of Tables **11**

List of Acronyms **12**

General introduction **13**

1 Principles of Cardiovascular Physiology and Automated Classification of Phonocardiogram Signals **15**

1 Introduction 15
2 Motivation and Problem Statement 15
2.1 Heart anatomy 16
2.2 The cardiac cycle 17
2.3 Cardiac Conduction System 18
3 Heart sound diagnosis 19
3.1 Electrocardiogram (ECG) Signals 19
3.2 Phonocardiogram (PCG) Signals 20
4 Cardiac Diseases Using PCG Signals 23
5 Objectives and Contributions 25
5.1 Research Objectives 25
5.2 Key Contributions 25
6 Conclusion 26

2 Approaches for PCG Signal Classification **27**

1 Introduction 27
2 Contribution and methodology for beat sounds classification 28
2.1 Feature Extraction for PCG Signals 29
3 The classification process 33
3.1 Machine Learning classification Algorithms 33
3.2 Deep Learning classification architecture 41

4 Datasets for beat sounds classification . . . 46
4.1 PhysioNet/CinC Challenge 2016 Dataset . . . 47
4.2 PhysioNet/CinC Challenge 2022 Dataset . . . 47
4.3 PASCAL Classifying Heart Sounds Challenge Dataset . . . 47
5 Evaluation Metrics for classification tasks . . . 48
5.1 The Accuracy . . . 48
5.2 The Precision . . . 49
5.3 The Recall . . . 49
5.4 F1-Score . . . 49
5.5 The ROC AUC . . . 49
6 Related Work in beat sonds classification . . . 49
7 Conclusion . . . 52

3 Adaptive PCG Analysis Framework: Attention Mechanisms for Variable-Length Signal Classification 53
1 Introduction . . . 53
2 Data Preparation and Evaluation Protocol . . . 54
2.1 Dataset description . . . 54
2.2 The PCG annotation files . . . 54
2.3 Dataset cleaning process . . . 55
2.4 The used test protocol . . . 57
3 PCG signal segmentation . . . 59
3.1 PCG signal description . . . 59
3.2 PCG signal segmentation description . . . 60
3.3 Cinc 2016 dataset statistics . . . 60
4 Cinc 2016 dataset challenges . . . 61
4.1 The unbalanced dataset challenge . . . 61
4.2 Environment dataset challenge . . . 62
4.3 Variability of Sequence Lengths . . . 62
5 Features Extraction Strategy for PCG Signals . . . 63
5.1 Temporal Features . . . 63
5.2 Signals Length Normalization . . . 64
6 Architecture of the Attention-Based Beat Selection for Variable-Length PCG Signals classification . . . 67
6.1 BiLSTM encoder architecture . . . 68
6.2 The Hyperparametres of the LSTM Model proposed . . . 71
6.3 Training results . . . 71
6.4 Visualization of the attention mechanism with key regions in the PCG signals . . . 73
6.5 Adaptative feature extraction process . . . 75
7 Classification Engine for Beat-Level Cardiac Anomaly Detection . . . 76

7.1 Random Forest: . . . 76
7.2 K-Nearest Neighbors (K-NN): . . . 76
7.3 Support Vector Machine (SVM): . . . 77
7.4 Decision Tree: . . . 77
7.5 Bagging: . . . 77
7.6 Boosting: . . . 77
8 Results and Discussions . . . 77
8.1 Normalization with duplication of the last beat . . . 78
8.2 Normalization with the duplication of whole Signal . . . 79
8.3 Normalization with fixed threshold . . . 80
8.4 Normalization with Attention-based dynamic strategy . . . 82
9 Conclusion . . . 82

Conclusion **83**

Bibliography **84**

List of Figures

1.1 The heart anatomy [1] . . . 17
1.2 The four chambers of heart [1] . . . 18
1.3 Electrical system of the heart [2] . . . 20
1.4 Standard ECG chest electrode positions . . . 20
1.5 Normal phonocardiogram signal for a single cardiac cycle. [3] . . . 22
1.6 the PCG signal including heart sounds (S1, S2, systole,diastole). [4] 23

2.1 Machine learning process diagram . . . 29
2.2 Peak Amplitude curve [5] . . . 31
2.3 Skewness curve [6] . . . 31
2.4 Kurtosis curve . . . 31
2.5 Architecture of SVM[7] . . . 34
2.6 Architecture of Decision Tree [8] . . . 35
2.7 Architecture of Bagging [9] . . . 36
2.8 Architecture of Random forest [10] . . . 37
2.9 Architecture of BRF . . . 39
2.10 Architecture of Gradient boosting [11] . . . 40
2.11 Architecture of of XGBoost [12] . . . 41
2.12 CNN Architectures of CNN [**?**] . . . 42
2.13 Architecture of RNN [13] . . . 43
2.14 Architecture of LSTM [14] . . . 45
2.15 Architecture of BiLSTM [15] . . . 46

3.1 Sample of a0001 . . . 55
3.2 Labeling empty of sample a0006 . . . 56
3.3 Comparison of Normal and Abnormal Cases in Six Patient Groups 57
3.4 dataset sizes . . . 58
3.5 Comparison of Normal and Abnormal Cases in Six Patient Groups whith train and test . . . 59
3.6 Signal length of sample a0001 . . . 60
3.7 Segmentation signal of sample a0001 . . . 60

3.8 Comparison of Normal vs Abnormal Patients 62
3.9 Signal length per patient 63
3.10 Enter Caption 64
3.11 Signal Padding by Last Beat Duplication with Last Beat 65
3.12 Complete Signal Duplication 66
3.13 Threshold strategy 66
3.14 Attention-Based Beat Selection for Variable-Length PCG Signals strategy 67
3.15 Architecture of Top Features 68
3.16 Architecture of LSTM 69
3.17 Training accuracy curve 72
3.18 Training loss curve 72
3.19 Attention Weight Distribution and Positional Focus in the Training Set 73
3.20 Attention Weight Distribution and Positional Focus in the Test Set 73
3.21 Example PCG signal with attention heatmap 74
3.22 Visualization of PCG Signal and Model Attention Weights 75
3.23 Affecting Weights of the beats 76
3.24 Classification for each strategys 78
3.25 Confusion Matrices with random forest 79
3.26 Confusion Matrices with random forest 80
3.27 Confusion Matrices with random forest 81

List of Tables

2.1 Extracted Temporal Features : Basic Signal Characteristics 30
2.2 Main steps in the computation of MFCCs 32
2.3 Summary of works on PCG data with features, classifiers, and results 51

3.1 Distribution of normal and abnormal recordings in the PhysioNet/CinC 2016 training dataset 54
3.2 Distribution of normal and abnormal recordings in the modified PhysioNet/CinC 2016 training dataset 56
3.3 Number of normal and abnormal samples per directory in the train and test sets 58
3.4 Summary of PCG recording durations in the training dataset. . . . 61
3.5 Summary statistics of the beats per patient. 61
3.6 Overview of the heartbeat sequence classification model architecture. 70
3.7 Training Hyperparameters for the Proposed Model 71
3.8 Train and test statistic 78
3.9 Performance de l'Ensemble Classifier 78
3.10 Performance comparison of classification models 79
3.11 Performance metrics of different classification models 79
3.12 Classifier Performance at Different Thresholds 81
3.13 Performance metrics of different classification models with Threshold =55 81
3.14 Performance metrics of different classification models with Attention-based dynamic strategy 82

List of Acronyms

PCG	Phonocardiogram
DL	Deep Learning
ML	Machine Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
S1	First Heart Sound
S2	Second Heart Sound
SMOTE	Synthetic Minority Over-sampling Technique
MFCC	Mel Frequency Cepstral Coefficients
STFT	Short-Time Fourier Transform
WT	Wavelet Transform
F1	F1 Score (Harmonic Mean of Precision and Recall)
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ReLU	Rectified Linear Unit
ADAM	Adaptive Moment Estimation (optimizer)
CV	Cross-Validation
API	Application Programming Interface
ICBHI	International Conference on Biomedical and Health Informatics
MIT-BIH	Massachusetts Institute of Technology – Beth Israel Hospital
CAD	Computer-Aided Diagnosis

General introduction

The effectiveness of artificial intelligence (AI) tools in supporting healthcare practitioners has been demonstrated across various domains, including the interpretation of medical imaging, the prediction of certain pathologies, and the in-depth analysis of digital signals such as electrocardiograms and phonocardiograms. However, the lack of availability of medical specialists—or even the complete absence of medical specialists in certain areas, particularly in rural regions—significantly impacts medical coverage for citizens. This is especially critical for the management of chronic diseases and urgent cases, where timely intervention is essential. Cardiovascular diseases, in particular, pose a major challenge to healthcare delivery, as they remain among the leading causes of mortality worldwide. Their diagnosis requires specific equipment and, crucially, expert interpretation of test results. Immediate and accurate diagnosis in emergency situations can be life-saving and help prevent unforeseen complications. In this context, the present master's thesis aims to contribute to the development of a diagnostic support framework for interpreting one of the most commonly used examinations in cardiac assessment: heart sound analysis via stethoscope. The stethoscope, now enhanced by digital technology, enables the acquisition of digital signals that are manipulable, exploitable, and reusable. This advancement encourages the development of intelligent solutions for signal interpretation using advanced AI algorithms, as well as the creation of interpretable models (XAI) to support practitioners in making reliable, evidence-based decisions. Nevertheless, interpretability in AI-based medical diagnostics requires access to annotated PCG (phonocardiogram) signals by domain experts. Typically, annotations in existing datasets are provided at the patient level, labeling the entire PCG recording. However, clinical diagnosis is often based on the analysis of individual heartbeats. Most studies in the literature detect cardiac rhythm anomalies on a per-beat basis, assuming that all beats within a pathological PCG signal share the same annotation. This assumption is not always valid, as pathological signals may still contain normal beats, thus calling this approach into question. To address this issue, our contribution proposes utilizing the entirety of the PCG signal for anomaly detection, while maintaining beat-level explainability. A major challenge in this approach is the variability in the length of PCG signals. To over-

come this, we have integrated an adaptive feature extraction technique within our framework, incorporating attention mechanisms to preserve the interpretability of diagnostic results. The structure of this thesis is as follows: Chapter 1 outlines the research problem, describes the characteristics of PCG signals, and discusses the challenges associated with their processing. Chapter 2 introduces existing AI models and architectures suitable for PCG anomaly detection, presents the various types of features that can be extracted from PCG signals, and concludes with a literature review of recent work in this area.

Chapter 3 provides a detailed description of our proposed framework's architecture, focusing on each component of the solution, and presents the results obtained along with relevant discussions.

Chapter 1

Principles of Cardiovascular Physiology and Automated Classification of Phonocardiogram Signals

1 Introduction

Cardiology is a medical specialty focused on disorders of the heart and certain components of the circulatory system. It involves the diagnosis and treatment of conditions such as coronary artery disease, heart failure, valvular disorders, and congenital heart defects. Among the most widely used diagnostic tools, the electrocardiogram (ECG) plays a key role in evaluating the heart's electrical activity to guide clinical decisions. Additionally, heart sounds can be captured as signals known as phonocardiograms (PCG). The analysis of these signals begins with a segmentation step, which is considered fundamental in any automated process for detecting and interpreting cardiac sounds.The phonocardiogram (PCG), often synchronized with the electrocardiogram (ECG), can also be auscultated using a stethoscope. Automated analysis of ECG and PCG signals offers significant advantages in cardiology by enhancing diagnostic efficiency, accuracy, and speed.

2 Motivation and Problem Statement

Cardiovascular diseases remain the leading cause of death worldwide, accounting for millions of fatalities each year. Early detection and diagnosis of cardiac abnormalities are therefore essential for timely intervention and effective treatment.

Among the various diagnostic tools, phonocardiography (PCG), which captures heart sounds using an electronic stethoscope, offers a non-invasive, low-cost, and accessible method for assessing cardiac health. Despite its potential, the interpretation of PCG signals is still largely subjective and dependent on the expertise of the clinician. In resource-limited settings, where experienced cardiologists are not always available, the risk of misdiagnosis increases. This highlights the need for automated and reliable classification systems to support medical decision-making and improve diagnostic outcomes. The aim of this study is to address these challenges by developing an automated PCG signal classification system based on advanced signal processing and an AI driven framework. By integrating features from both the time and frequency domains, with robust classification algorithms. The proposed approach contributes to more efficient, accessible, and explainable cardiac diagnostics.

2.1 Heart anatomy

The heart is a vital organ, responsible for circulating blood throughout the body. It serves as the central component of the circulatory system and is composed of muscle and connective tissue. This muscular organ has four chambers that temporarily store blood before pushing it onward. The heartbeat is triggered by electrical signals, which drive the movement of blood through these chambers. The brain and nervous system play a key role in regulating the heart's activity. [1] The heart is anchored in connective structures that firmly to both the lungs and the diaphragm. It is composed of muscular tissue known as the myocardium. Structurally, the heart is divided into four chambers: two upper chambers called atria and two lower chambers known as ventricles, as illustrated in Figure 1.1. The atria, Located at the superior aspect of the heart, are responsible for receiving deoxygenated blood from the veins [1]

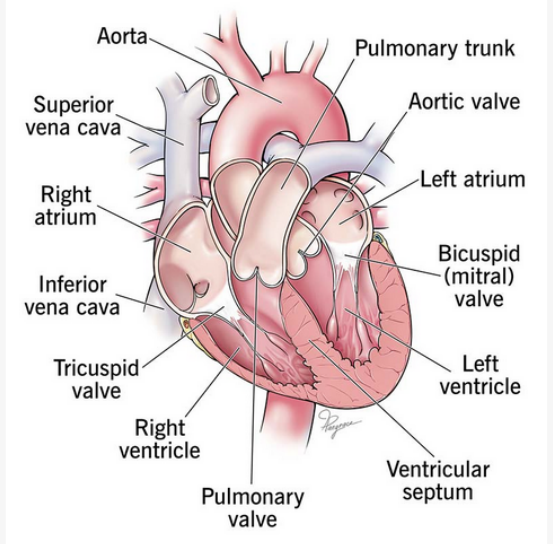


Figure 1.1: The heart anatomy [1]

2.2 The cardiac cycle

The heart is a vital organ that functions as a pump to maintain blood circulation throughout the body. With each beat, it propels blood through the arteries to supply the tissues. The process begins when the right atrium fills with oxygen-poor blood coming from various organs such as the muscles, the brain, and even the heart itself. Once the atrium is full, it contracts, causing the tricuspid valve to open and allowing the blood to flow into the right ventricle. As the right ventricle fills with blood, it prepares for its next role: pumping the blood to the lungs. When the ventricle contracts, the tricuspid valve closes to prevent backflow, and the pulmonary valve opens, allowing blood to be sent to the lungs for oxygenation via the pulmonary artery. After oxygenation, the left atrium receives oxygen-rich blood from the lungs through the pulmonary veins. The left atrium contracts, pushing blood through the mitral valve into the left ventricle. Finally, the left ventricle, which is the strongest chamber of the heart, contracts forcefully, sending the oxygenated blood through the aortic valve into the aorta and then to the rest of the body. This process, known as the cardiac cycle, ensures that oxygen and nutrients are delivered to tissues throughout the body, while waste products like carbon dioxide are carried away for elimination.As represented in Figure 1.2 .

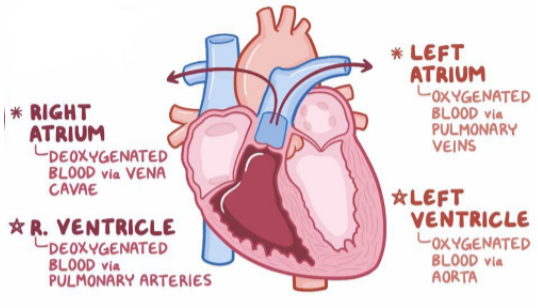


Figure 1.2: The four chambers of heart [1]

The heart consists of four distinct chambers: two atria at the top and two ventricles at the bottom, one on each side of the heart. Each chamber plays a specific role in the circulation of blood:

- **Right Atrium** : wo main veins transport oxygen-poor blood to the right atrium. The superior vena cava brings blood from the upper part of the body, while the inferior vena cava carries blood from the lower part. The right atrium then sends this blood to the right ventricle.
- **Right Ventricle** : The lower right chamber pumps deoxygenated blood to the lungs via the pulmonary artery, where it will be enriched with oxygen.
- **Left Atrium** : Once the lungs have oxygenated the blood, it is transported to the left atrium by the pulmonary veins. The left atrium then sends the blood to the left ventricle.
- **Left Ventricle** : The left ventricle is slightly larger than the right ventricle. It expels oxygen-rich blood throughout the body.

2.3 Cardiac Conduction System

The heart's electrical conduction system functions like the wiring of a complex circuit, ensuring that the heartbeat follows a regular rhythm and timing. Electrical impulses begin at the upper part of the heart and travel downward. The key components of this system include:[16]

- **Natural pacemaker (SA node)**: Initiates the electrical signal, setting the base rhythm for each heartbeat. [16]
- **Signal relay node (AV node)**: Delays the impulse briefly, allowing the upper chambers to complete their contraction before the lower chambers activate.[16]

– **Conduction pathway (Bundle fibers)**: Splits the signal into left and right branches, directing it to the respective lower chambers. [16]
– **Signal distribution network (Purkinje fibers)**: Rapidly disseminates the electrical impulse across the lower chambers, ensuring synchronized contraction.[16]

3 Heart sound diagnosis

To accurately diagnose heart conditions, doctors often analyze both the electrical and sound signals the heart produces. Electrocardiography (ECG) tracks electrical activity, while phonocardiography (PCG) records heart sounds and valve movements. Together, these noninvasive methods give a clearer picture of heart function, helping detect and classify heart problems more effectively. The next sections explain how ECG and PCG work and why they're important in heart exams

3.1 Electrocardiogram (ECG) Signals

An electrocardiogram (ECG) is a simple and noninvasive method for recording the electrical activity generated by the heart as it beats. To collect an ECG, small adhesive sensors are placed on the skin at specific points on the chest, arms, and legs. These sensors are connected by wires to a recording device, which captures the heart's electrical signals and displays them as a waveform on a screen..[17] The ECG waveform provides valuable information about the timing and pattern of each heartbeat. By examining the shapes and intervals of the waves, it is possible to detect irregularities in the heart's rhythm and identify unusual patterns that may indicate underlying issues. Different sections of the ECG signal correspond to different parts of the heart and can reveal whether the electrical activity is spreading normally. In the context of automated analysis, ECG signals are commonly used as input data for algorithms that classify heartbeats as normal or abnormal. The clear structure of the ECG waveform makes it suitable for feature extraction and pattern recognition, which are essential steps in developing reliable classification systems. Ten electrodes are used: six are positioned at specific locations across the chest (V1–V6), and four are placed on the limbs (right arm, left arm, right leg, left leg). Correct placement of these electrodes is essential to ensure accurate recording and analysis of the heart's electrical signals, as illustrated in Figure 1.3.

