TECNOLOGIA SETÚBAL
ESCOLA SUPERIOR POLITÉCNICO SETÚBAL

SAÚDE
ESCOLA SUPERIOR POLITÉCNICO SETÚBAL

Catarina Alexandra dos Santos Cantante Martins Catalino

Theoretical Study of Multiple Heart Sounds and Identification Using Deep Learning

Master's Research Project Dissertation in Biomedical Engineering

SUPERVISOR

Professor José Inácio Pinto Rosado Rocha

December 2025

PRR Plano de Recuperação e Resiliência
REPÚBLICA PORTUGUESA
Financiado pela União Europeia NextGenerationEU

Catarina Alexandra
dos Santos Cantante
Martins Catalino

Theoretical Study of Multiple Heart Sounds and Identification Using Deep Learning

EXAMINATION COMMITTEE

Chairperson: Prof. Dr. Célio Gabriel Figueiredo Pina, Instituto Politécnico de Setúbal
Supervisor: Prof. José Inácio Pinto Rosado Rocha, Instituto Politécnico de Setúbal
Vogal: Prof. Dr. Rui Pedro Batoreo Amaral, Instituto Politécnico de Setúbal

December 2025

PRR Plano de Recuperação e Resiliência
REPÚBLICA PORTUGUESA
Financiado pela União Europeia NextGenerationEU

“Patience and perseverance have a magical effect before which difficulties disappear and obstacles vanish.”

— John Quincy Adams

PRR Plano de Recuperação e Resiliência
REPÚBLICA PORTUGUESA
Financiado pela União Europeia NextGenerationEU

Acknowledgments

I would like to express my sincere gratitude to my supervisor, José Rocha, for his guidance, dedication, and constant support throughout the development of this work. I would also like to thank Professor Miguel López for his availability and valuable assistance with the installation of Python libraries, which were fundamental for the development of this project.

I am deeply grateful to my family, especially my parents, not only for their encouragement and unconditional love, but also for their sacrifices and financial support, which made the completion of this academic cycle possible.

I also extend my thanks to my friends, who were always present to motivate me and provide moments of relaxation, helping me maintain balance throughout this journey.

This work would not have been possible without your support. Thank you very much.

Resumo

As doenças cardiovasculares (DCV) continuam a ser uma das principais causas de mortalidade a nível mundial, tornando essencial o desenvolvimento de métodos automáticos que apoiem o diagnóstico precoce. A análise de sons cardíacos é bastante utilizada na prática clínica por ser um método não invasivo, económico e de elevada relevância diagnóstica. No entanto, a interpretação da auscultação depende da experiência do examinador e é frequentemente afetada por ruído ambiental, variabilidade acústica e limitações na perceção de sons patológicos, o que pode conduzir a erros de diagnóstico.

Neste contexto, este trabalho apresenta o desenvolvimento de um sistema de classificação automática de sons cardíacos baseado em *deep learning*, utilizando redes neuronais convolucionais (CNN) aplicadas a representações tempo-frequência obtidas através da Transformada de Fourier de Curta Duração (STFT). Foram desenvolvidas duas abordagens distintas: uma para a classificação dos eventos fundamentais do ciclo cardíaco (S1, S2, sístole e diástole) e outra para a deteção binária entre sons normais e patológicos. Os sinais PCG da base de dados pública PhysioNet/CinC Challenge 2016 foram segmentados em janelas de eventos em segmentos de 2 segundos, a partir dos quais foram criados espectrogramas STFT, utilizados como entrada para diferentes arquiteturas de CNN. Para mitigar o desequilíbrio entre classes e aumentar a robustez do modelo, foram aplicadas técnicas de *data augmentation*, nomeadamente *additive noise* e *pitch shifting*.