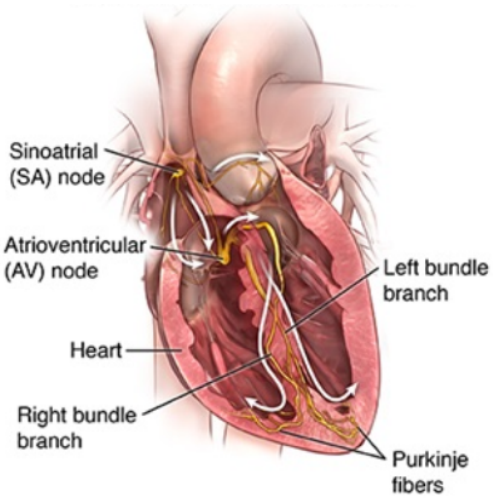


Figure 1.3: Electrical system of the heart [2]

3.2 Phonocardiogram (PCG) Signals

A phonocardiogram (PCG) is a digital recording of the sounds produced by the heart during its activity. These sounds mainly result from the opening and closing of heart valves and the flow of blood through the heart. To capture a PCG, a sensitive microphone or a contact-type sensor is placed on the chest, converting the acoustic vibrations into electronic signals that can be visualized and analyzed as waveforms.[18],As presented in Figure 1.4

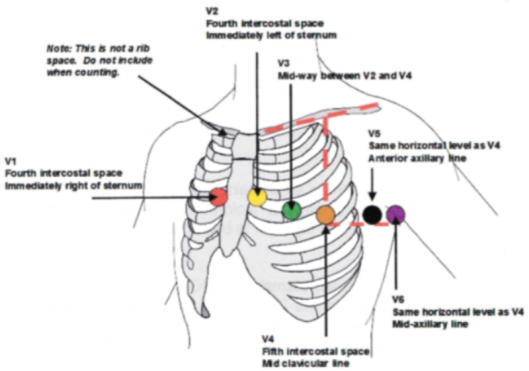


Figure 1.4: Standard ECG chest electrode positions

Several types of sensors and devices are used for PCG signal acquisition:

- **Electronic stethoscopes**: These are modern stethoscopes equipped with

built-in microphones or piezoelectric sensors to detect heart sounds and transmit them as electrical signals for further processing [16]

- **Electret microphones and piezoelectric sensors**: These are commonly used due to their high sensitivity and ability to capture low-frequency heart sounds. Some systems use air-filled tubes or 3D-printed sensor housings to enhance signal quality. [16]
- **Wearable and portable devices**: Recent advances have led to the development of wearable systems that integrate PCG sensors with ECG electrodes, enabling real-time, synchronized recording of both signals for comprehensive cardiac monitoring. [16]

3.2.1 Main Elements of Heart Sounds

PCG signals provide valuable information about the timing and quality of heart sounds, which are closely linked to the mechanical events of the cardiac cycle. The main heart sounds, known as S1 and S2, correspond to specific phases:

- **S1 (first sound)**: Occurs when the valves between the upper and lower chambers of the heart close at the start of contraction. This sound is typically low in frequency and has a deep tone.[16]
- **S2 (second sound)**: Happens when the valves leading out of the heart close as the heart relaxes. This sound is usually shorter and higher in pitch compared to S1.[16]

Occasionally, additional sounds (S3 and S4) may be present, which can provide further insight into heart function. PCG signals are often analyzed together with ECG signals to improve the detection and classification of normal and abnormal heartbeats. For automated analysis, PCG signals are segmented into individual heartbeats, and features such as timing, amplitude, and frequency characteristics are extracted. These features are then used by machine learning models to classify heart sounds, supporting efficient and objective cardiac assessment.As clarified by Figure 1.5.

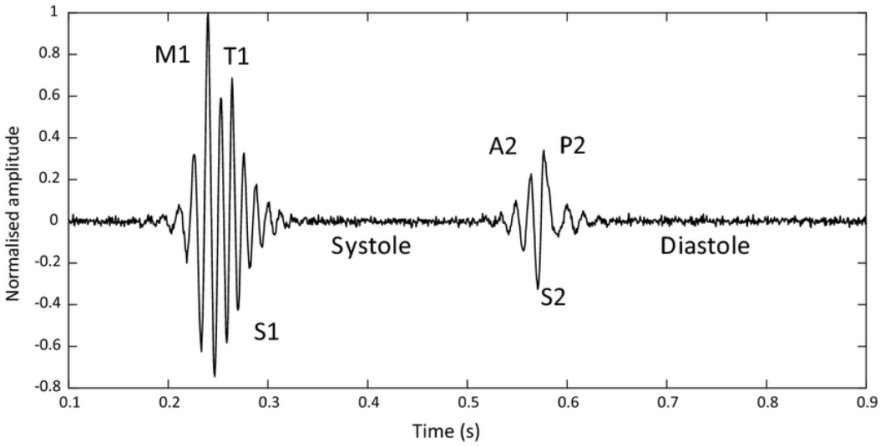


Figure 1.5: Normal phonocardiogram signal for a single cardiac cycle. [3]

3.2.2 Phases of the Cardiac Cycle in PCG Signals

Each heartbeat recorded in a phonocardiogram (PCG) signal is characterized by four distinct phases, which correspond to specific mechanical events of the cardiac cycle: the first heart sound (S1), systole, the second heart sound (S2), and diastole. Accurate identification and segmentation of these cardiac phases are essential for the analysis and interpretation of PCG signals in automated auscultation and computer-aided diagnostic systems. Figure 1.6 provides a detailed illustration of these phases, delineating their temporal sequence and clinical significance within the cardiac cycle. :[19]

S1 (First Heart Sound) The first heart sound (S1) marks the onset of ventricular systole and is produced by the closure of the mitral and tricuspid valves, which creates vibrations in the heart walls and blood. This sound is characterized by its low-frequency range (10–200 Hz), relatively long duration (100 ms), and higher amplitude compared to other heart sounds. In cardiac signal analysis, S1 plays a key role as a reference point for segmenting individual heartbeats in phonocardiograms (PCG). For improved accuracy, its detection is often synchronized with the R-peak of the ECG signal to ensure proper alignment between electrical and mechanical cardiac events.

Systole The period between the first (S1) and second (S2) heart sounds corresponds to ventricular systole, when the heart contracts to pump blood—the right ventricle sending blood to the lungs and the left ventricle supplying the body. Acoustically, this phase is normally quieter than S1 and S2 but may include murmurs (abnormal sounds) in cases of heart disease. These murmurs, when

present, often point to underlying conditions such as valve stenosis or regurgitation, making their detection clinically valuable for diagnosis.

S2 (Second Heart Sound) The second heart sound (S2) signals the end of systole and beginning of diastole, caused by the closure of the aortic and pulmonary valves. It is characterized by a higher frequency range (20–250 Hz) and shorter duration (60 ms) compared to S1. During inspiration, S2 often splits into two audible components—A2 (aortic valve closure) and P2 (pulmonary valve closure)—which can reflect normal respiratory variation or potential cardiovascular abnormalities. In cardiac signal analysis, S2 serves as a critical marker for distinguishing between systolic and diastolic phases, with its splitting pattern providing valuable diagnostic clues.

Diastole The period following the second heart sound (S2) until the next S1 marks ventricular diastole, when the heart relaxes and refills with blood from the atria. While this phase is typically quiet in healthy individuals, abnormal sounds like S3 (associated with rapid ventricular filling) or S4 (linked to atrial contraction into stiff ventricles) may appear. Diastolic murmurs—such as those caused by aortic regurgitation or mitral stenosis—are particularly significant, as they often indicate underlying valve dysfunction or structural heart disease. Detecting these sounds enhances the diagnostic precision of cardiac assessments.

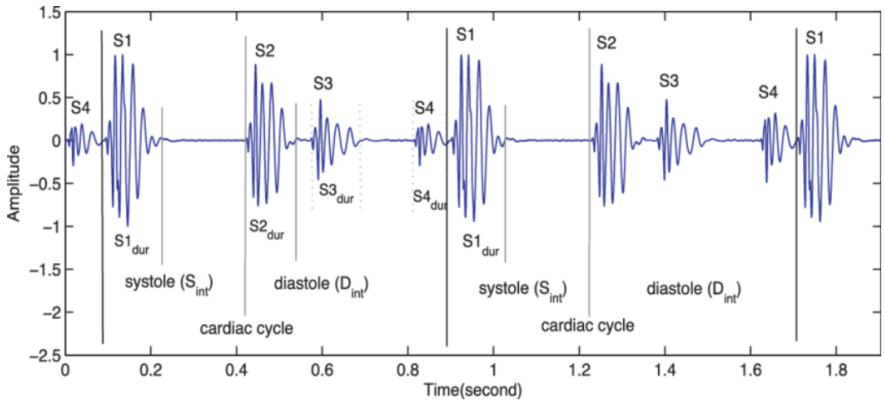


Figure 1.6: the PCG signal including heart sounds (S1, S2, systole,diastole). [4]

4 Cardiac Diseases Using PCG Signals

Cardiovascular diseases (CVDs) are globally prevalent and represent a critical focus for AI-driven diagnostic innovations, particularly in phonocardiogram (PCG)

signal classification. The following conditions are central to this study, as they produce distinct acoustic patterns in PCG recordings that vision transformer (ViT) models can detect and classify:[20]

- **Arrhythmias**: Arrhythmias disrupt the heart's electrical conduction system, causing irregular rhythms (tachycardia, bradycardia, or fibrillation). These irregularities alter the timing and periodicity of heart sounds (S1, S2) in PCG signals, providing measurable features for AI models to identify rhythm abnormalities.[20]
- **Cardiomyopathies**:Structural changes in the heart muscle (thickening, dilation, or stiffening) modify blood flow dynamics, often producing low-frequency murmurs or attenuated heart sounds. These acoustic shifts are detectable in PCG signals and serve as biomarkers for automated classification.[20]
- **Congestive Heart Failure (CHF)**:CHF reduces cardiac output, leading to fluid retention and abnormal heart sounds such as ventricular gallop or atrial gallop. These diastolic sounds, combined with prolonged systolic intervals, create identifiable patterns in PCG signals.[20]
- **Coronary Artery Disease (CAD)**: Plaque buildup in coronary arteries restricts blood flow, potentially causing ischemic murmurs or subtle changes in systolic duration. While CAD often manifests indirectly in PCG, advanced AI models can correlate these temporal features with ECG data for early detection.[20]
- **Diabetes-Related Cardiac Dysfunction**: Chronic high blood glucose can damage cardiac tissue and blood vessels, increasing the risk of developing various heart diseases. While not a direct heart sound abnormality, diabetes is a major risk factor for other detectable cardiac conditions.[20]
- **Myocardial Infarction (Heart Attack)**: Blockage of a coronary artery deprives part of the heart muscle of oxygen, potentially resulting in new or intensified murmurs or abnormal heart sounds that can be captured by PCG[20]
- **Heart Valve Diseases**: Malfunctioning valves—whether stenotic (narrowed) or regurgitant (leaky)—produce characteristic murmurs and altered heart sounds. These acoustic features are central to PCG-based diagnostics and are a primary focus for AI-driven classification systems.[20]
- **Hypertension (High Blood Pressure)**: Elevated blood pressure increases the workload on the heart and can lead to structural changes, sometimes producing abnormal heart sounds or accentuated components in the PCG.[20]

5 Objectives and Contributions

Automated classification of PCG signals is a rapidly evolving field, driven by the need for objective, accessible, and accurate cardiac diagnostics. By integrating advanced signal processing with state-of-the-art machine learning, this master thesis aims to address the limitations of traditional auscultation and provide a scalable solution for heart sound analysis.

5.1 Research Objectives

The primary objective of this master thesis is to develop and validate a robust, patient-specific framework for automated phonocardiogram (PCG) signal classification, distinguishing normal and abnormal cardiac conditions through advanced signal processing and hybrid machine learning architectures. The work is structured around the following objectives:

- Preprocessing and Segmentation: Develop preprocessing pipelines for raw PCG data, including noise reduction and beat-level segmentation synchronized with cardiac cycles (S1, systole, S2, diastole), leveraging ECG alignment for improved temporal accuracy.
- Multimodal Feature Extraction: Extract time-domain features (amplitude, skewness, kurtosis, phase durations) and spectral descriptors such as Mel Frequency Cepstral Coefficients (MFCCs) to capture both morphological and acoustic characteristics of heart sounds.
- Attention-Driven Beat Selection: Propose a novel attention-based module to dynamically prioritize diagnostically relevant beats in variable-length PCG signals, mitigating challenges posed by inter-patient variability and non-stationary noise.
- Hybrid Classification Architecture: Implement a dual-stage framework combining recurrent neural networks (RNNs) for adaptive feature extraction with ensemble classifiers (e.g., Random Forest, SVM) for final decision-making, enhancing robustness against signal artifacts.

5.2 Key Contributions

This thesis advances PCG-based cardiac diagnostics through the following innovations:

- Interpretable Beat Selection Framework: A dynamically weighted attention mechanism that identifies pathological segments while providing explainable visualizations of model focus areas, addressing the "black-box" limitation of conventional deep learning approaches.
- LSTM-Ensemble Hybrid Model: Integration of bidirectional RNNs for temporal context modeling with classical machine learning classifiers,
- Multimodal Feature Integration: Fusion of handcrafted time-frequency features with learned deep representations, enabling comprehensive characterization of murmurs, split sounds, and other acoustic anomalies critical for patient-specific diagnosis.

This work bridges the gap between deep learning's representational power and traditional machine learning's interpretability by proposing a hybrid architecture for automated cardiac classification. The framework combines attention-based feature selection with bidirectional LSTM networks and ensemble classifiers, offering a clinically viable solution for resource-constrained settings. The methodology aligns with emerging trends in explainable AI (XAI) for healthcare, as demonstrated through comparative analyses against baseline models, including adaptive feature extraction process and LSTM-ML hybrid architectures.

6 Conclusion

This chapter has provided an overview of the fundamental aspects of the cardiovascular system, including its anatomical structure, prevalent cardiac diseases, and the principles underlying phonocardiography and heart sound analysis. These foundational concepts are essential for the accurate interpretation and processing of PCG signals in both clinical and computational contexts. Phonocardiography, as a noninvasive diagnostic modality, offers valuable insights into cardiac function and complements traditional tools such as the electrocardiogram (ECG) by capturing acoustic information that may reveal subtle pathological changes. With the increasing complexity and volume of biomedical data, there is a growing need for automated, objective, and scalable approaches to cardiac diagnostics. The next chapter will explore artificial intelligence techniques—particularly those leveraging advanced machine learning and deep learning architectures—for the automatic identification of cardiac abnormalities from PCG signals. Integrating these AI-driven methods with established biomedical knowledge promises to enhance diagnostic accuracy, efficiency, and accessibility, ultimately contributing to improved cardiovascular care.

Chapter 2

Approaches for PCG Signal Classification

1 Introduction

Cardiovascular diseases (CVDs) remain a leading cause of global mortality, underscoring the urgent need for accurate and accessible diagnostic tools. Among non-invasive techniques, phonocardiography (PCG) offers a cost-effective means of detecting heart abnormalities by analyzing acoustic signals from cardiac activity. However, interpreting PCG signals is complex due to their inherent variability, susceptibility to noise, and the subtle distinctions between pathological and normal heart sounds. Traditional diagnostic approaches rely on manual auscultation or rule-based algorithms, which are often subjective, time-consuming, and limited in scalability. Recent advancements in artificial intelligence (AI), particularly deep learning, have opened new frontiers in biomedical signal processing. Vision transformers (ViTs) and attention mechanisms, originally developed for natural language and image processing, present a promising solution. By dynamically weighting the importance of different signal segments, these techniques can enhance both the accuracy and explainability of PCG analysis.

This chapter systematically examines the theoretical foundations, existing methodologies, and research gaps in PCG signal classification. First, it explores the role of AI in biomedical signal processing, highlighting key challenges and opportunities. Next, it provides an in-depth analysis of PCG signals, and the limitations of conventional analysis techniques. A critical review of related work , comparing traditional methods, deep learning approaches, and hybrid models.[21]

By synthesizing these elements, this chapter establishes the rationale for the proposed AI-driven framework, which seeks to overcome existing limitations through innovative beat selection, optimized feature representation, and transparent decision-

making—paving the way for more reliable cardiac diagnostics.[21]

2 Contribution and methodology for beat sounds classification

This section outlines the core principles and approaches that underpin our proposed system for heart beat sound classification. It provides a structured overview of the tech- niques and tools employed to address the challenges associated with phonocardio- gram (PCG) signal analysis. Our methodology is designed to ensure the extraction of meaningful information from heart sound recordings and to enable robust classi- fication of heartbeats.

first We begin by presenting the preprocessing steps required to handle real-world PCG data. These include denoising, normalization, and segmentation procedures tailored to manage signals of varying lengths and recording conditions. Next, we explore the feature extraction strategies employed in our system. Specif- ically, we focus on the computation of temporal features and frequency features. Additionally, to enhance the system's ability to capture the most relevant patterns, we integrate machine learning and deep learning components such as LSTM networks equipped with attention mechanisms. This allows the model to dynamically identify and weigh the most informative segments within each patient's heart sound signal.

we descuss in this model of classification in Machine learning .

This figure 2.1 ,illustrates the standard workflow for a machine learning classification task. It starts with collecting a dataset, which is then cleaned and used for feature extraction. The processed data is split into training and test sets. A learning algorithm is chosen and applied to the training data to build the model. The trained model is then used to make predictions (score) on the test data, and finally, its performance is evaluated to determine its accuracy and effectiveness.

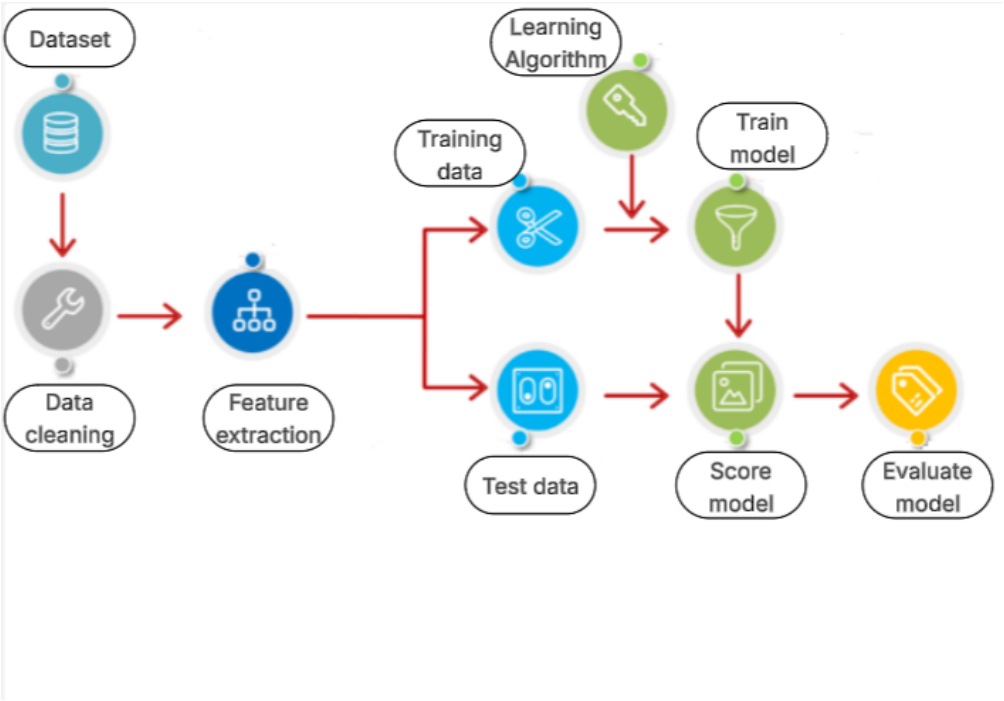


Figure 2.1: Machine learning process diagram

2.1 Feature Extraction for PCG Signals

Feature extraction plays a crucial role in the analysis of biomedical signals, especially phonocardiogram (PCG) signals, which contain rich temporal and spectral information. In particular, frequency-based feature extraction methods allow for the transformation of raw time-domain signals into representations that highlight their spectral properties. This is especially relevant for PCG data, where the characterization of heart sound patterns, murmurs, and anomalies is often better captured in the frequency domain and temporal domain. Techniques such as Mel-Frequency Cepstral Coefficients (MFCCs), Short-Time Fourier Transform (STFT), and Mel-scaled spectrograms are commonly used to encode time-frequency dynamics in a compact and discriminative manner. These features enhance the ability of machine learning and deep learning models to detect subtle variations in cardiac cycles, ultimately improving the accuracy of classification and diagnosis tasks [22] .

2.1.1 Temporal Features

Temporal features broadly refer to data attributes that vary over time or are linked to the temporal sequence of events. These features capture the dynamic,

time-dependent patterns within data and are widely utilized in fields such as signal processing, computer vision, as presented in table 2.1 .

Table 2.1: Extracted Temporal Features : Basic Signal Characteristics

#	**Feature**	**Description / Formula**
1	Mean S1	$\mu_{S1} = \frac{1}{N}\sum_{i=1}^{N} x_i$; Average amplitude of S1
2	Mean S2	$\mu_{S2} = \frac{1}{N}\sum_{i=1}^{N} x_i$; Average amplitude of S2
3	Mean Systole	$\mu_{Sys} = \frac{1}{N}\sum_{i=1}^{N} x_i$; Mean amplitude during systole
4	Mean Diastole	$\mu_{Dias} = \frac{1}{N}\sum_{i=1}^{N} x_i$;Mean amplitude during diastole
5	S1 Duration	$T_{S1} = \frac{F_{S1}-D_{S1}}{f_s}$; S1 segment duration
6	S2 Duration	$T_{S2} = \frac{F_{S2}-D_{S2}}{f_s}$; S2 segment duration
7	Systolic Duration	$T_{Sys} = \frac{F_{S2}-F_{S1}}{f_s}$; S1 end to S2 time
8	Diastolic Duration	$T_{Dias} = \frac{F_{\text{next_S1}}-F_{S2}}{f_s}$; S2 to next S1 time
9	Peak S1 Amplitude	$A_{S1} = \max(x_{S1})$; Maximum S1 amplitude
10	Peak S2 Amplitude	$A_{S2} = \max(x_{S2})$; Maximum S2 amplitude
11	S1 Skewness	$\gamma_{S1} = \frac{\mathbb{E}[(x-\mu)^3]}{\sigma^3}$; S1 distribution asymmetry
12	S2 Skewness	$\gamma_{S2} = \frac{\mathbb{E}[(x-\mu)^3]}{\sigma^3}$; S2 distribution asymmetry
9	Peak S1 Amplitude	$A_{S1} = \max(x_{S1})$; Maximum S1 amplitude
10	Peak S2 Amplitude	$A_{S2} = \max(x_{S2})$; Maximum S2 amplitude
11	S1 Skewness	$\gamma_{S1} = \frac{\mathbb{E}[(x-\mu)^3]}{\sigma^3}$; S1 distribution asymmetry
12	S2 Skewness	$\gamma_{S2} = \frac{\mathbb{E}[(x-\mu)^3]}{\sigma^3}$; S2 distribution asymmetry
13	S1 Kurtosis	$\kappa_{S1} = \frac{\mathbb{E}[(x-\mu)^4]}{\sigma^4} - 3$; S1 peakedness
14	S2 Kurtosis	$\kappa_{S2} = \frac{\mathbb{E}[(x-\mu)^4]}{\sigma^4} - 3$; S2 peakedness

Figures 2.2, 2.3, and 2.4 illustrate the peak amplitude, skewness, and kurtosis of the extracted temporal features, respectively. These figures provide a comprehensive overview of the statistical characteristics of the analyzed signals, thereby facilitating a deeper understanding of their underlying temporal dynamics.

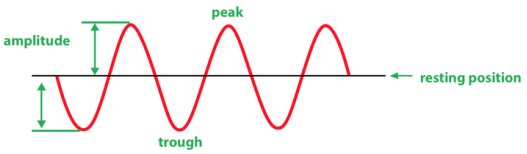


Figure 2.2: Peak Amplitude curve [5]

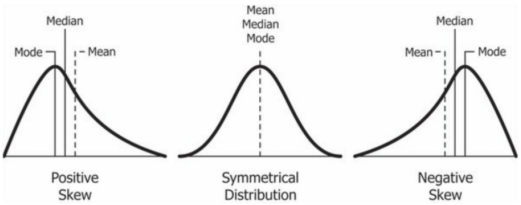


Figure 2.3: Skewness curve [6]

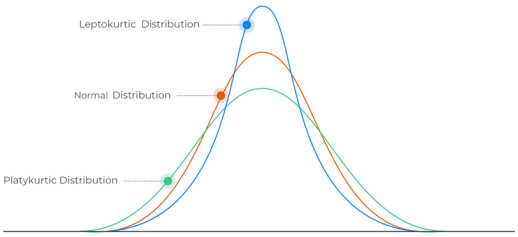


Figure 2.4: Kurtosis curve

2.1.2 Frequency Features

Mel-Frequency Cepstral Coefficients, is a feature extraction technique used in audio and speech processing. It's a set of coefficients that represent the spectral characteristics of a sound in a way that's well-suited for tasks like speech recognition and music analysis.

- MFCCs Mel Frequency Cepstral Coefficients extracted from each PCG segment. These coefficients represent the short-term power spectrum of the signal based on a non-linear mel scale of frequency. Each segment is represented by a sequence of 30 MFCCs, preserving the spectral evolution of the signal.

  This table 2.3, summarizes the main steps involved in the computation of Mel Frequency Cepstral Coefficients (MFCCs). The process starts by segmenting the audio signal into short overlapping frames, followed by applying

a window function to reduce edge effects. Each frame is then transformed into the frequency domain using the Fast Fourier Transform (FFT). The resulting spectrum is filtered through a Mel-scale filter bank that mimics human auditory perception. The logarithm of the filter energies is computed to approximate the non-linear response of the ear, and finally, a Discrete Cosine Transform (DCT) is applied to obtain compact, decorrelated cepstral coefficients. Typically, the first 12 or 13 coefficients are retained for further audio analysis tasks such as speech recognition or speaker identification.

Step	**Description**
Framing	The audio signal is segmented into short frames (20 to 40 ms) with partial overlap (usually 50%) to capture fine temporal variations.
Windowing	Each frame is multiplied by a Hamming window to reduce discontinuity effects at the edges of the frames.
Fast Fourier Transform (FFT)	The FFT of each frame is computed to obtain its power spectrum, representing the frequency distribution of the signal.
Mel Filter Bank Application	The spectrum is passed through a series of triangular filters spaced according to the Mel scale, which models human frequency perception (finer resolution at low frequencies, coarser at high frequencies).
Logarithm of Filtered Powers	The logarithm of the energies from each Mel filter is taken, corresponding to a scale closer to human auditory perception.
Discrete Cosine Transform (DCT)	The DCT is applied to the logarithmic values to obtain cepstral coefficients, which are compact and decorrelated spectral descriptors.
Coefficient Selection	Typically, the first 12 or 13 coefficients are retained (excluding the 0th coefficient representing overall energy), sometimes accompanied by their temporal derivatives (delta, delta-delta) to capture dynamics.

Table 2.2: Main steps in the computation of MFCCs

- STFT (Short-Time Fourier Transform) Short-Time Fourier Transform (STFT) represents the time-frequency content of the PCG signal. It provides a 2D matrix (time $\times$ frequency bins) that captures how the frequency compo-

nents of the signal evolve over time. It is useful for analyzing non-stationary segments.

3 The classification process

Classification is a supervised machine learning process where a model is trained to assign input data to predefined categories or labels. The model learns patterns from labeled training data and then predicts the class of new, unseen instances. This process is widely used in applications such as image recognition, speech analysis, and bioinformatics, where understanding the sequential or temporal nature of data can be crucial for accurate classification

3.1 Machine Learning classification Algorithms

A machine learning algorithm is a computational approach that allows a model to identify patterns and relationships within data to make predictions or decisions without explicit programming for the task. In supervised learning tasks like classification, the algorithm learns to associate input features with output labels from labeled training data. For example, Support Vector Machines (SVM), Random forest (RF). [23]

3.1.1 SVM support vector machine

Support Vector Machine (SVM) is a supervised classification algorithm commonly applied to classification and regression problems. It identifies the optimal hyperplane that separates data points of different classes by maximizing the margin—the greatest possible distance between the hyperplane and the closest data points from each class, called support vectors. Maximizing this margin enhances the model's ability to generalize to new, unseen data and helps prevent overfitting.

SVM is capable of handling both linearly separable and non-linearly separable datasets. In cases where data is not linearly separable, SVM employs kernel functions—such as linear, polynomial, radial basis function (RBF), or sigmoid kernels—to implicitly transform the data into higher-dimensional spaces where a linear separation can be achieved. [24], As clarified by Figure 2.5 .

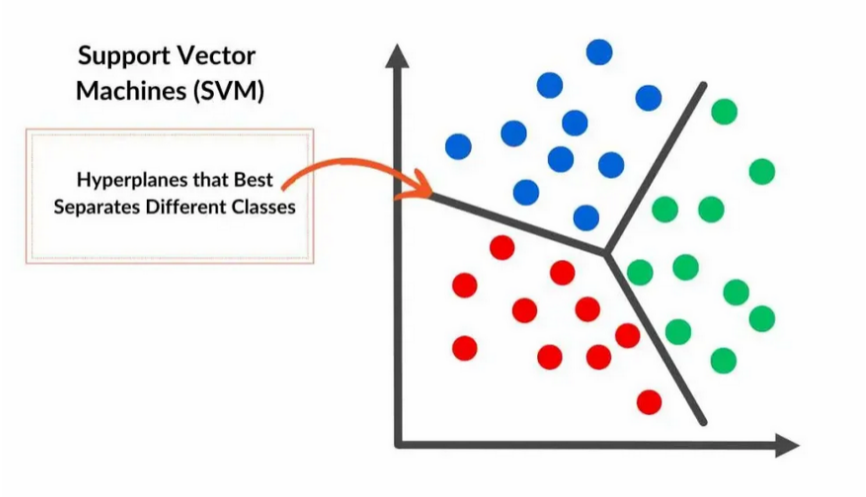


Figure 2.5: Architecture of SVM[7]

The main hyperparameters of a Support Vector Machine (SVM) classifier are:

- **gamma**: Used with non-linear kernels (such as RBF); it defines how far the influence of a single data point reaches. A high value can lead to overfitting.
- **degree**: Degree of the polynomial kernel function (*poly* kernel only).
- **coef0**: Independent term in kern

3.1.2 Decision Tree

A Decision Tree is a non-parametric supervised learning algorithm commonly used for both classification and regression tasks. It models decisions using a hierarchical tree structure composed of a root node, internal (decision) nodes, branches, and leaf (terminal) nodes. The root node represents the entire dataset, and at each internal node, the data is split based on feature values to create more homogeneous subsets. This recursive partitioning continues until the leaf nodes represent final outcomes or class labels.

The tree is built using a top-down, greedy approach that selects optimal split points to maximize homogeneity within subsets. Smaller trees tend to produce purer leaf nodes, but as the tree grows deeper, data fragmentation can occur, potentially leading to overfitting. Decision Trees are intuitive and easy to visualize, making them useful for understanding decision-making processes. However, they require careful tuning .[8] As clarified by Figure 2.6 .

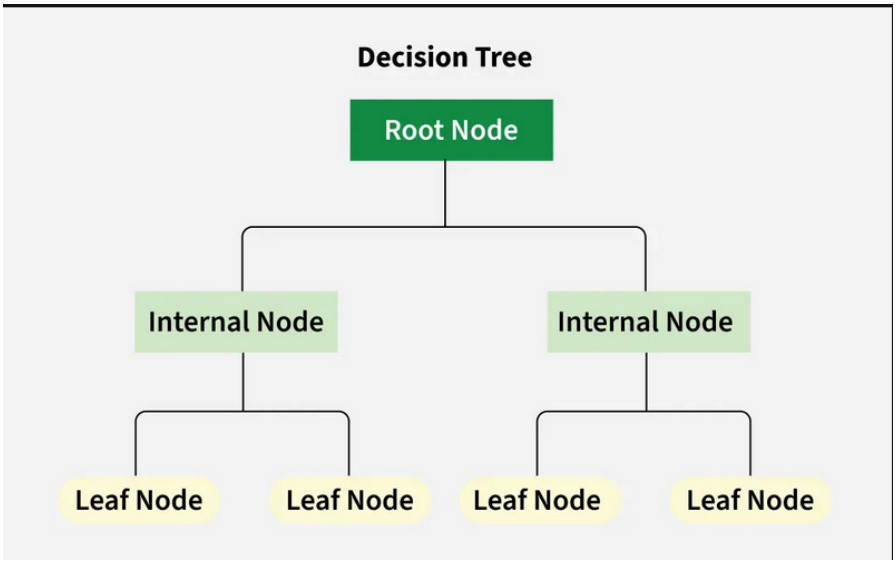


Figure 2.6: Architecture of Decision Tree [8]

- **Root Node:** Serves as the starting point, encompassing the full dataset.[8]
- **Branches:** Connect nodes and indicate the path from one decision to the next.[8]
- **Internal Nodes:** Mark points where choices are made based on the features of the input data.[8]
- **Leaf Nodes:** Represent the endpoints of branches and correspond to the final decisions or predictions.[8]

Decision trees rely on several important hyperparameters that influence their effectiveness:

- **Maximum Depth**: Defines how deep the tree can grow, limiting the number of splits to avoid overfitting.[8]
- **Minimum Samples for Split**: Specifies the least number of samples needed to divide a node, which controls tree complexity.[8]
- **Minimum Samples per Leaf**: Determines the smallest number of samples required at a leaf node to ensure stability in predictions.[8]
- **Maximum Features**: Indicates how many features to evaluate when searching for the best split at each node.[8]
- **Splitting Criterion**: The metric used to evaluate splits, such as "gini impurity" for classification or "mean squared error" for regression.[8]

3.1.3 Bagging Algorithms

Bagging, short for Bootstrap Aggregating, is an ensemble learning method designed to increase the accuracy and robustness of machine learning models. It works by creating multiple subsets of the original dataset through random sampling with replacement. Each subset is used to train an independent base learner—often decision trees. Once all models are trained, their predictions are combined, typically using majority voting for classification tasks or averaging for regression tasks. This approach reduces overfitting and variance by ensuring that the final prediction is less sensitive to fluctuations in the training data.

bagging (Bootstrap Aggregating), multiple subsets are created by randomly sampling the original training dataset with replacement, allowing the same data point to appear in more than one subset. Each subset is then used to train a separate, independent base learner. The final prediction is made by combining the outputs of all individual models—typically through averaging for regression tasks or majority voting for classification—resulting in improved accuracy and reduced variance.

[25] As clarified by Figure 2.7 .

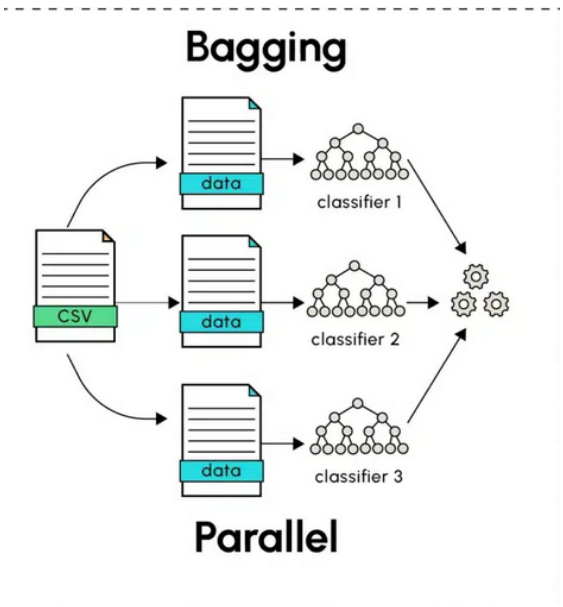


Figure 2.7: Architecture of Bagging [9]

Bagging Hyperparameters

Bagging models rely on several key hyperparameters that control the training process. These include:

- **n_estimators:** Number of base estimators in the ensemble.[25]

- **max_samples:** Number or proportion of training samples drawn for each base learner.[25]
- **max_features:** Number or proportion of features used for training.[25]
- **bootstrap:** Whether sampling is done with replacement.[25]
- **oob_score:** Whether out-of-bag error is computed for validation.[25]
- **base_estimator:** The type of model used[25]

a) Randim forest

Random Forest is a supervised machine learning algorithm commonly applied to both classification and regression problems. It builds an ensemble, or "forest," composed of multiple decision trees, each trained on randomly selected subsets of the training data and features. This randomness fosters diversity among the trees, which helps to reduce overfitting and enhances the model's ability to generalize to new data.

During training, each tree is constructed independently using a bootstrap sample of the dataset and a randomly chosen subset of features for node splitting. In classification tasks, the final prediction is determined by majority voting across all trees, whereas in regression tasks, it is computed as the average of the predictions from all individual trees.

[26] As clarified by Figure 2.8

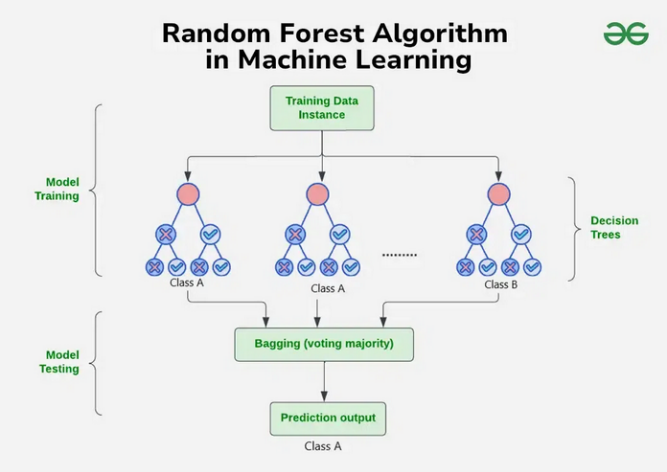


Figure 2.8: Architecture of Random forest [10]

The Random Forest algorithm has several important hyperparameters that influence its performance:

- **Number of trees (n_estimators)**: Specifies how many decision trees the forest will contain. A larger number usually improves performance but increases computation time.
- **Maximum depth (max_depth)**: Limits the depth of each tree. Shallow trees reduce overfitting but may lower accuracy.
- **Minimum samples per leaf (min_samples_leaf)**: Determines the minimum number of samples required to be at a leaf node. It helps prevent the model from learning noise in the data.
- **Maximum features (max_features)**: Sets the number of features to consider when looking for the best split at each node. Lower values increase diversity between trees.
- **Bootstrap sampling**: A boolean value that decides whether bootstrap samples are used when building trees. If set to `True`, each tree is trained on a random sample of the data with replacement.
- **Criterion**: Defines the function used to measure the quality of a split (e.g., `gini` or `entropy` for classification, `mse` for regression).

b) Balanced Random Forest

Balanced Random Forest (BRF) is an adaptation of the traditional Random Forest algorithm specifically designed to better handle imbalanced datasets. In a standard Random Forest, each decision tree is trained on a bootstrap sample randomly drawn from the entire dataset. This random sampling can cause poor performance when one class is significantly underrepresented. To address this issue, BRF creates, for each tree, two bootstrap samples of equal size: one from the minority class and one from the majority class. The size of these samples matches the number of instances in the minority class, thereby balancing the training data for each tree. By combining majority class undersampling with ensemble learning, BRF improves classification accuracy on imbalanced data.[27] , as clarified by Figure 2.9

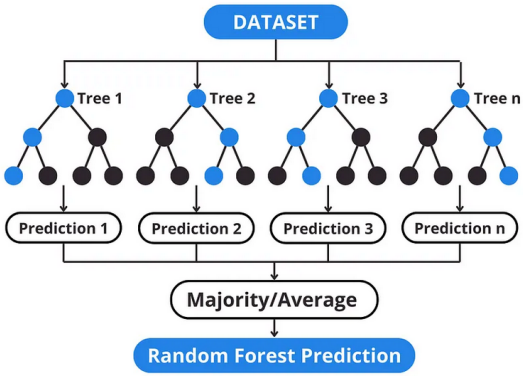


Figure 2.9: Architecture of BRF

3.1.4 Boosting Algorithm

Boosting is a machine learning technique that improves prediction accuracy by combining several simple models. These models are built one after another, with each new model focusing on correcting the mistakes made by the previous ones. By progressively reducing errors, the overall system becomes stronger and more accurate. This step-by-step process helps turn weak individual models into a powerful combined model.

a) Gradient boosting

Gradient boosting is a machine learning technique that enhances prediction accuracy by integrating several decision trees into one comprehensive model. Originally developed by Jerome Friedman, this method iteratively builds new models that focus on fixing the mistakes made by previous ones, thereby refining overall performance. By effectively capturing intricate relationships within the data, gradient boosting performs well across a wide range of prediction problems [11] as shown in Figure 2.10.