Foram avaliados quatro modelos CNN, integrados com funções de ativação ReLU e GELU aplicadas entre as diferentes camadas, e ajustadas progressivamente através de sucessivas iterações de otimização de hiperparâmetros. O melhor modelo alcançou uma *accuracy* de 91,35%, um *recall* de 84,97%, uma precisão de 81,18%, uma especificidade de 93,47% e um *F1-score* de 83,43% na tarefa de classificação binária. Estes resultados superam abordagens tradicionais baseadas na extração manual de características e aproximam-se do desempenho de arquiteturas mais avançadas que realizam a extração automática de características de forma *end-to-end* através de redes convolucionais, tal como reportado na literatura.

Palavras-chave: Fonocardiograma, Classificação de sons cardíacos, Características tempo-frequência, Rede Neuronal Convolucional.

Abstract

Cardiovascular diseases (CVDs) remain one of the leading causes of mortality worldwide, highlighting the need for automatic methods that support early diagnosis. Heart sound analysis is widely used in clinical practice due to its non-invasive, low-cost nature and diagnostic relevance. However, auscultation is highly dependent on the examiner's experience and can be affected by environmental noise, acoustic variability and the subtle perception of pathological sounds, which may lead to diagnostic inaccuracies.

In this context, this work presents the development of an automatic heart sound classification system based on deep learning, using convolutional neural networks (CNN) applied to time–frequency representations obtained through the Short-Time Fourier Transform (STFT). Two distinct approaches were explored: one for the classification of the fundamental cardiac cycle events (S1, S2, systole and diastole) and another for binary discrimination between normal and pathological heart sounds. PCG signals from the public PhysioNet/CinC Challenge 2016 dataset were segmented into event windows and into fixed 2-second segments, from which STFT spectrograms were generated and used as input for different CNN architectures. To mitigate class imbalance and increase model robustness, data augmentation techniques were applied, namely additive noise and pitch shifting.

Four CNN models were evaluated, incorporating activation functions ReLU and GELU inserted between convolutional and dense layers, and progressively refined through iterative hyperparameter tuning. The best-performing model achieved an accuracy of 91.35%, recall of 84.97%, precision of 81.18%, specificity of 93.47% and an F1-score of 83.43% in binary classification. The best-performing model achieved an accuracy of 91.35%, recall of 84.97%, precision of 81.18%, specificity of 93.47% and an F1-score of 83.43% in binary classification. These results outperform traditional approaches based on handcrafted feature extraction, and approach the performance of more advanced end-to-end models that extract features automatically through convolutional architectures, as reported in the state of the art.

Keywords: Phonocardiogram, Heart sound classification, Time-frequency features, Convolutional Neural Network.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

A/D	Analog-to-digital
A2	Aortic valve
ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Programming Interface
AR	Aortic Regurgitation
AS	Aortic Stenosis
ASD	Atrial Septal Defect
ASY	Atrial Systole
AV	Atrioventricular
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CVDs	Cardiovascular Diseases
Cv2	OpenCV
CWT	Continuous Wavelet Transform
DC	Direct Current
DL	Deep Learning
DM	Diastolic Murmurs
DT	Decision Tree
ECG	Electrocardiogram

FIR	Finite Impulse Response
FN	False Negative
FP	False Positive
GAP	Global Average Pooling
GAs	Genetic Algorithms
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HMM	Hidden Markov Models
HSMM	Hidden semi-Markov Models
IIR	Infinite Impulse Response
IM	Innocent Murmurs
KNN	k-Nearest Neighbor
LPF	Low-Pass Filter
M1	Mitral valve
MFCC	Mel Frequency Cepstral Coefficient
ML	Machine Learning
MR	Mitral Regurgitation
MS	Mitral Stenosis
P2	Pulmonary valve
PCG	Phonocardiogram
PS	Pulmonary Stenosis
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit

RF	Random Forest
RI	Rapid inflow
RNN	Recurrent Neural Network
S1	First heart sound
S2	Second heart sound
S3	Third heart sound
S4	Fourth heart sound
SM	Systolic Murmurs
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
T1	Tricuspid valve
TN	True Negative
TP	True Positive
TR	Tricuspid Regurgitation
TS	Tricuspid Stenosis
VSD	Ventricular Septal Defect
Vs Code	Visual Studio Code

List of Symbols

s	Seconds
ms	Milliseconds
Hz	Hertz
x	Original signal
t	Time variable
n	Discrete-time index (sample)
$\delta(n)$	Discrete unit impulse
M	Million

Chapter 1
Introduction

This chapter provides a general overview of the topic addressed in this dissertation. It begins with a brief description of the clinical and technological context related to cardiac sound analysis. The main research problems are then identified, the study objectives are defined, and finally, the structure and organization of the document are presented.