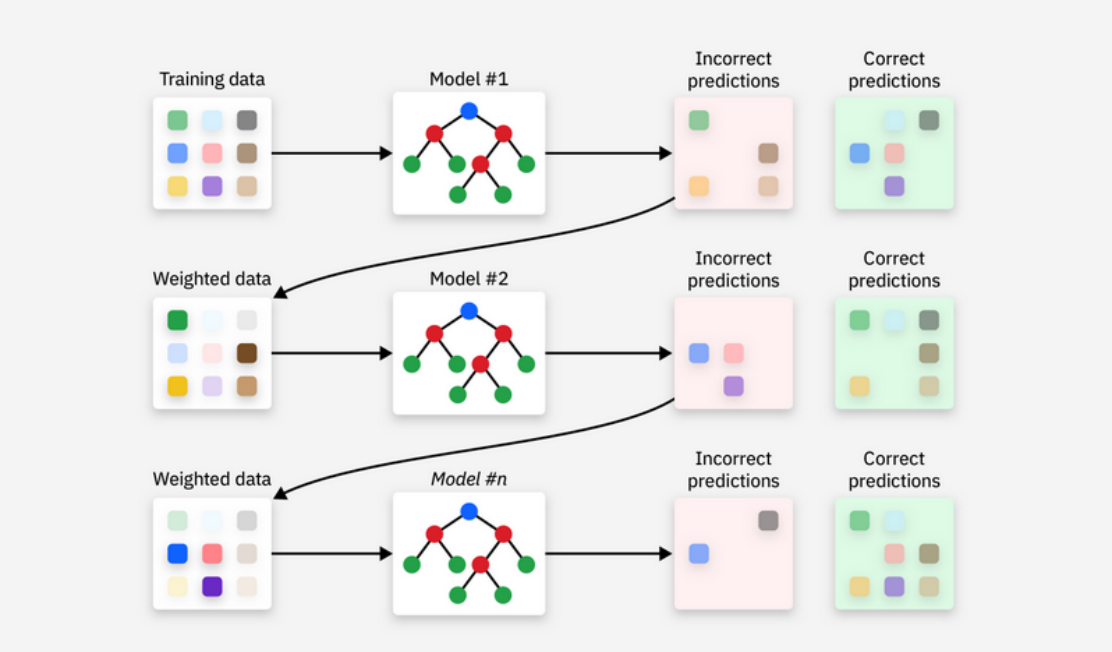


Figure 2.10: Architecture of Gradient boosting [11]

Gradient Boosting: Key Hyperparameters

- **n_estimators** : Nombre d'arbres (ou étapes) dans l'ensemble. Un plus grand nombre peut améliorer la performance mais augmente le temps d'entraînement.[11]
- **learning_rate** : Taux d'apprentissage. Il contrôle l'impact de chaque arbre sur le modèle final. Des valeurs plus faibles nécessitent plus d'estimations. [11]
- **max_depth** : Profondeur maximale des arbres de décision individuels. Contrôle la complexité du modèle. [11]
- **min_samples_split** : Nombre minimum d'échantillons requis pour diviser un nœud interne.[11]
- **min_samples_leaf** : Nombre minimum d'échantillons requis dans un nœud terminal (feuille).[11]
- **subsample** : Proportion des échantillons utilisés pour entraîner chaque arbre. Cela introduit de la diversité et peut aider à réduire le surapprentissage.[11]
- **max_features** : Nombre maximal de caractéristiques à considérer lors de la recherche de la meilleure division.[11]

b) XGBoost

XGBoost (Extreme Gradient Boosting) is a highly efficient and scalable open-source machine learning algorithm built upon the gradient boosting framework. It constructs an ensemble of weak learners—usually decision trees—in a sequential process where each new tree aims to correct the errors made by the previous ones. XGBoost improves upon traditional gradient boosting by integrating advanced regularization methods, such as L1 and L2 regularization, to reduce overfitting, and by optimizing computational performance for faster training and prediction. These enhancements make XGBoost particularly well-suited for supervised learning tasks including classification, regression, and ranking, especially when dealing with large, high-dimensional structured datasets. Its combination of speed, accuracy, scalability, and flexibility has contributed to its widespread adoption in data science competitions and practical applications.[28] ,we explain this in figure 2.11.

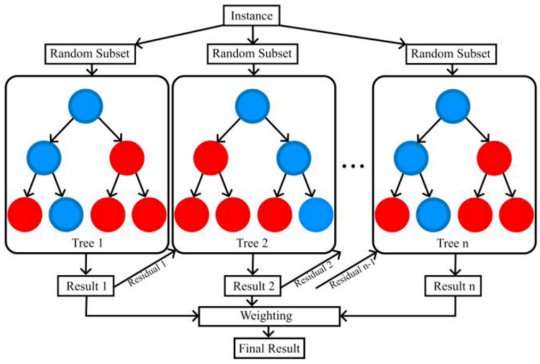


Figure 2.11: Architecture of of XGBoost [12]

3.2 Deep Learning classification architecture

Deep learning is a branch of machine learning that employs multilayered artificial neural networks, to capture and model complex patterns and decision-making processes inspired by the human brain.

These models can handle both structured and unstructured data, automatically learning relevant features ,often the need for features extraction process. Deep learning models excel at automatically extracting meaningful features from large volumes of unstructured data. This strength allows them to surpass traditional machine learning methods in tasks that require understanding high-dimensional inputs. As a result, deep learning has become a cornerstone technology behind breakthroughs in fields like medical image analysis, real-time language translation, autonomous robotics, and creative AI generation..[29]

3.2.1 Convolutional Neural Networks CNNs

The Convolutional Neural Networks (CNNs) are a specialized deep learning model primarily developed for processing and analyzing visual data such as images and videos. CNNs automatically learn hierarchical features by applying convolutional filters (kernels) that slide over the input data to detect local patterns like edges, textures, and shapes. These filters share weights across spatial dimensions, allowing the network to identify features regardless of their location, which provides translation invariance.

The typical CNN architecture is composed of multiple layers, including convolutional layers that extract features, pooling layers that reduce spatial dimensions and enhance robustness to spatial variations, and fully connected layers that perform classification or regression based on the extracted features. By stacking these layers, CNNs progressively learn more complex and abstract representations of the input data. [30] As shown in Figure 2.12.

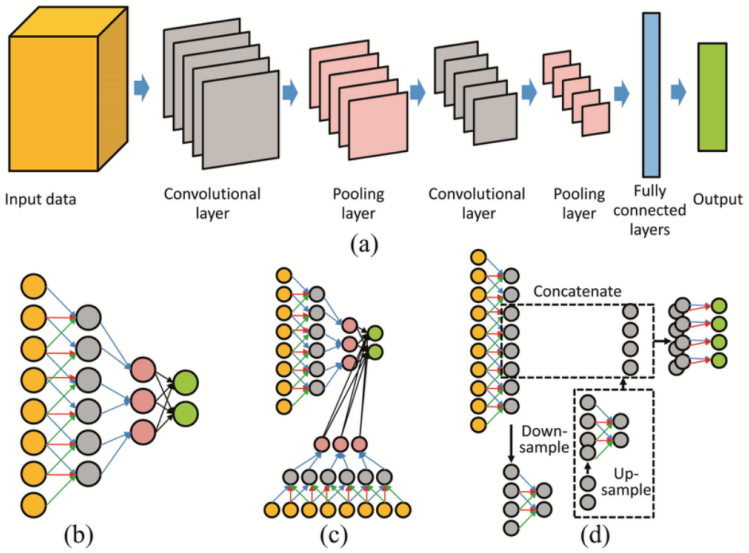


Figure 2.12: CNN Architectures of CNN [**?**]

Hyperparameters of CNN

- **Learning rate:** Controls how much the model weights are updated during training.
- **Batch size:** Number of training samples processed before updating the model.

- **Number of epochs:** How many times the entire training dataset is passed through the network.
- **Dropout rate:** Percentage of neurons randomly dropped to prevent overfitting.
- **Optimizer:** Algorithm used to update the weights, Adam, SGD.

3.2.2 Recurrent Neural Networks RNNs

The Recurrent Neural Networks (RNNs) are a deep learning model specifically designed to handle sequential data by retaining an internal memory of previous inputs. Unlike traditional neural networks, RNNs feature recurrent connections that enable information to persist across time steps, allowing the network to capture temporal dependencies and contextual information within sequences such as text, speech, or time series data. At each time step, an RNN processes the current input together with the hidden state (memory) from the previous step to generate an output and update its internal state. This recurrent mechanism allows the model to incorporate past information into its predictions, making it particularly effective for tasks like speech recognition, language modeling, and machine translation. [31], As shown in Figure2.13.

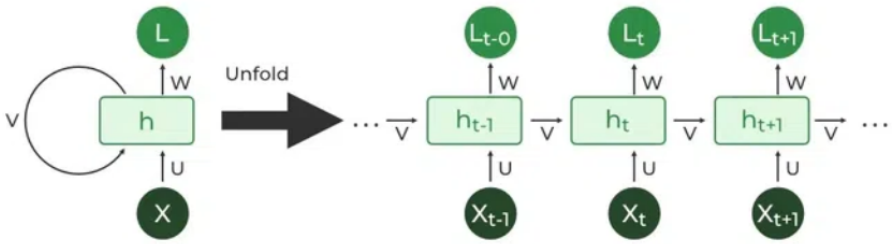


Figure 2.13: Architecture of RNN [13]

Hyperparameters of RNN

- **Number of hidden layers:** The total layers of recurrent units stacked.
- **Number of units per layer:** The number of neurons in each recurrent layer.

- **Sequence length (timesteps):** The length of input sequences processed at once.
- **Batch size:** Number of sequences processed before updating model weights.
- **Learning rate:** Controls the step size during weight updates.
- **Number of epochs:** Number of passes through the entire training dataset.
- **Dropout rate:** Fraction of neurons randomly ignored to reduce overfitting.
- **Activation function:** Non-linear function applied inside the RNN units (e.g., tanh, ReLU).
- **Optimizer:** Algorithm used to update weights, such as Adam or RMSprop.
- **Gradient clipping threshold:** Prevents exploding gradients by capping gradients.

3.2.3 Long Short-Term Memory LSTM

Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural network (RNN) commonly utilized in deep learning. They are particularly effective at learning and remembering information over long sequences, which makes them well-suited for tasks involving sequence prediction. Unlike standard neural networks, LSTMs feature feedback loops that allow them to consider the full sequence of input data rather than isolated points. This capability enables LSTMs to identify and forecast patterns in sequential datasets such as time series, natural language, and audio signals. Their ability to capture long-range dependencies has made LSTMs a key technique in AI, driving significant advances across many applications by extracting meaningful insights from sequential information. [32] As shown in Figure 2.13.

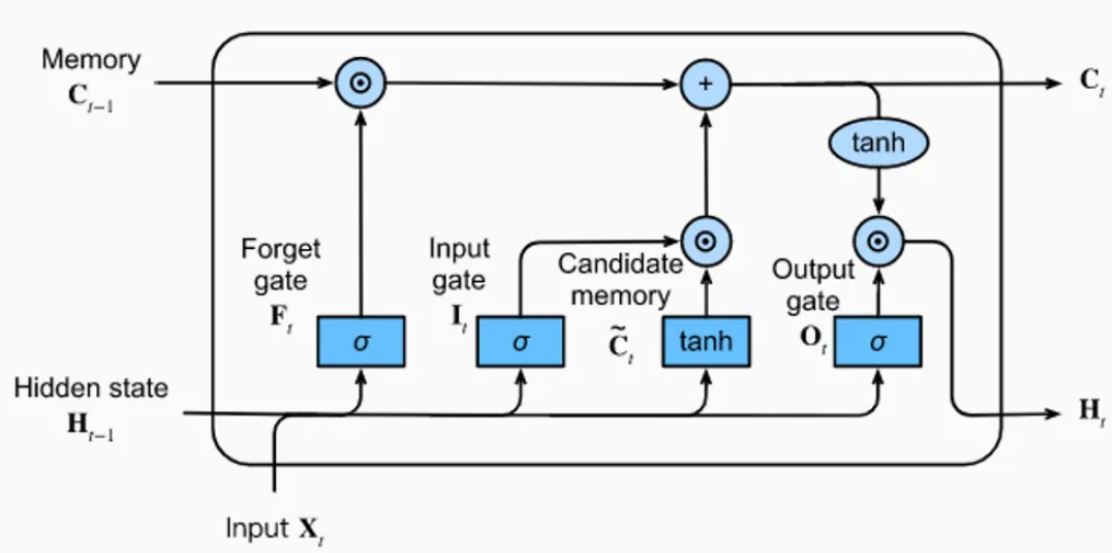


Figure 2.14: Architecture of LSTM [14]

1. Bidirectional LSTM Architecture

   A Bidirectional Long Short-Term Memory (BiLSTM) network is an advanced type of recurrent neural network designed to process sequential data by capturing information from both past and future contexts. Unlike traditional LSTMs that analyze sequences in a single forward direction, BiLSTMs consist of two parallel LSTM layers: one processes the input sequence from start to end (forward pass), while the other processes it from end to start (backward pass). The outputs of these two layers are then combined—commonly by concatenation—to form a richer representation at each time step, enabling the model to better understand dependencies in the data.[15] We explain this in figure 2.15.

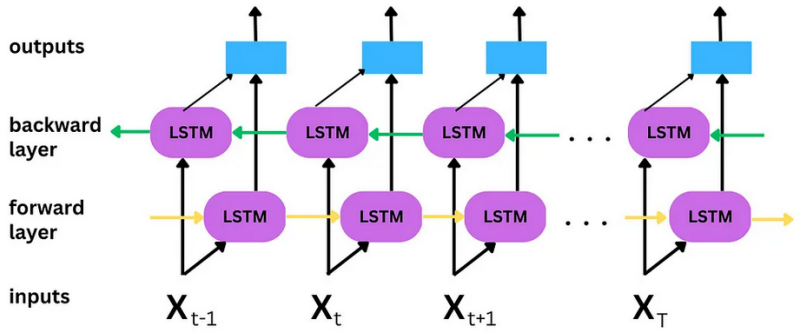


Figure 2.15: Architecture of BiLSTM [15]

- Number of LSTM units per layer, which controls the network's capacity to learn complex patterns.
- Sequence length or input size, defining how many time steps are considered.
- Choice of output combination method (concatenation, summation, or averaging).
- Training parameters such as batch size, learning rate, and number of epochs.
- Embedding dimensions, important for textual data as they determine the size of vector representations for input tokens.
- Proper tuning of these hyperparameters is essential to balance model performance with computational efficiency, especially since BiLSTMs typically require more resources than unidirectional models.

4 Datasets for beat sounds classification

Data set the foundational element of any scientific or technical analysis., data can originate from various sources such as physiological signal recordings (ECG, PCG), medical imaging, or clinical databases. The quality of the collected data directly impacts the reliability of the results obtained. Therefore, data preprocessing—including cleaning, normalization, and temporal alignment—is a crucial step before any modeling or feature extraction. Moreover, the volume, diversity, and complexity of data often require the use of advanced analytical methods, such as machine learning, to extract meaningful and actionable information [33]

4.1 PhysioNet/CinC Challenge 2016 Dataset

The PhysioNet/Computing in Cardiology Challenge 2016 dataset is one of the most widely used benchmarks for heart sound classification. It contains a total of 3,153 phonocardiogram (PCG) recordings collected from 764 patients across a range of clinical environments and devices. The recordings were sourced from multiple institutions, including primary care centers, hospitals, and rural clinics, making the dataset highly heterogeneous.

Each recording is between **5 and 120 seconds** long and was manually labeled as either **normal** or **abnormal** by expert clinicians. The labels reflect the presence or absence of cardiovascular pathology, but not the exact type of pathology. The dataset includes noisy and clean recordings, sampled at different rates (mostly 2,000 Hz), adding realism and complexity.

The main challenge in CinC 2016 was to develop algorithms that can classify a heart sound as normal or abnormal under real-world conditions. This dataset established a benchmark for evaluating machine learning models (e.g., CNN, LSTM, SVM) on heart sound classification.[22]

4.2 PhysioNet/CinC Challenge 2022 Dataset

The PhysioNet/Computing in Cardiology Challenge 2022 dataset focused on the **multilabel classification of heart murmurs and clinical outcomes** from PCG recordings. The dataset, known as **CirCor DigiScope**, contains over **5,000 PCG recordings** from approximately **1,568 pediatric patients** in Brazil.

Each patient has recordings from up to four auscultation locations (aortic, pulmonary, tricuspid, mitral), and each recording is labeled with:

- The presence of **heart murmurs** (present, absent, unknown)
- The patient's **clinical outcome** (normal or abnormal)

Recordings are sampled at **4,000 Hz**, and many contain **background noise and variable quality**, simulating a realistic auscultation setting. Additional metadata is provided, such as patient age, sex, height, weight, and auscultation location. The goal of the challenge was to develop models for early cardiac screening, especially in low-resource settings. [22]

4.3 PASCAL Classifying Heart Sounds Challenge Dataset

The PASCAL Classifying Heart Sounds Challenge dataset (2009) was an earlier effort to standardize heart sound analysis. It includes a small but well-annotated set of **normal and abnormal PCG recordings**, mostly recorded under controlled conditions.

Each recording is a few seconds long, and the dataset is manually labeled by cardiologists. Though limited in size, this dataset was valuable for early research on heart sound segmentation and classification using traditional signal processing techniques (Fourier transform, wavelet transform) and early machine learning methods.

Compared to CinC datasets, PASCAL's dataset has lower variability and is more suitable for proof-of-concept algorithms rather than robust real-world performance. [22]

5 Evaluation Metrics for classification tasks

The performance of biomedical classification tasks is typically evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC AUC. These metrics provide a comprehensive view of the model's ability to classify signals correctly, considering both the proportion of correct classifications and the balance between false positives and false negatives. It is essential to use appropriate evaluation metrics to ensure a fair assessment of the model's performance, particularly when working with imbalanced datasets. To objectively assess the performance of our classification models on biomedical signals such as PCG, we rely on standard evaluation metrics, including accuracy, precision, recall, F1-score, and ROC AUC (Receiver Operating Characteristic - Area Under Curve). These metrics are computed based on the confusion matrix, which includes the following terms:

- True Positives (TP): Correctly predicted positive cases
- True Negatives (TN): Correctly predicted negative cases
- False Positives (FP): Negative cases incorrectly predicted as positive
- False Negatives (FN): Positive cases incorrectly predicted as negative

5.1 The Accuracy

Accuracy measures the overall correctness of the classifier:

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

5.2 The Precision

Precision, or Positive Predictive Value, measures the proportion of predicted positives that are actually positive:

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

5.3 The Recall

Recall, or Sensitivity, measures the proportion of actual positives that are correctly identified:

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

5.4 F1-Score

The F1-score is the harmonic mean of precision and recall, balancing both concerns:

$$\text{The F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.5 The ROC AUC

The ROC AUC (Receiver Operating Characteristic - Area Under the Curve) evaluates the model's ability to distinguish between classes over all threshold settings. It is computed as the area under the ROC curve, which plots the True Positive Rate against the False Positive Rate:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

A higher AUC indicates better separability and robustness of the classifier.

6 Related Work in beat sonds classification

Recent advances in phonocardiogram (PCG) analysis have largely prioritized beat-level classification, utilizing both conventional signal processing techniques and modern deep learning frameworks. Early methodologies often relied on hand-crafted features paired with traditional classifiers, such as:

- Antonio Della Corte et al. (2023) proposed a novel method that transforms one-dimensional phonocardiogram (PCG) signals into two-dimensional color images using Partitioned Iterated Function Systems (PIFS). These images are then used as input to a convolutional neural network (CNN) for classification. Notably, their approach does not require any form of preprocessing or segmentation, simplifying the pipeline while maintaining performance. The model achieved a modified accuracy of 0.85 on the PhysioNet/CinC 2016 dataset [34].

- Schmidt et al. (2018) developed a deep learning model based on Long Short-Term Memory (LSTM) networks for end-to-end classification of PCG recordings. Their method bypasses manual feature engineering by learning directly from spectro-temporal representations of the heart sounds. The LSTM model was validated on public PCG datasets and achieved a precision of 88.5%, confirming its effectiveness for sequential signal analysis [35].

- Fuad Noman et al. (2018) designed an ensemble model that integrates both 1D and 2D CNN architectures. They used raw PCG signals and Mel-Frequency Cepstral Coefficients (MFCCs) as input features to exploit different levels of temporal and spectral information. The hybrid ensemble achieved an accuracy of 89.22% on the PhysioNet/CinC 2016 dataset [36].

- Jonathan Rubin et al. (2017) introduced a method using time-frequency heatmaps generated from PCG recordings, which are then classified by a CNN. The visual representation allowed the CNN to detect relevant spectral patterns. Their system achieved a specificity of accuracy 0.95 and [37].

- Ahmed Patwa et al. (2023) applied wavelet scattering transform to extract robust and translation-invariant features from PCG signals. They compared different deep learning models including 1D CNNs, LSTMs, and convolutional recurrent neural networks (C-RNNs). Among these, precision 96,30

- Tang et al. (2018) adopted a classical machine learning approach using 515 handcrafted features from time, frequency, and entropy domains. These features were fed into a Support Vector Machine (SVM), which reached a balanced accuracy of 88% on the PhysioNet/CinC 2016 dataset [38].

- Potes et al. (2016) presented a hybrid system combining deep CNN-based features with an AdaBoost classifier. This approach enabled the model to learn informative features while benefiting from ensemble classification. The system achieved 86.02% balanced accuracy on the challenge dataset [39].

- Yuhao Zhang et al. (2016) developed a segmental CNN architecture tailored to classify noisy PCG recordings. The method processes individual signal segments to improve robustness and reduce the impact of background noise. Their system achieved 87.5% accuracy on the PhysioNet/CinC 2016 dataset [40].
- Yatao Zhang et al. (2016) extracted 49 handcrafted features including temporal and spectral characteristics such as amplitude and power spectral density (PSD). These features were used with a kernel-based SVM, which achieved a cross-validation accuracy of 87.3% [41].

Table 2.3: Summary of works on PCG data with features, classifiers, and results

Authors	**Features**	**Classifier**	**Results**
Antonio Della Corte et al. (2023)	Sequential, 100 features extracted from PCG using PIFS	CNN	85% modified accuracy
Schmidt et al. (2018)	Sequential (Spectro-temporal), around 20 features learned end-to-end	LSTM	88.5% precision
Fuad Noman et al. (2018)	Sequential (MFCC, raw signal)	CNN (1D, 2D), Hybrid	89.22% accuracy
Jonathan Rubin et al. (2017)	Sequential (time-frequency maps)	CNN	Specificity 0.95
Ahmed Patwa et al. (2023)	Sequential (Wavelet Scattering)	CNN (1D), LSTM, C-RNN	Precision 96.3% (CNN best)
Tang et al. (2018)	Temporal, Frequency, Entropy (515 features)	SVM	88% balanced accuracy
Potes et al. (2016)	Temporal, Frequency	CNN + AdaBoost (Hybrid)	86.02% balanced accuracy
Yuhao Zhang et al. (2016)	Raw signal segments	CNN (Segmental)	87.5% accuracy
Yatao Zhang et al. (2016)	Temporal, Spectral (49 features)	SVM (Kernel)	87.3% cross-validation accuracy

7 Conclusion

This chapter has reviewed the theoretical background and related work in the field of biomedical signal processing, focusing on AI methods, PCG signal analysis, and attention-based models. The research gaps identified, including the lack of interpretability and suboptimal beat selection, have motivated the proposed hybrid approach. This approach integrates attention mechanisms with classical machine learning classifiers, offering a promising solution to the challenges in PCG classification. The insights gained from this chapter inform the system design and experiments presented in Chapter 3.

Chapter 3

Adaptive PCG Analysis Framework: Attention Mechanisms for Variable-Length Signal Classification

1 Introduction

This chapter presents the core of our research work, outlining the methodology developed for the classification of heart sounds. Unlike many studies that focus on beat-level classification, our approach processes the phonocardiogram (PCG) signal at the patient level, considering the entire recording as a whole. The main objective of our system is to automatically distinguish between normal and abnormal heart sounds, providing a decision support tool for clinical use. Our method integrates signal processing techniques and deep learning, aiming to extract and learn relevant discriminative features. Initially, we manually extract 14 temporal and morphological features from the PCG signal, including attributes such as amplitude, duration of heart sound components (S1, S2, systole, diastole), asymmetry, and flatness. In addition to these handcrafted features, our approach leverages attention mechanisms within a deep learning model to automatically learn high-level representations from the signal. These mechanisms allow the model to focus on the most informative parts of the input, enhancing classification performance and interpretability. This dual-feature strategy—combining manually selected features and deep attention-based representations—aims to optimize the overall detection accuracy. Throughout this chapter, we detail each step of the proposed pipeline, from preprocessing and feature extraction to model training and evaluation

2 Data Preparation and Evaluation Protocol

This section provides a detailed description of the dataset, including its structure, annotation format, and the cleaning procedures applied to ensure data quality. Additionally, we define the test protocol used to evaluate model performance.

2.1 Dataset description

The dataset is composed of six distinct training subsets, labeled from Training-a to Training-f. Each subset contains a unique distribution of normal and abnormal heart sound recordings. In total, the dataset comprises 3,240 recordings The recordings encompass a wide range of subjects, including both healthy individuals and patients with heart conditions, as well as both pediatric and adult populations. As a result, each subset reflects its own specific recording environment and clinical context.we summarise in table 3.1 .

Table 3.1: Distribution of normal and abnormal recordings in the PhysioNet/CinC 2016 training dataset

Data set	**Normal**	**Abnormal**	**Total**
Training-a	117	292	409
Training-b	386	104	490
Training-c	7	24	31
Training-d	27	28	55
Training-e	1958	183	2141
Training-f	80	34	114
Total	**2575**	**665**	**3240**

The dataset consists of six distinct directories, each containing phonocardiogram (PCG) recordings collected under specific environmental conditions. The number of recordings per folder varies, reflecting differences in data distribution across the acquisition scenarios. To illustrate this distribution, Figure 2 presents a histogram visualizing the sample counts per folder, highlighting potential imbalances in the dataset.

2.2 The PCG annotation files

The dataset provides annotation files in MAT format for each subset (A-F), containing references to corresponding PCG recordings in WAV format. Following the PhysioNet2016 standard, annotations are implemented at the patient level,

meaning each label applies to the complete recording rather than individual heartbeats, where the binary coding scheme uses -1 to denote "Normal" and 1 for "Abnormal" . Beyond this patient-level classification, the annotation files also encode detailed temporal sequences of cardiac events, with each event precisely timestamped in milliseconds.As shown in Figure 3.1 , this temporal data follows a two-column structure where the first column specifies the event's time point and the second column identifies the corresponding cardiac phase or sound: S1 (first heart sound) marking closure of the atrioventricular valves (mitral/tricuspid) and initiating systole (ventricular contraction), followed by S2 (second heart sound) produced by semilunar valve (aortic/pulmonary) closure and transitioning into diastole (ventricular relaxation). This dual annotation approach supports both holistic pathological assessment and fine-grained analysis of cardiac cycle dynamics.

1	1	'diastole'
2	679	'S1'
3	959	'systole'
4	1479	'S2'
5	1679	'diastole'
6	2719	'S1'
7	2999	'systole'
8	3479	'S2'
9	3679	'diastole'
10	4719	'S1'
11	4959	'systole'
12	5479	'S2'
13	5679	'diastole'

Figure 3.1: Sample of a0001

2.3 Dataset cleaning process

While examining these files, we identified entries that were either empty or incomplete corresponding label ('N').. To maintain the integrity of the dataset and avoid potential issues during analysis, we systematically removed these empty entries. The cleaned dataset now contains only valid, fully annotated records, as summarized in Table 2. This step ensures that subsequent analyses and machine learning tasks are based on accurate and reliable data, this is example in figure 3.2 .

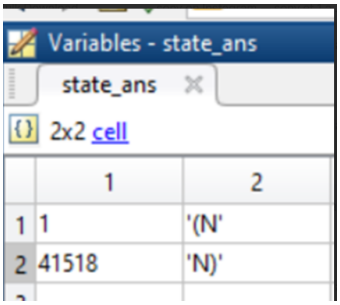


Figure 3.2: Labeling empty of sample a0006

After cleaning the data, we reviewed all annotation files to identify and remove any entries that were empty or incomplete. This process ensured that the dataset only contained valid and reliable records, The table below explains 3.2 . As a result, the final dataset is free from missing or erroneous annotations.

Table 3.2: Distribution of normal and abnormal recordings in the modified PhysioNet/CinC 2016 training dataset

Data set	**Normal**	**Abnormal**	**Total**
Training-a	101	291	392
Training-b	355	13	368
Training-c	3	24	27
Training-d	25	27	52
Training-e	1921	5	1926
Training-f	77	31	108
Total	**2482**	**391**	**2873**

The stacked bar chart clearly illustrates the spread of normal and abnormal medical cases across six patient subsets, highlighting a significant imbalance, where abnormal cases vastly exceed normal ones,we summarize in figure 3.3 .

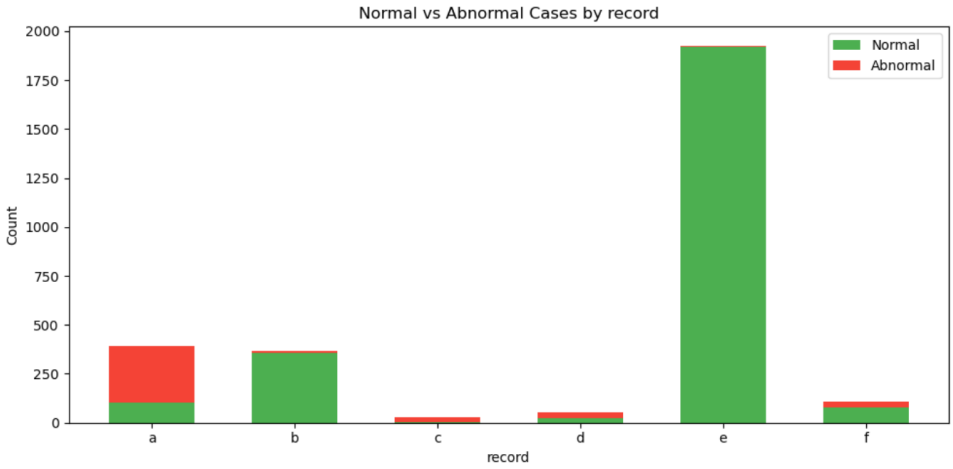


Figure 3.3: Comparison of Normal and Abnormal Cases in Six Patient Groups

2.4 The used test protocol

To rigorously evaluate model performance and ensure reliable generalization, the dataset is systematically partitioned into two distinct subsets: a training set (70%) and a test set (30%). The training set serves as the foundation for model development, enabling the algorithm to learn underlying patterns by optimizing its parameters . During this phase, techniques like cross-validation may further divide the training data into smaller folds to fine-tune hyperparameters and mitigate overfitting without touching the test data. In contrast, the test set (30%). is strictly isolated during the entire training process, acting as an unbiased benchmark to simulate real-world performance. It is used exactly once—after model finalization—to provide an objective assessment of generalization capability. This separation prevents data leakage and ensures that performance metrics.reflect the model's ability to handle unseen data. The 70/30 split balances computational efficiency (adequate training samples) with statistical reliability (sufficient test samples), though ratios may be adjusted for imbalanced datasets or time-series scenarios where temporal ordering matters.As illustrated in Figure 3.4 .

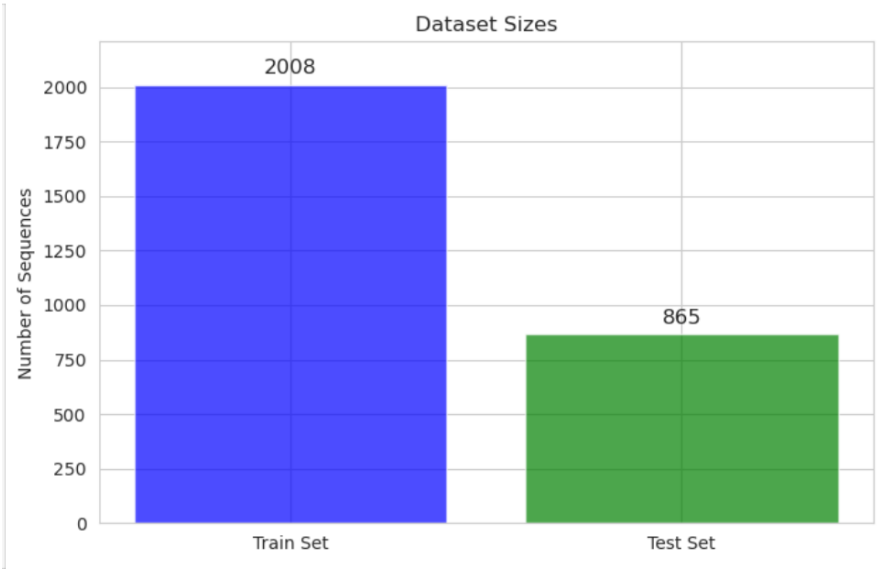


Figure 3.4: dataset sizes

This bar chart displays in table 3.3 .the distribution of normal and abnormal labels across six datasets (a to f) for both training and testing sets. The y-axis represents the count of samples, while the x-axis lists the datasets. Each bar is divided into four color-coded segments: dark green for normal samples in the training set, red for abnormal samples in the training set, light green for normal samples in the test set, and pink for abnormal samples in the test set.

Table 3.3: Number of normal and abnormal samples per directory in the train and test sets

Directory	**Train Normal**	**Train Anormal**	**Test Normal**	**Test Abnormal**
a	90	184	26	92
b	205	52	90	21
c	5	13	2	7
d	18	18	8	8
e	1242	106	538	40
f	51	24	26	7

From the chart, it is clear that dataset ”e” contains the largest number of samples, with a strong predominance of normal labels in both training and test sets. In contrast, datasets ”c” and ”d” have the fewest samples. The proportion of abnormal samples varies between datasets, especially in the training set. This visualization highlights class imbalance and differences in label distribution between datasets and splits ,we summarize in figure 3.5 .

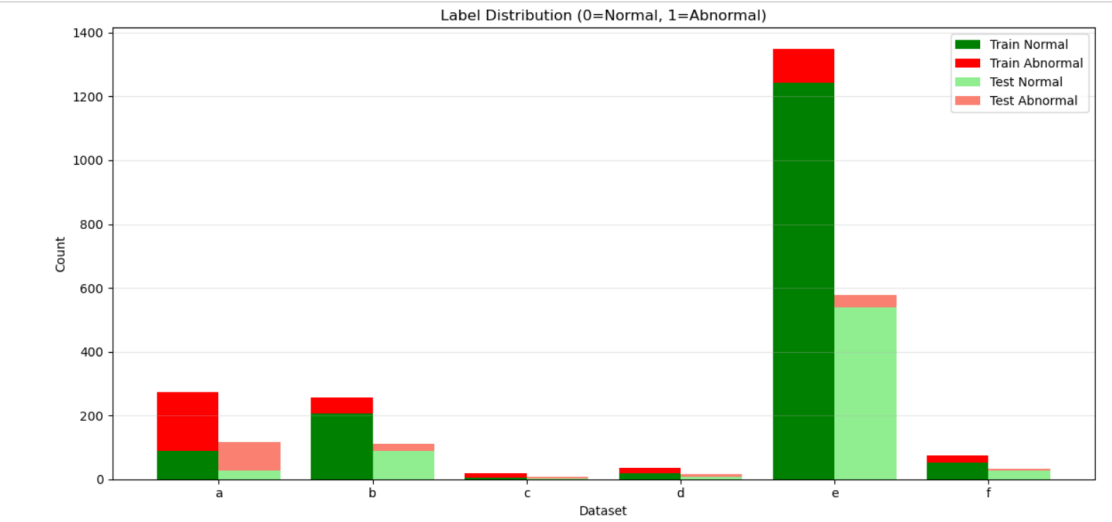


Figure 3.5: Comparison of Normal and Abnormal Cases in Six Patient Groups whith train and test

3 PCG signal segmentation

The segmentation of phonocardiogram (PCG) signals involves isolating key components such as the first (S1) and second (S2) heart sounds, systole, and diastole intervals. This process enhances feature extraction for classification by distinguishing pathological murmurs from normal heartbeats. Advanced techniques like wavelet transforms, envelope detection, and machine learning are used to improve segmentation accuracy, particularly in noisy recordings. Effective segmentation is critical for reliable heart sound analysis and automated diagnosis

3.1 PCG signal description

The process begins with the acquisition of raw PCG signals, which are audio recordings capturing the acoustic activity of the heart. These signals serve as the foundational input for the entire system. for each PCG recording. In this context, we opted to annotate all segments based on the annotation file for each patient's PCG recording. As exemplified in Figure 3.6 .

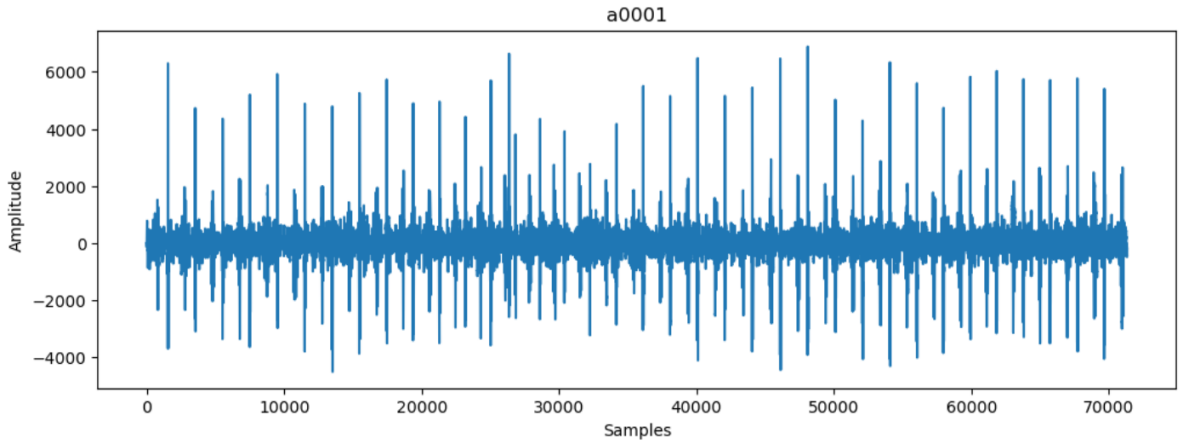


Figure 3.6: Signal length of sample a0001

3.2 PCG signal segmentation description

In this stage, the continuous PCG signal is segmented into individual heartbeats. This segmentation is crucial for isolating discrete cardiac cycles, allowing for more precise analysis of heart sounds and potential anomalies within each beat(S1,S2,Systole,Diastole) after the results of the segmentation,As exemplified in Figure 3.7 .