1.1. Contextual overview

Cardiovascular diseases (CVDs) are the leading cause of death worldwide and are responsible for a significant number of deaths and disabilities. In 2021, they caused more than 20.5 million deaths, which corresponds to 32 % of all deaths registered worldwide, according to estimates by (Di Cesare et al., 2024). These pathologies are mainly the result of abnormalities in the electrical conduction of the heart, malformations of the heart valves and disorders of the heart muscle, and affect people of all ages, from newborns to the elderly (Benjamin et al., 2018).

The analysis of heart sounds is one of the most widely used methods for detecting CVD, with auscultation being the main technique used by healthcare professionals. Although it is a simple, inexpensive, and non-invasive procedure that has been used for over 200 years (S. Sathyanarayanan et al., 2023), its interpretation depends heavily on the examiner's experience. In some cases, sound quality can be impaired by ambient noise or very low intensity, making it difficult to perceive some relevant sounds. In addition, the complexity of sound patterns can make it difficult to distinguish between normal and pathological sounds, thus compromising the accuracy of the diagnosis (Chen et al., 2021; Tariq et al., 2022). Despite these limitations, heart sounds can be recorded using a phonocardiogram (PCG) (Tariq et al., 2022), a portable device that graphically represents the signals. Compared to an electrocardiogram (ECG), the PCG provides an economical alternative that allows for the interpretation and diagnosis of a clinical condition, directing the patient to more specific early treatments (Fattah et al., 2017; Tariq et al., 2022).

Recently, with the emergence of digital stethoscopes and mobile phones (Fattah et al., 2017; Liao et al., 2023), it has become possible to improve the quality of heart sounds using signal processing techniques and deep learning algorithms. These algorithms have shown the ability to increase diagnostic accuracy, improve analysis productivity, and overcome limitations associated with conventional auscultation, contributing to cost reduction and optimizing healthcare efficiency (Tariq et al., 2022). Recent studies indicate that deep learning models can significantly support clinical decision-making by providing systems capable of analyzing patterns, identifying changes, and making predictions similar to those of a specialist (Alhussein et al., 2019).

1.2. Problem statement

Although cardiac auscultation is widely used by healthcare professionals to detect CVD, it still has notable limitations. The quality of the examination can be affected by noise from the body itself, the surrounding environment, or interference from equipment, which may compromise the perception of low-frequency heart sounds. Such limitations can lead to incorrect diagnoses and, consequently, to unnecessary or inappropriate treatment.

Cardiac signals vary in morphology from patient to patient, influenced by factors such as age, physical activity, or the presence of associated diseases. This variability makes it challenging to interpret and to distinguish between normal and pathological sounds, affecting the accurate identification of the main components of the cardiac cycle (S1, S2, systole, and diastole)[1] and, in turn, the correct detection of clinical conditions.

Given these challenges, it is important to develop more reliable and objective analytical methods, capable of overcoming the limitations of traditional auscultation and enhancing the early detection of cardiac pathologies through advanced computational techniques.

1.3. Research goals

The main objective of this dissertation is to develop and evaluate a deep learning- based model capable of classifying multiple types of heart sounds and performing their identification with high accuracy.

The contribution of this work can be summarized in the following steps:

- Collection and preparation of heart sounds data from phonocardiograms obtained from different sources.
- Application of preprocessing techniques for noise removal and signal normalization.
- Exploration and selection of deep learning architectures suitable for classification, using Convolutional Neural Networks (CNN) to process images generated from different sounds representations, such as spectrograms.
- Implementation of data augmentation techniques, specifically noise addition and pitch shifting, applied to the dataset to improve model training and testing performance.