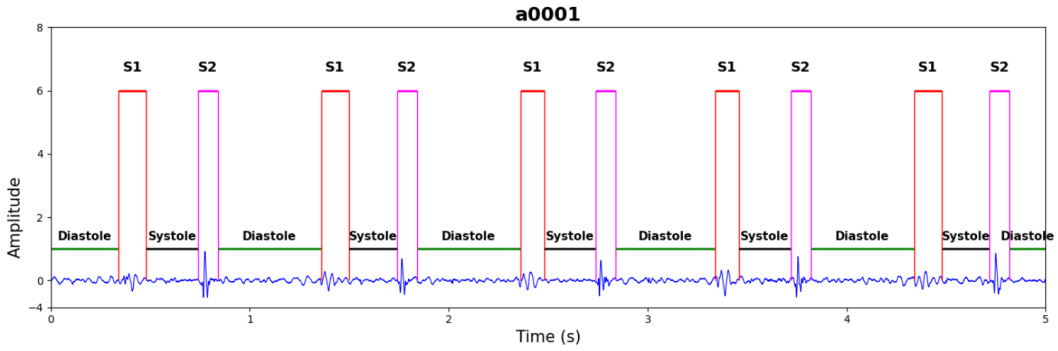


Figure 3.7: Segmentation signal of sample a0001

3.3 Cinc 2016 dataset statistics

The table 3.4 , presents basic statistics on the duration of an event or process, measured in seconds. It includes three key values:

Maximum duration: The longest recorded duration is 145.00 seconds.

Minimum duration: The shortest recorded duration is 5.31 seconds.

Average duration: The mean duration across all observations is 22.53 seconds.

These statistics provide an overview of the range and central tendency of the durations measured.

Table 3.4: Summary of PCG recording durations in the training dataset.

Statistic	**Duration (seconds)**
Total patients	2873
Maximum duration	145.00
Minimum duration	5.31
Average duration	22.53

The table 3.5 , summarizes key statistics related to the number of beats. It includes three main values:

Mean: The average number of beats is 28.9.

Maximum: The highest recorded number of beats is 173.0.

Minimum: The lowest recorded number of beats is 3.0.

These statistics provide an overview of the distribution and variability in the beats data.

Table 3.5: Summary statistics of the beats per patient.

Statistic	**Value**
Total patients	2873
Mean beats per patient	28.9
Maximum beats	173
Minimum beats	3

4 Cinc 2016 dataset challenges

The PhysioNet/CinC 2016 dataset presents several challenges for researchers . It contains a large variety of heart sound recordings with varying quality, noise levels, and patient conditions, making accurate analysis difficult. Additionally, the dataset exhibits significant class imbalance, with many more normal cases than abnormal ones in certain records.

4.1 The unbalanced dataset challenge

One of the main challenges in our classification task is the imbalanced distribution between the two classes: normal and abnormal heart sounds. In the

PhysioNet/CinC 2016 dataset, a larger proportion of the recordings are labeled as normal, while abnormal cases represent a minority. This imbalance can lead to biased models that favor the majority class during training. in the Figure 3.8 for a comparison of the results.

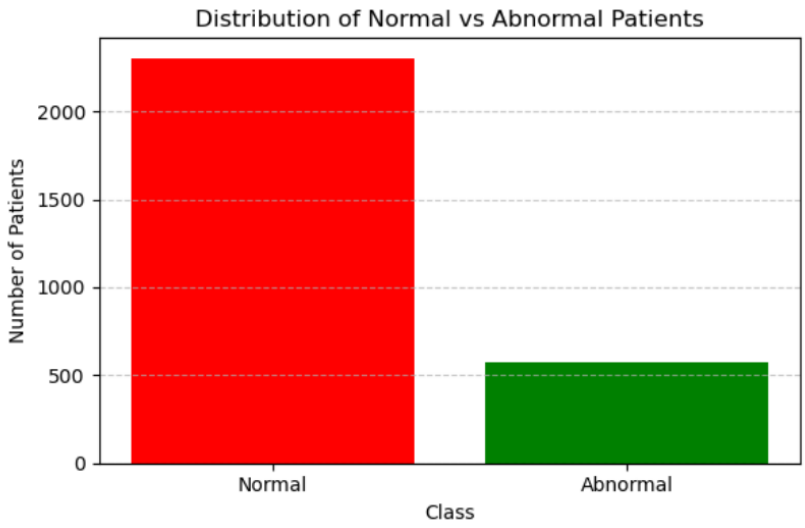


Figure 3.8: Comparison of Normal vs Abnormal Patients

4.2 Environment dataset challenge

One of the key challenges encountered in working with the PhysioNet/CinC Challenge 2016 dataset is the inconsistency in recording environments across patients. Since the dataset was collected independently by different research teams in various hospitals and clinical centers across multiple countries and continents, each patient's heart sound recordings may vary due to several environmental factors.

4.3 Variability of Sequence Lengths

A significant challenge in processing PCG signals from the PhysioNet/CinC Challenge 2016 dataset lies in the inconsistent length of heart sound recordings across different patients. Each patient's signal may differ in duration due to factors such as the length of the examination, the number of recorded heartbeats, and interruptions or pauses during the acquisition process. As shown in Figure 3.9 .

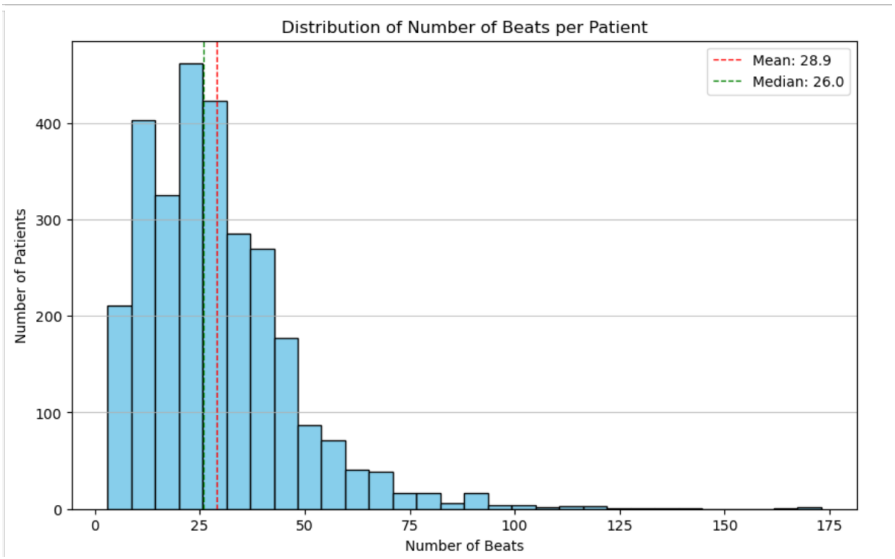


Figure 3.9: Signal length per patient

5 Features Extraction Strategy for PCG Signals

Effective PCG analysis requires a robust feature extraction strategy combining time-domain, frequency-domain, and time-frequency approaches. Time-domain features capture heart sound durations and intervals, while spectral analysis reveals frequency characteristics. Wavelet transforms provide localized time-frequency information, and statistical measures quantify signal properties. Feature selection techniques like PCA or mutual information optimize the extracted features, ensuring efficient and accurate classification of normal and abnormal heart sounds. This multi-domain approach enhances diagnostic reliability in automated cardiac assessment systems

5.1 Temporal Features

Effective classification of phonocardiogram (PCG) signals relies on the extraction of discriminative features that capture both temporal and spectral characteristics. Fourteen key features we extracted, including time-domain parameters such as heart sound durations and intervals, frequency-domain measures like spectral entropy and power ratios, and time-frequency representations through wavelet transforms.

Mean S1	Mean S2	Mean Systole	Mean Diastole	Dur. S1	Dur. S2	Dur. Systole	Dur. Diastole	Peak S1	Peak S2	Skew. S1	Skew. S2	Kurt. S1	Kurt. S2

5.2 Signals Length Normalization

In order to train deep learning models effectively, it is essential that all input sequences have the same length. However, in real-world scenarios, phonocardiogram (PCG) signals collected from different patients naturally vary in duration due to differences in recording conditions, heart rates, and signal quality.

To address this challenge, we implemented a normalization strategys all PCG signals have a uniform length.

5.2.1 Normalization with zero-padding

To ensure that all patient PCG recordings have a consistent length for subsequent analysis and machine learning processing, we applied zero-padding to the signals. This procedure involves extending each PCG signal by appending zeros at the end until it reaches a predefined, uniform length. By doing so, we eliminate discrepancies caused by varying recording durations across patients. This uniformity is essential for batch processing and for feeding the data into models that require fixed-size input vectors. Zero-padding preserves the original signal content while standardizing the input dimensions, thus facilitating fair comparison and efficient computation during the feature extraction and classification stages.As presented in Figure 3.10 .

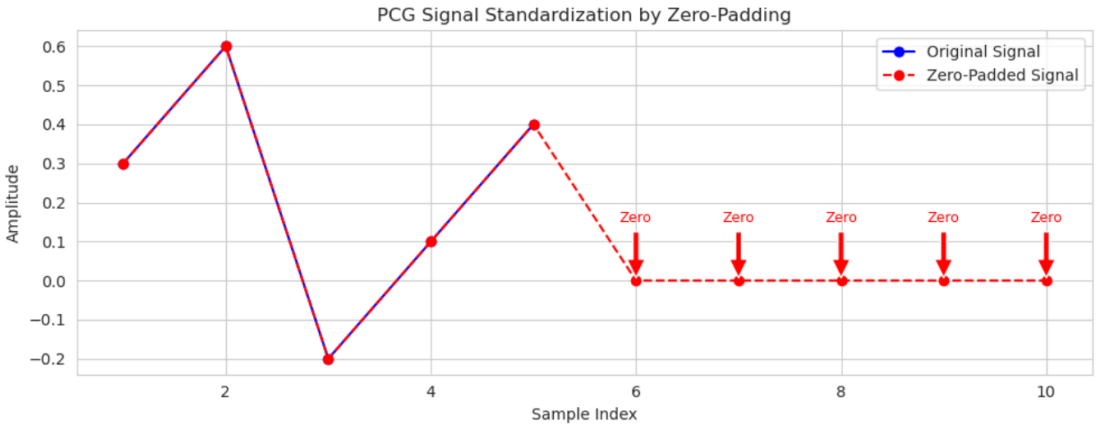


Figure 3.10: Enter Caption

5.2.2 Normalization with duplication of the last beat

To ensure that all PCG segments have the same length, we implemented a strategy based on duplicating the last cardiac cycle. When a segment is shorter than the max length, we repeatedly append copies of its final beat until the segment reaches the required size. This method is particularly advantageous when dealing with datasets containing recordings of varying durations, as it guarantees that all segments are standardized without losing important signal information. As presented in Figure 3.11 .

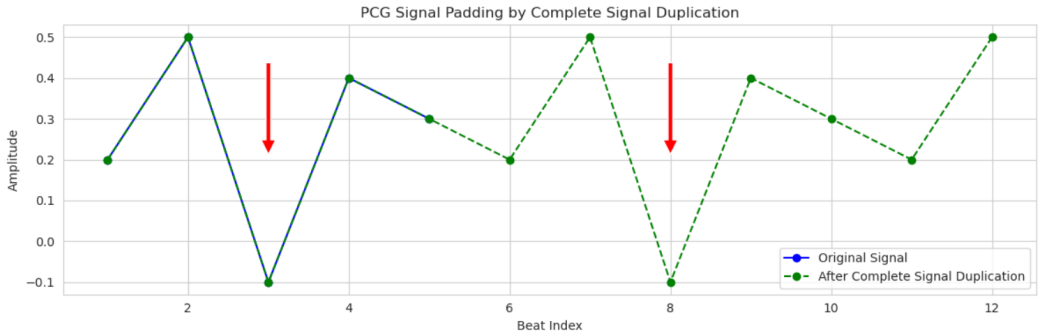


Figure 3.11: Signal Padding by Last Beat Duplication with Last Beat

5.2.3 Normalization with the duplication of whole Signal

In order to standardize the length of phonocardiogram (PCG) recordings across different patients, we employed a method consisting of duplicating the entire signal sequentially until reaching max length. This approach involves concatenating multiple copies of the original PCG signal end-to-end, effectively extending its duration without introducing artificial silence or distortion. By replicating the complete signal, we preserve the intrinsic temporal and morphological characteristics of the heart sounds, ensuring that the extended signal remains physiologically consistent. As presented in Figure 3.12 .

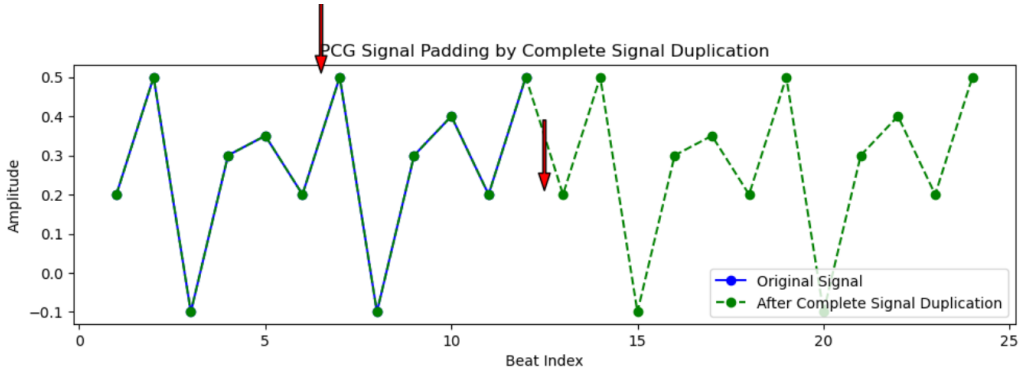


Figure 3.12: Complete Signal Duplication

5.2.4 Normalization with fixed threshold

This diagram outlines the approach used to standardize the length of PCG (phonocardiogram) signals based on a predefined threshold:

If the signal is longer than the threshold, the signal is truncated to match the threshold length. This ensures that only the portion of the signal within the specified duration is retained for further analysis.

If the signal is shorter than the threshold, the method involves repeatedly duplicating the last cardiac cycle (beat) until the signal reaches the desired length. This preserves the physiological characteristics of the original signal while achieving uniformity across all samples.

If the signal length is exactly equal to the threshold, the signal is left unchanged and used as is.

This strategy guarantees that all PCG signals processed in subsequent steps have a consistent length. we summarise this in figure 3.13.

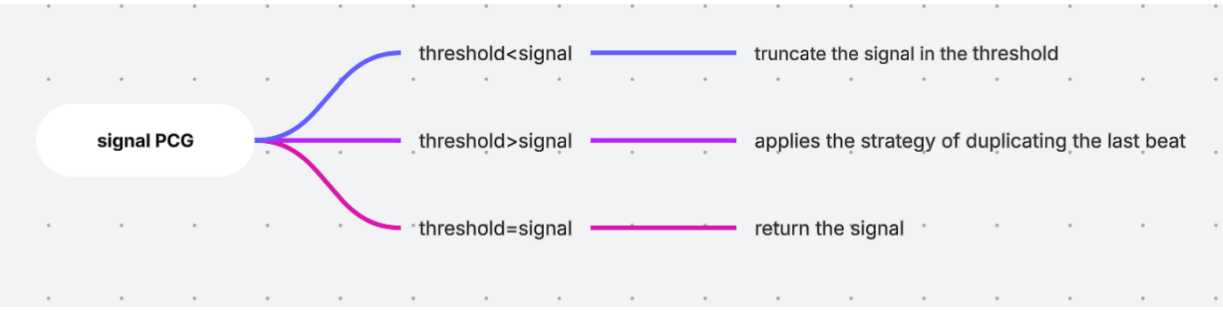


Figure 3.13: Threshold strategy

5.2.5 Normalization with Attention-based dynamic strategy

Our main contribution in this work is the development of an adaptive fixed-length strategy based on an LSTM with an attention mechanism to robustly address the variability in the number of cardiac cycles across PCG recordings. In this approach, each PCG signal is initially segmented into its constituent beats, and a set of time- domain and statistical features is extracted from each beat. Rather than relying on zero- padding or arbitrary beat duplication, the model is designed to select the K most informative beats within each recording. The attention mechanism integrated into the LSTM architecture enables the model to automatically identify and assign higher weights to beats that are most relevant for the classification task, thereby emphasizing those likely to contain diagnostically significant information. During both training and inference, only the K beats with the highest attention scores are retained for each recording, ensuring a consistent and fixed-length input for the neural network regardless of the original signal length. If a recording contains fewer than K beats, minimal padding is applied to fill the remaining positions, but the essential information always originates from actual cardiac cycles, as illustrated in Figure 3.14 .



Figure 3.14: Attention-Based Beat Selection for Variable-Length PCG Signals strategy

6 Architecture of the Attention-Based Beat Selection for Variable-Length PCG Signals classification

The architecture relies on a BiLSTM-based encoder coupled with an attention mechanism to effectively classify phonocardiogram (PCG) signals. The BiLSTM component is designed to extract long-term temporal features across heartbeats, while the attention module selectively emphasizes time regions that carry

the most clinical significance. To address the inherent data imbalance in heart sound datasets.we plan this in the figure 3.15 .

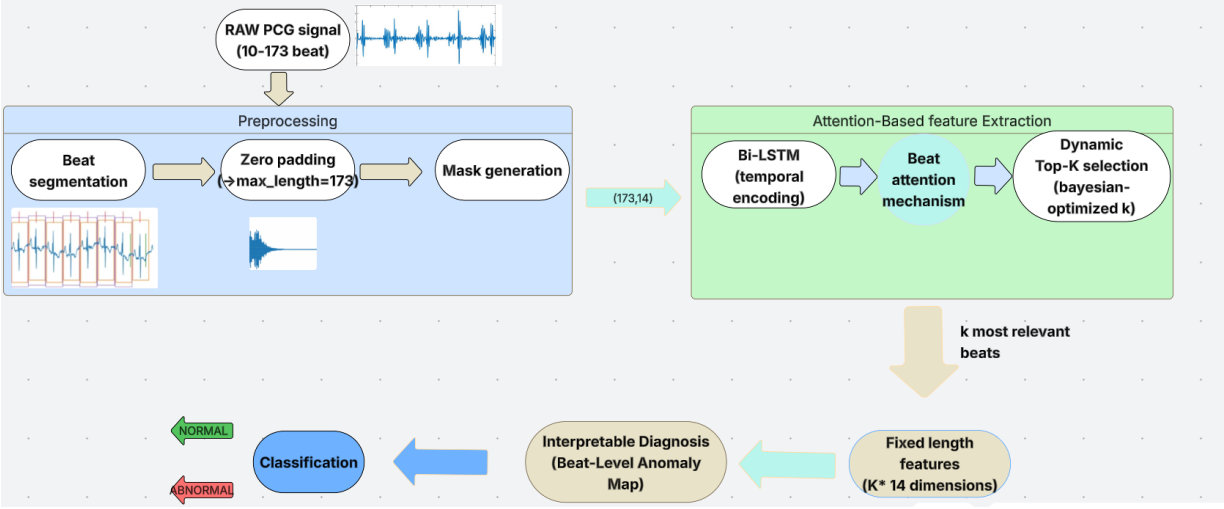


Figure 3.15: Architecture of Top Features

6.1 BiLSTM encoder architecture

The figure 3.16 ,above illustrates the architecture of a deep learning model designed for sequential data analysis. The model takes two inputs: a multivariate time series and an auxiliary sequence. These inputs are processed and combined using a Lambda layer, followed by a bidirectional LSTM layer that captures temporal dependencies in both forward and backward directions. An attention mechanism (BeatAttention) is then applied to focus on the most relevant time steps. Finally, a dense layer produces the model's output, enabling effective sequence classification or regression. This architecture leverages both recurrent and attention-based components to enhance performance on complex sequence tasks.

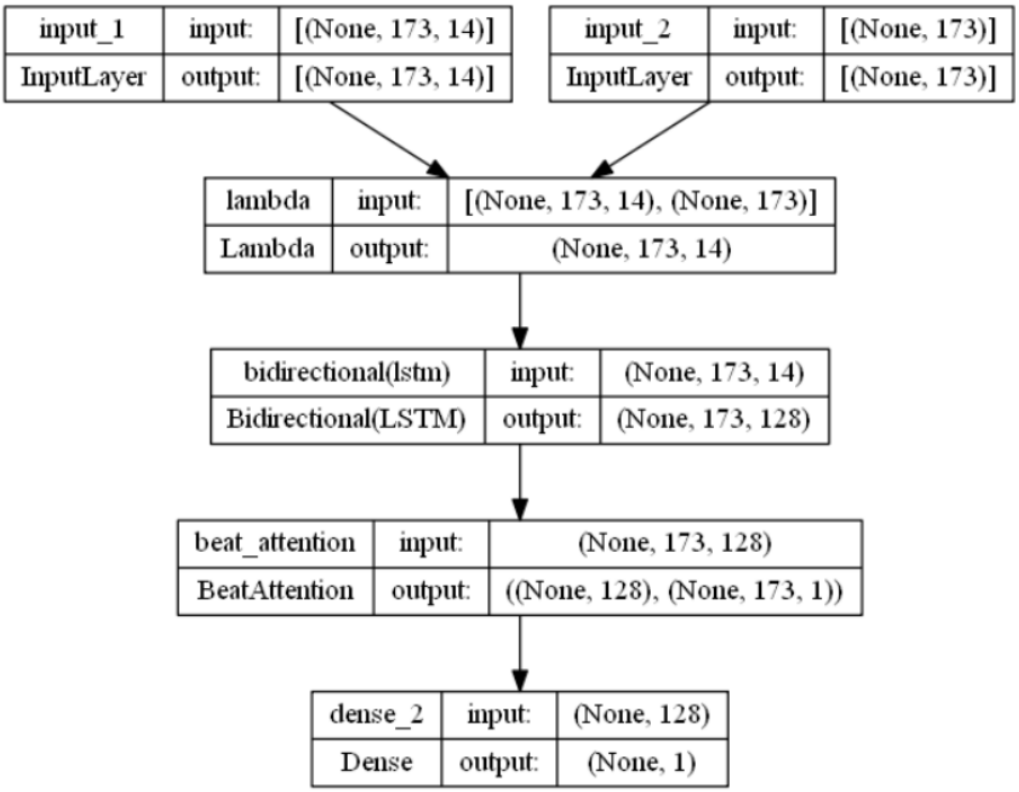


Figure 3.16: Architecture of LSTM

The proposed architecture is designed to classify PCG signals based on beat-level features using a deep learning model enhanced with an attention mechanism. The model begins with two input layers: the main input layer receives a sequence of beats with each beat represented by 14 features, while the second input is a boolean mask indicating valid time steps in each sequence, allowing the model to ignore padded values. To ensure temporal dependencies are learned effectively, the core of the model features a Bidirectional LSTM layer with 64 units in each direction (forward and backward), outputting a hidden representation of shape (batch size, max beats, 128). A dropout rate of 0.3 is applied for regularization.

Following the LSTM layer, a custom attention mechanism called BeatAttention is employed. This layer computes attention scores for each time step using two dense sub layers: the first (W) with 32 hidden units applies a non-linear transformation, and the second (V) is a single-unit dense layer (without bias) that produces scalar attention scores for each beat. These scores are then masked to ignore padded values by adding a large negative number to the padded positions. A softmax function is applied to normalize the scores across time steps, generating

attention weights. These weights are then used to compute a context vector as a weighted sum of the LSTM outputs, effectively summarizing the most informative parts of the sequence into a single vector of shape (batch size, 128).

Finally, a Dense output layer with a sigmoid activation function is applied to this context vector to produce a binary classification output, indicating whether the PCG signal corresponds to a normal or abnormal heartbeat pattern. The model is compiled using the Adam optimizer and binary cross-entropy loss, and it is trained for 200 epochs with a batch size of 32. During training, both the input data and the corresponding masks are provided, ensuring that the model correctly processes sequences of varying lengths. Additionally, a secondary model is used to extract and visualize the attention weights, allowing interpretation of which beats contributed most to the final decision.

we summarize that on the table 3.6 .

Table 3.6: Overview of the heartbeat sequence classification model architecture.

Step	**Layer / Operation**	**Output Shape**	**Description**
Input	Input Layer (max_beats, 14)	(batch_size, max_beats, 14)	Sequence of heartbeats, each represented by 14 extracted features.
Mask	Boolean Mask Input	(batch_size, max_beats)	Indicates valid time steps in the sequence for variable-length input handling.
Bidirectional LSTM	Bidirectional(LSTM(64, return_sequences=True, dropout=0.3))	(batch_size, max_beats, 128)	Encodes temporal context by processing the sequence in both directions.
Attention	BeatAttention(32)	(batch_size, 128)	Computes a weighted context vector from the sequence using attention scores.
Final Output	Dense(1, activation='sigmoid')	(batch_size, 1)	Outputs the probability of the heartbeat sequence being abnormal.

6.2 The Hyperparametres of the LSTM Model proposed

The hyperparameters, set before training, determine the behaviour and performance of the model. They include the optimizer (Adam), the loss function (Focal Loss), the batch size, the number of epochs, as well as techniques such as early stopping and SMOTE oversampling to balance the classes. Their adjustment is crucial for optimising learning, avoiding overlearning and improving generalisation. Several metrics (accuracy, precision, recall, F1-score, ROC AUC) are used to evaluate the model, with threshold optimisation to maximise the F1-score. we summarise in table 3.7 .

Table 3.7: Training Hyperparameters for the Proposed Model

Component	**Description**
Optimizer	Adam
Loss Function	Focal Loss ($\gamma = 2$, $\alpha = 0.75$)
Batch Size	32
Epochs	200
Early Stopping	Patience = 15, Restore best weights = True
Oversampling Technique	SMOTE (Synthetic Minority Over-sampling Technique)
Input Reshaping	Flatten for SMOTE, then reshape to `(N, max_beats, features)`
Class Weighting	Computed using `compute_class_weight(balanced)`
Evaluation Metrics	Accuracy, Precision, Recall, F1 Score, ROC AUC, Balanced Accuracy
Threshold Optimization	Search over [0.1, 0.9] step = 0.05 (based on F1-score)
Confusion Matrix	Raw and normalized values visualized with heatmaps

6.3 Training results

The model was trained over 200 epochs using the training dataset, with the goal of minimizing the loss function and improving accuracy. During training, the model's performance on both training and validation sets was monitored to detect overfitting. Early stopping or regularization techniques were applied as needed to enhance generalization. The training curves of accuracy and loss provide insight into the model's learning behavior and convergence.

6.3.1 Model Accuracy and Loss

These figures 3.17 and 3.18 ,illustrate the model's training and validation performance over 200 epochs, highlighting signs of overfitting. The figure 3.17 ,shows

that training accuracy steadily increases, surpassing 0.95, while validation accuracy remains relatively stable around 0.84, indicating the model fits the training data well but does not generalize better to unseen data.the figure 3.18, complements this by showing the training loss consistently decreasing to near 0.1, whereas validation loss initially decreases slightly but then rises and fluctuates, further confirming overfitting as the model captures noise or specific patterns in the training set that do not apply to new data. Together, these accuracy and loss curves reveal that while the model improves on training data, its validation performance plateaus or worsens, a classic symptom of overfitting

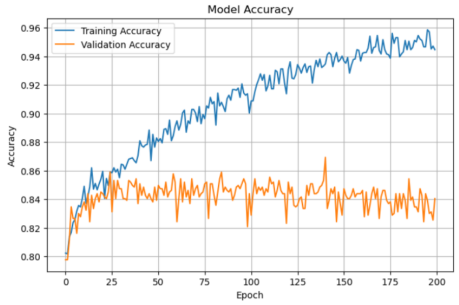


Figure 3.17: Training accuracy curve



Figure 3.18: Training loss curve

6.3.2 Analysis of Model Attention Patterns on Training and Test Data

The attention weight distributions and mean attention by position plots for both the training and test sets provide valuable insights into the model's focus when processing input sequences. in figure3.19 and figure 3.20 .

The attention weight analysis reveals that in both training and test sets, the model exhibits a highly selective attention pattern, with most weights clustered near zero, indicating sparsity and focus on a few critical time steps. The mean attention by position plots show that the model consistently allocates the highest attention to the earliest time steps of the sequence, with a sharp decline thereafter. This consistent pattern across training and test data suggests that the model robustly identifies and prioritizes the most informative initial parts of the input signal, demonstrating effective generalization rather than overfitting. Overall, these visualizations confirm the model's ability to attend selectively to key input regions in both seen and unseen data.

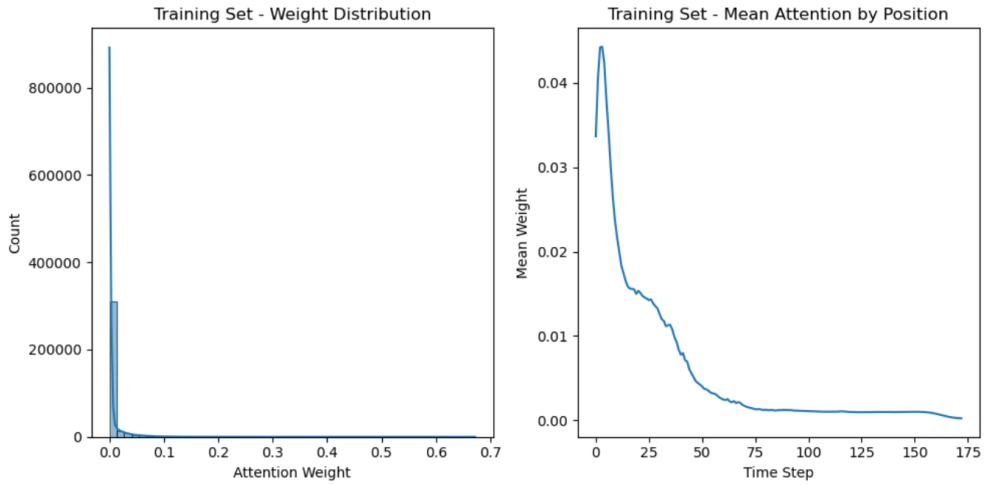


Figure 3.19: Attention Weight Distribution and Positional Focus in the Training Set

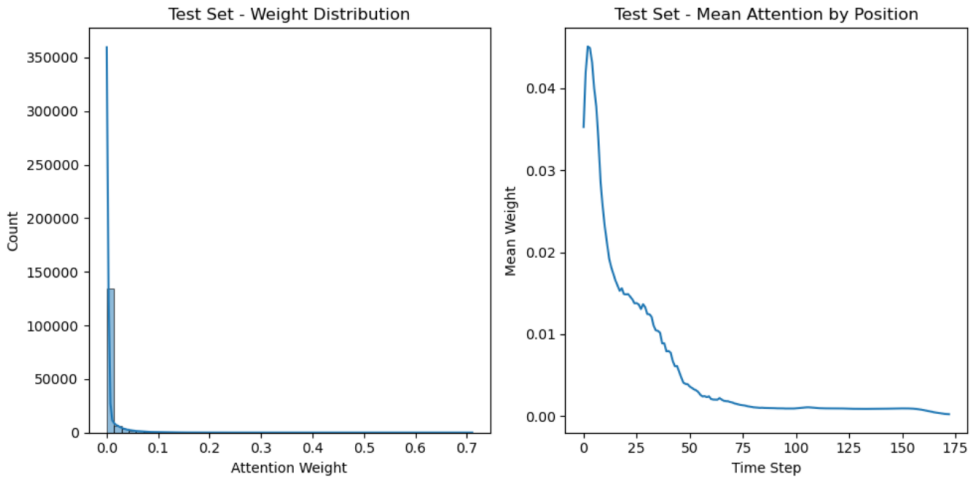


Figure 3.20: Attention Weight Distribution and Positional Focus in the Test Set

6.4 Visualization of the attention mechanism with key regions in the PCG signals

Visualizing a PCG signal along with an attention map allows us to intuitively identify the time segments that the AI model considers most important for distinguishing a normal heart sound from a pathological one, see figure 20. On the horizontal axis, each time window corresponds to a successive heartbeat, while the

vertical axis indicates the intensity of the acoustic signal. Attention weights, represented by colored areas or numerical values superimposed on the signal, reflect the relative importance assigned by the model to each segment: high weights highlight critical regions of the signal, often associated with anomalies such as murmurs or abnormal noises, while low weights indicate less relevant or silent segments.As illustrated in Figure 3.21 .

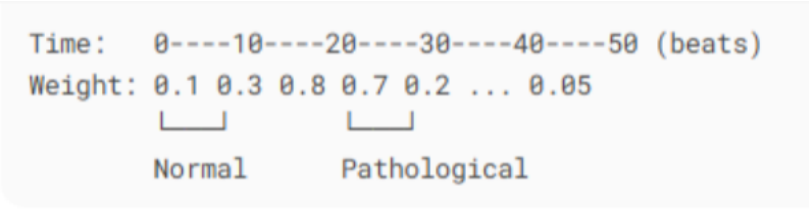


Figure 3.21: Example PCG signal with attention heatmap

The figure 3.21 , demonstrates how the attention mechanism identifies important parts of a PCG signal: In the top plot, multiple PCG signal waveforms are shown, illustrating the variation in amplitude over time steps. The bottom plot presents the attention weights assigned by the model to each time step. Most time steps receive relatively low attention weights (orange bars), but three specific time steps stand out with the highest values (red bars). These top three time steps, highlighted in red, are where the model focuses most when analyzing the signal. This suggests that the model considers these segments of the PCG signal to be the most informative or relevant for its prediction. Ideally, these highly attended time steps should correspond to clinically meaningful features in the heart sound, such as murmurs or abnormal beats.

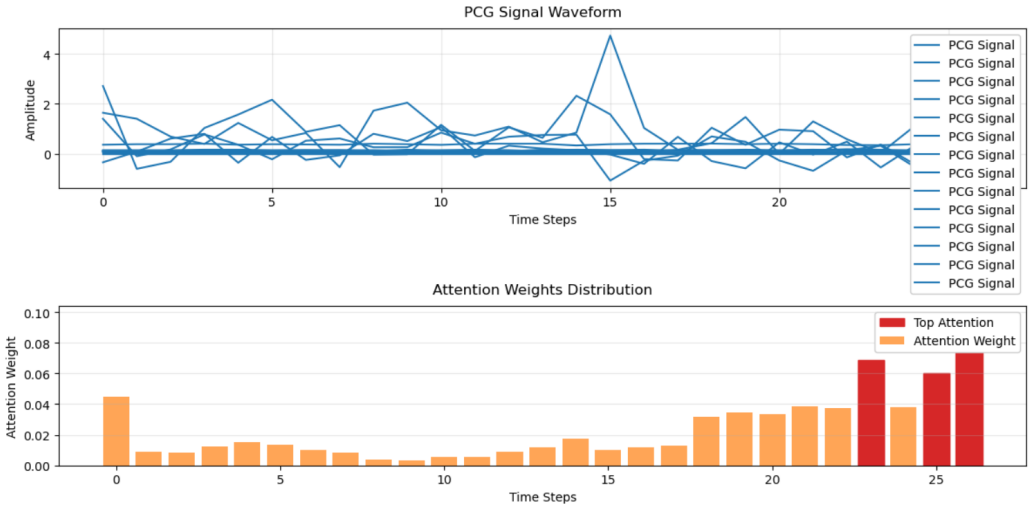


Figure 3.22: Visualization of PCG Signal and Model Attention Weights

6.5 Adaptative feature extraction process

The heart sound classification process begins with a data preparation phase, which in- cludes the input PCG signal sequences for training and testing, and attention weights. Next, an initial feature extraction step is carried out using a dedicated module that selects the most relevant beats (top K) and applies normalization using a standard scaler. Bayesian optimization is then performed to identify the optimal model hyperparam- eters, specifically the number of beats K and the maximum depth of the Random Forest algorithm. Once the best parameters are identified, final feature extraction is carried out using the optimal K, and the features are rescaled. The final classification model is trained using Random Forest with the optimized parameters. in Figure 23 ,displays a horizontal bar chart of the top 20 most important features identified by the model, ranked by their contribution to predictive performance. The most significant features predominantly belong to the "F5" frequency band, with "Beat 9 F5" ranked highest, followed by "Beat 10 F5" and "Beat 8 F5." This pattern indicates that both the specific beat and frequency band are crucial in the model's decision-making process. The prominence of the F5 band suggests it is particularly discriminative for the task, while the variation in beat indices highlights the temporal or sequential information captured by the model. The gradual decline in importance scores among these features reflects varying degrees of influence, providing valuable insights for feature selection, model interpretation, and further domain-specific research into the physiological or signal processing significance of these beat-frequency combinations

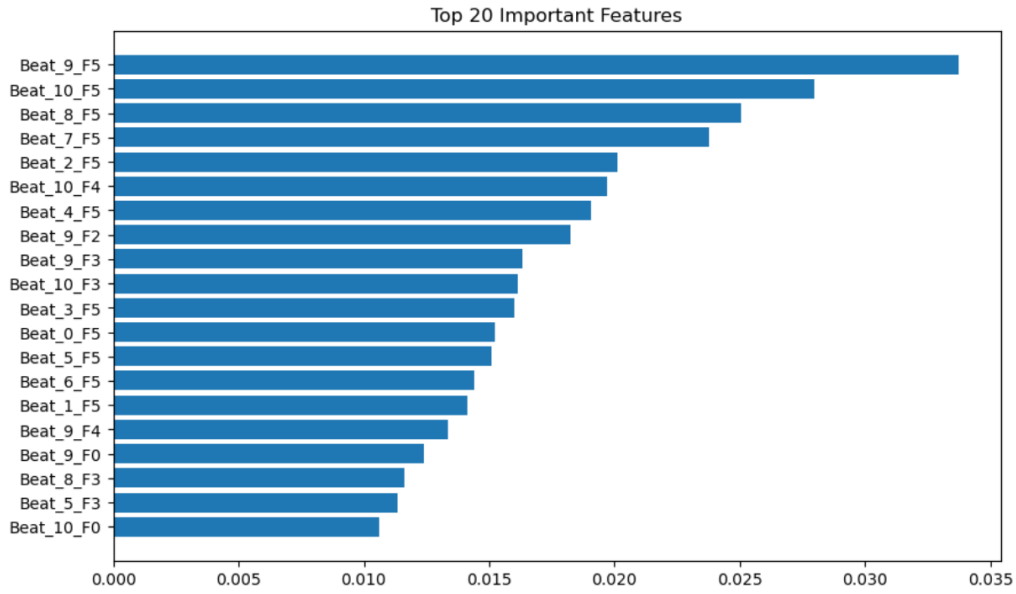


Figure 3.23: Affecting Weights of the beats

7 Classification Engine for Beat-Level Cardiac Anomaly Detection

The selection of optimal hyperparameters was informed by computational limitations and the clinical imperative for interpretability. For each classifier (Random Forest, Gradient Boosting, SVM, and k-NN), parameter choices were prioritized to address class imbalance via intrinsic methods and to minimize overfitting to the predominant normal beats.