This dissertation also aims to address the following research questions:

- Among the different CNN parameterizations tested, which configuration provides the best balance between performance, generation, and computational cost?
- How does the performance of the proposed model compare to traditional heart sounds analysis methods and classical machine learning models?

[1] First heart sound (S1), Second heart sound (S2)

1.4. Document structure

The structure of this dissertation is organized into the following chapters:

- Chapter 1 – Introduction provides a brief description of the developed project, explaining the contextual overview, the problem statement, the research goals to be achieved, and finally, the overall document structure.
- Chapter 2 – Systematic Review presents the literature review, addressing the main concepts related to cardiac physiology and anatomy, the heart cycle, and heart sound auscultation. It also explores the theoretical framework associated with Artificial Intelligence and Neural Networks, with particular emphasis on activation functions and convolutional neural networks applied to biomedical signal analysis.
- Chapter 3 – Methodology describes the methodology developed in this work, including the characterization of the database, the data augmentation techniques, the implementation plan, and the steps of preprocessing, segmentation, and feature extraction. It also details the architecture of the CNN-based classification models developed throughout the study.
- Chapter 4 – Results and Evaluation present the results and the evaluation process. It describes the experimental procedure carried out and the processing applied to each dataset, both for the binary task (normal vs. pathological) and for the cardiac cycle phase classification (S1, S2, systole, and diastole), while highlighting the total number of parameters used as input to the neural network. Only the results of the binary classification (normal vs. abnormal) are reported, supported by performance metrics, learning curves, and the corresponding confusion matrices.
- Chapter 5 – Discussion is dedicated to the analysis and interpretation of the results, emphasizing their comparison with the state of the art and the identification of limitations.
- Finally, the Conclusion summarizes the main contributions of this work and the impact of the obtained results.

Chapter 2

Systematic Review

This chapter provides a brief overview of the physiological and anatomical aspects of the heart, as well as the fundamentals of the cardiac cycle. It also addresses the origin and clinical significance of heart sounds, with emphasis on the role of auscultation as a diagnostic tool. The main focus is on distinguishing normal heart sounds from additional acoustic events, such as gallop rhythms and murmurs, which may indicate pathological conditions.

2.1. Heart Physiology and Anatomy

The anatomy and physiology of the heart are intrinsically interlinked and are fundamental to understanding blood circulation and, consequently, the production of heart sounds. The heart has a complex structure, the function of which depends on the morphology of its cavities, valves and the electrical activity that regulates the cardiac cycle, made up of contraction (systole) and relaxation (diastole) phases (Trifunović-Zamaklar et al., 2022).

This organ is in the center of the thoracic cavity, slightly deviated to the left, between the two lungs, and has a pyramidal shape with its base facing upwards (Santo, A. 2016). The heart consists of four cavities: two atria (right and left) and two ventricles (right and left), as shown in Figure 2.1 (Santos & Lucindo, n.d.). The right atrium and right ventricle are responsible for receiving venous blood and conducting it to the lungs via the pulmonary circulation. In turn, the left atrium and left ventricle receive arterial blood from the lungs and distribute it throughout the body via the systemic circulation (Seeley et al., 2004; Miguel et al., 2011).

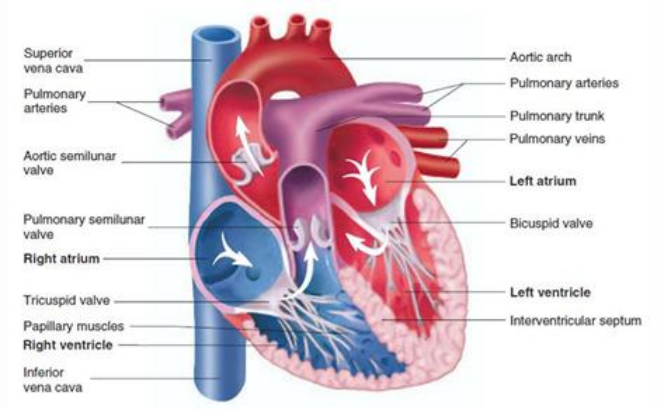

Figure 2.1 - Heart Physiology (Seeley, Rod R. et al., 2004).