7.1 Random Forest:

We used the `RandomForestClassifier` implementation with `n_estimators=100`, meaning the forest is composed of 100 trees. A larger number of trees can improve model performance but also increases computation time. To ensure reproducibility, we set `random_state=42`.

7.2 K-Nearest Neighbors (K-NN):

We employed the `KNeighborsClassifier` with `n_neighbors=5`, which means classification is based on the 5 nearest neighbors for majority voting. A larger

number of neighbors makes the model more robust but less sensitive to local variations.

7.3 Support Vector Machine (SVM):

We used the `SVC` implementation with an RBF kernel (`kernel='rbf'`), a regularization parameter `C=1.0`, and an RBF kernel coefficient `gamma='scale'`. To ensure reproducibility, `random_state=42` was fixed.

7.4 Decision Tree:

We used `DecisionTreeClassifier` with `random_state=42` to control reproducibility.

7.5 Bagging:

We applied `BaggingClassifier` with a base estimator `DecisionTreeClassifier` and `n_estimators=50`.

7.6 Boosting:

We used both `AdaBoostClassifier` and `GradientBoostingClassifier` with 100 trees each to improve performance through sequential aggregation.

8 Results and Discussions

The dataset used in this study is unbalanced, which posed certain challenges during analysis. We applied 14 temporal features to each strategy to comprehensively capture the relevant patterns. To evaluate the performance, we employed average-based methods across these features. The results indicate that despite the imbalance, the selected temporal features effectively contributed to distinguishing between classes in most strategies. These findings highlight the importance of feature selection and averaging techniques in handling unbalanced datasets. Further analysis and comparison with existing methods will be discussed to contextualize these results within the broader literature.

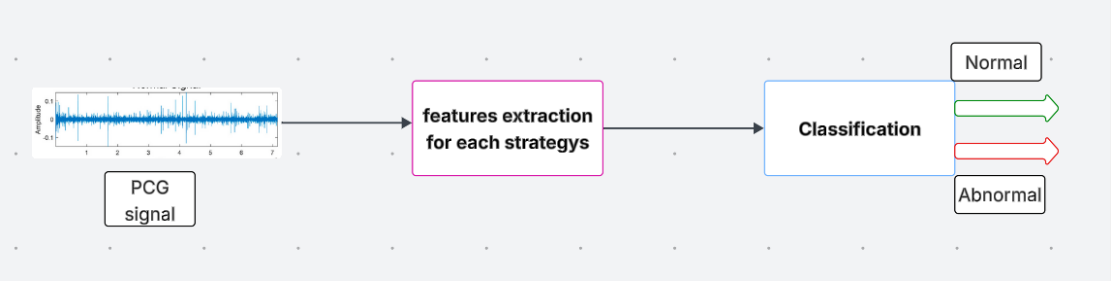


Figure 3.24: Classification for each strategys

This table (table3.9) presents the distribution of normal and abnormal samples across the training and test datasets. The training set contains 1,611 normal and 397 abnormal instances, while the test set includes 690 normal and 175 abnormal instances. The data shows a similar class imbalance in both splits, with normal cases significantly outnumbering abnormal ones

Table 3.8: Train and test statistic

Category	**Normal**	**Anormal**
Train	1 611	397
Test	690	175

8.1 Normalization with duplication of the last beat

In have this strategy, we duplicate the last beat up to max ,by calculating (14 features * 173 maxbeat) length on all patients

The ensemble classifier, likely based on the Random Forest algorithm, achieved an overall accuracy of 89.60% on the test set. This high accuracy indicates that the model is effective at distinguishing between the two classes

Table 3.9: Performance de l'Ensemble Classifier

Metric	**Value**
Accuracy	89.68%
Precision	86.96%
Recall	57.14%
F1-score	68.90%

Table 3.10: Performance comparison of classification models

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
Decision Tree	0.81	0.53	0.52	0.52	0.70
Bagging	0.84	0.83	0.60	0.70	0.75
AdaBoost	0.86	0.85	0.62	0.72	0.78
Gradient Boosting	0.88	0.88	0.66	0.76	0.82

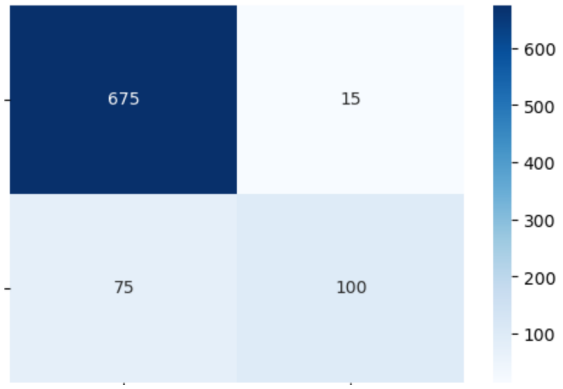


Figure 3.25: Confusion Matrices with random forest

8.2 Normalization with the duplication of whole Signal

In have this strategy, we duplicate complete signal up to max ,by calculating (14 features * 173 maxbeat) length on all patients.

The ensemble classifier, presumably utilizing the Random Forest algorithm, attained an overall accuracy of 88.9% on the test dataset. This strong performance demonstrates the model's effectiveness in accurately differentiating between the two classes.we summarize those Performance metrics of different classification models in table 3.10 .

Model	Accuracy	Precision	Recall	F1	ROC AUC
Random Forest	0.889017	0.792593	0.611429	0.690323	0.913408
Decision Tree	0.808092	0.526012	0.520000	0.522989	0.700580
Bagging	0.886705	0.829060	0.554286	0.664384	0.914124
AdaBoost	0.853179	0.710526	0.462857	0.560554	0.850066
Gradient Boosting	0.876301	0.778689	0.542857	0.639731	0.899594

Table 3.11: Performance metrics of different classification models

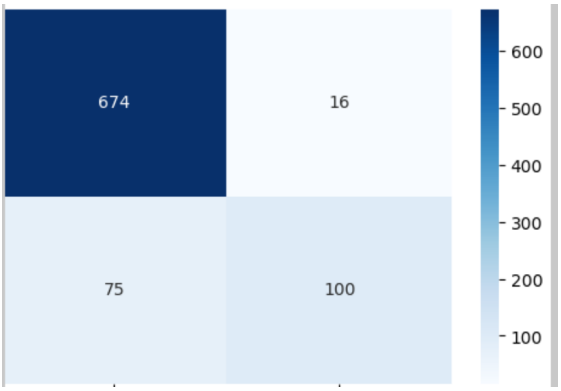


Figure 3.26: Confusion Matrices with random forest

8.3 Normalization with fixed threshold

In have this strategy,by calculating (14 features * length of threshold) length on all patients.

We evaluated the performance of the Random Forest classifier by adjusting various decision thresholds, as shown in Table 9. The accuracy varies slightly between 88.67% and 89.44% depending on the chosen threshold. Similarly, the average precision and average F1-score remain stable around 88–89%, indicating the model's robustness to threshold changes. The best balance between precision and recall appears at threshold around 75, where accuracy peaks at 89.44% with a correspondingly high F1-score. These results suggest that fine-tuning the decision threshold can optimize the Random Forest's performance without sacrificing stability.we summarize this in the table 3.11 .

Table 3.12: Classifier Performance at Different Thresholds

Threshold	Accuracy (%)	Avg Précision	Avg F1-score
20	88,79	88	88
25	88.67	88	88
30	88.79	88	88
35	89,36	89	88
40	89,13	89	88
45	88,67	88	88
50	88,90	89	88
55	89,01	89	88
60	89,13	89	88
65	88,90	89	88
70	89,25	89	88
75	89,44	89	89
80	89,25	89	88

Table 3.13: Performance metrics of different classification models with Threshold =55

Model	Accuracy	Precision	Recall	F1	ROC AUC
Random Forest	0.890173	0.807692	0.600000	0.688525	0.915337
Decision Tree	0.823121	0.569620	0.514286	0.540541	0.707867
Bagging	0.891329	0.858407	0.554286	0.673611	0.912923
AdaBoost	0.853179	0.710526	0.462857	0.560554	0.850066
Gradient Boosting	0.882081	0.806723	0.548571	0.653061	0.899081

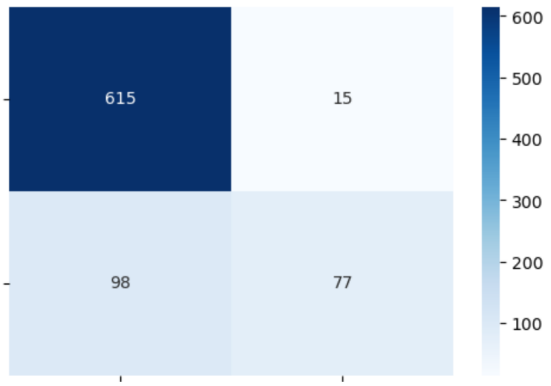


Figure 3.27: Confusion Matrices with random forest

8.4 Normalization with Attention-based dynamic strategy

The table 3.13 ,summarizes the performance metrics of various classifiers evaluated on the dataset. Metrics such as accuracy, precision, recall, F1-score, and ROC AUC are reported to provide a comprehensive comparison. Overall, the Random Forest and Gradient Boosting classifiers demonstrate strong accuracy and ROC AUC values, indicating good discrimination ability. However, recall scores remain relatively low across all models, suggesting challenges in correctly identifying positive cases. These results highlight the trade-offs between different classifiers and the importance of selecting the appropriate model based on the specific application needs.

Table 3.14: Performance metrics of different classification models with Attention-based dynamic strategy

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**
Random Forest	0.8428	0.8000	0.2971	0.4333	0.8601
Decision Tree	0.7873	0.4749	0.4857	0.4802	0.6747
SVM	0.8451	0.8154	0.3029	0.4417	0.8438
Bagging	0.8405	0.7403	0.3257	0.4524	0.8515
AdaBoost	0.8116	0.5811	0.2457	0.3454	0.7831
Gradient Boosting	0.8439	0.7273	0.3657	0.4867	0.8516

9 Conclusion

In this chapter, we presented a comprehensive methodology for classifying heart sound (PCG) signals as normal or abnormal, focusing on a patient-based processing strategy. The proposed system integrates several key stages: signal preprocessing, feature extraction, and model training.

To address variability in signal duration among patients, we implemented a standardized segmentation , ensuring consistent input size for the classifiers. Temporal features (such as mean amplitude, duration of cardiac cycles) were extracted to capture the physiological and acoustic properties of heart sounds.

We then introduced a hybrid deep learning model combining convolutional layers for spatial feature extraction, recurrent layers for temporal dynamics, and attention mechanisms to enhance discriminative power. This architecture was trained and evaluated on the PhysioNet/CinC Challenge 2016 dataset, enabling robust classification of PCG signals.

General Conclusion

This study developed a novel methodology for classifying heart sounds at the patient level, distinguishing between normal and abnormal cases using a combination of handcrafted temporal and morphological features and deep learning attention mechanisms. The integration of manual feature extraction (such as amplitude, duration of heart sound components, asymmetry, and flatness) with automatic high-level representation learning enhanced the classification accuracy and interpretability of the model.

The research successfully met its objective of creating a decision support tool for clinical use by processing entire PCG recordings rather than individual beats, which is a departure from many existing studies. The approach was rigorously evaluated using the PhysioNet/CinC 2016 dataset, which includes a wide variety of heart sound recordings from diverse patient populations and recording environments.

Several challenges were addressed throughout the study:

The dataset's inherent class imbalance, with a predominance of normal cases over abnormal ones, was carefully managed to avoid biased model training. Variability in recording environments and signal lengths required normalization strategies such as zero-padding and duplication of the last cardiac cycle to standardize input data for deep learning models. The segmentation of PCG signals into key components (S1, S2, systole, diastole) was crucial for effective feature extraction and classification.

Despite these achievements, certain limitations remain, including the variability in recording conditions and the imbalance in the dataset, which may affect the generalizability of the model. Future research could focus on expanding the dataset with more balanced and diverse samples, improving noise robustness, and exploring advanced feature extraction and deep learning techniques to further enhance classification performance.

In summary, this work contributes a comprehensive framework for automated heart sound classification that combines traditional signal processing with modern deep learning, offering promising potential for clinical decision support in cardiac diagnostics.

Bibliography

[1] C. Clinic, "Heart," n.d., accessed: Apr. 21, 2025. [Online]. Available: https://my.clevelandclinic.org/health/body/21704-heart

[2] J. Mulindi. (2023, Jun.) The instrumentation for recording ecg signals. Accessed: 2025-05-17. [Online]. Available: https://www.biomedicalinstrumentationsystems.com/the-instrumentation-for-recording-ecg-signals/

[3] A. Unknown, "Normal phonocardiogram signal for a single cardiac cycle," 2019, accessed: 2025-04-21. [Online]. Available: https://www.researchgate.net/figure/Normal-phonocardiogram-signal-for-a-single-cardiac-cycle_fig1_338803547

[4] V. N. Varghees and K. I. Ramachandran, "Illustrates the pcg signal including heart sounds (s1, s2, s3, s4)," https://www.researchgate.net/figure/llustrates-the-PCG-signal-including-heart-sounds-S1-S2-S3-S4-1-2-Measured-average_fig2_262806530, 2014, accessed: 2025-04-21.

[5] WGBH Educational Foundation. (2024) Describe a wave. Accessed: 2025-05-28. [Online]. Available: https://lsintspl3.wgbh.org/en-us/lesson/buac20-il-describewave/2?as_guest=True

[6] QuantInsti. (2024) Skewness - quantra glossary. Accessed: 2025-05-28. [Online]. Available: https://quantra.quantinsti.com/glossary/Skewness

[7] S. Intelligence. (2024, May) Support vector machines (svm): A comprehensive guide. Spot Intelligence. Online resource about Support Vector Machines. [Online]. Available: https://spotintelligence.com/2024/05/06/support-vector-machines-svm/

[8] GeeksforGeeks. (2023) Decision tree. [Online; accessed 16-May-2025]. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[9] DataScientest, “Bagging vs boosting: What is the difference?” n.d., accessed: May 16, 2025. [Online]. Available: https://datascientest.com/en/bagging-vs-boosting

[10] GeeksforGeeks. (2023) Random forest algorithm in machine learning. GeeksforGeeks. Online tutorial on Random Forest algorithm. [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

[11] IBM. (2024) Gradient boosting. Accessed: 2025-05-16. [Online]. Available: https://www.ibm.com/topics/gradient-boosting

[12] B. Öztornaci. (2024) Simplified structure of xgboost [figure 5]. Accessed: 2025-05-28. [Online]. Available: https://www.researchgate.net/figure/Simplified-structure-of-XGBoost_fig5_381857634

[13] GeeksforGeeks. (2023) Introduction to recurrent neural network. GeeksforGeeks. Online tutorial about Recurrent Neural Networks. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[14] O. Calzone, “An intuitive explanation of lstm,” https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c, 2022, accédé le 24 mai 2025.

[15] S. Nath, “Why is bilstm better than lstm?” https://medium.com/@souro400.nath/why-is-bilstm-better-than-lstm-a7eb0090c1e4, 2023, article de blog Medium, accédé le 24 mai 2025.

[16] Cleveland Clinic. (2023) Heart conduction system (cardiac conduction). [Online]. Available: https://my.clevelandclinic.org/health/body/21648-heart-conduction-system

[17] Johns Hopkins Medicine. (2023) Electrocardiogram. Johns Hopkins Medicine. Accessed: June 25, 2024. [Online]. Available: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electrocardiogram

[18] Encyclopædia Britannica, “Phonocardiography,” 2025, accessed: 2025-04-21. [Online]. Available: https://www.britannica.com/science/phonocardiography

[19] StatPearls Publishing, *StatPearls [Internet].* Treasure Island (FL): StatPearls Publishing, 2025, accessed: 2025-04-21. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK541010/

[20] Cleveland Clinic, “Heart: Anatomy & function,” https://my.clevelandclinic.org/health/body/21704-heart, 2025, accessed: 2025-04-21.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Entropy*, vol. 23, no. 6, p. 667, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/6/667

[22] OpenAI, “Chatgpt (gpt-4),” https://chat.openai.com, 2025, accessed: 2025-05-10.

[23] IBM. (2024) What is machine learning? Accessed: 2025-05-27. [Online]. Available: https://www.ibm.com/think/topics/machine-learning

[24] IBM, “What is a support vector machine (svm)?” https://www.ibm.com/think/topics/support-vector-machine, 2023, accessed: 2023-11-15. [Online]. Available: https://www.ibm.com/think/topics/support-vector-machine

[25] IBM, “What is bagging?” n.d., available at https://www.ibm.com/think/topics/bagging, Accessed: May 16, 2025.

[26] IBM, “What is random forest?” https://www.ibm.com/think/topics/random-forest, 2023, accessed: 2024-06-15. [Online]. Available: https://www.ibm.com/think/topics/random-forest

[27] F. M. Taufiq. (2023) Balanced random forest. Accessed: 2025-05-28. [Online]. Available: https://medium.com/@fadleemt/balanced-random-forest-d5dc9c896bb4

[28] XGBoost Developers, *XGBoost Documentation*, XGBoost, 2020, release 3.0.0. [Online]. Available: https://xgboost.readthedocs.io/en/release_3.0.0/

[29] IBM. (2024) What is deep learning? Accessed: 2025-05-27. [Online]. Available: https://www.ibm.com/think/topics/deep-learning

[30] IBM, “What are convolutional neural networks?” https://www.ibm.com/think/topics/convolutional-neural-networks, 2023, accessed: 2023-11-15. [Online]. Available: https://www.ibm.com/think/topics/convolutional-neural-networks

[31] Amazon Web Services. (2023) What is a recurrent neural network (rnn)? AWS. Online resource about Recurrent Neural Networks. [Online]. Available: https://aws.amazon.com/what-is/recurrent-neural-network/

[32] A. Vidhya. (2021, 03) Introduction to long short term memory (lstm). Analytics Vidhya. Online tutorial about Long Short-Term Memory networks. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

[33] IBM. (2025) What is data? Accessed: 2025-05-17. [Online]. Available: https://www.ibm.com/think/topics/data

[34] A. Della Corte, G. De Pietro, and L. Gallo, "Cnn-based classification of phonocardiograms using fractal techniques," *Biomedical Signal Processing and Control*, vol. 80, p. 104300, 2023.

[35] S. E. Schmidt, C. Holst-Hansen, J. Hansen, E. Toft, and J. J. Struijk, "Heart sound classification using deep learning," *Biomedical Signal Processing and Control*, vol. 42, pp. 146–157, 2018.

[36] F. Noman, S. Rehman, S. Islam, and S. Faul, "Ensemble of 1d and 2d convolutional neural networks for classification of heart sound signals," *Computers in Biology and Medicine*, vol. 100, pp. 24–33, 2018.

[37] J. Rubin, R. Abreu, A. Ganguli, and S. Nelaturi, "Classifying heart sound recordings using deep convolutional neural networks," *Computing in Cardiology*, vol. 44, pp. 1–4, 2017.

[38] H. Tang, T. Wan, D. He, and X. Wang, "Heart sound classification based on multiple domain features and svm," in *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 2018, pp. 5862–5866.

[39] C. Potes, S. Parvaneh, A. Rahman, and B. Conroy, "Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds," in *Computing in Cardiology*, vol. 43, 2016, pp. 621–624.

[40] Y. Zhang, Q. Wu, Y. Zhang, and B. Liu, "Segmental convolutional neural networks for detection of abnormal heart sounds," *Procedia Computer Science*, vol. 129, pp. 121–128, 2018.

[41] Y. Zhang, X. Li, T. Liu, and B. Liu, "Heartbeat sounds classification using sparse representation and svm," in *2016 Computing in Cardiology Conference (CinC)*. IEEE, 2016, pp. 373–376.