The separation between the right and left heart cavities is ensured by the interarticular and interventricular septa, which prevent oxygen-rich blood from mixing with non-oxygenated blood, except in pathological conditions (Seeley et al., 2004; Miguel et al., 2011).

The blood flow inside the heart is regulated by four heart valves that ensure unidirectional flow (Santos & Lucindo, n.d.). These include two atrioventricular valves, namely the mitral (bicuspid) valve (which separates the left atrium from the left ventricle) and the tricuspid valve (which separates the right atrium from the right ventricle), and two semilunar valves, known as the aortic valve (which separates the left ventricle from the aorta) and the pulmonary valve (which separates the right ventricle from the pulmonary artery), as shown in Figure 2.2 (Ribeiras, 2022; Miguel et al., 2011).

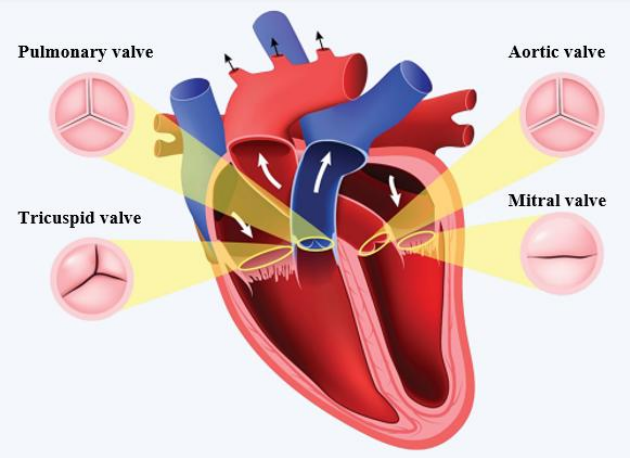

Figure 2.2 - Location of the four heart valves (Ribeiras, 2022) [Adapted].

The atrioventricular valves allow blood to pass from the atria to the ventricles. When they close, they produce the first heart sound S1, corresponding to the first sound of the cardiac cycle, also known as the *"lub"* of the *"lub-dub"* sound. The semilunar valves regulate the outflow of blood from the ventricles to the pulmonary artery trunk and the aorta. They open when intraventricular pressure increases and close when pressure decreases, generating the S2 heart sound, the second *"dub"* (S.Sathyanarayanan et al., 2023).

Analysis of cardiac activity is fundamentally dependent on the S1 and S2 heart sounds, as they represent the closing of the valves during the cardiac cycle and reflect the systolic and diastolic phases, respectively (Ríos-Prado et al., 2019).

2.2. Heart Cycle

The cardiac cycle consists of a coordinated sequence of changes in pressure, volume, and electrical activity, which are reflected in acoustic signals detectable through the phonocardiogram. These changes occur with each heartbeat and are regulated by the heart valves, which ensure the unidirectional flow of blood between the heart chambers and the great vessels (Hall & Guyton, 2011; Wright et al., 2020). This dynamic is graphically represented by the Wiggers diagram (Figure 2.3), which integrates Electrocardiogram (ECG) signals (depolarization and repolarization events) (Wright et al., 2020), pressure variations in the chambers and vessels, ventricular volume, and the phonocardiogram (PCG), allowing for a direct correlation between the electrical, mechanical, and acoustic events of the cardiac cycle.

The cardiac cycle consists of two major phases:

- Systole - corresponds to the period of contraction of the ventricles and subsequent ejection of blood into the great vessels (aorta and pulmonary artery) (Hall & Guyton, 2011);
- Diastole - represents the period of ventricular relaxation, during which the ventricles fill with blood from the atria (Hall & Guyton, 2011).

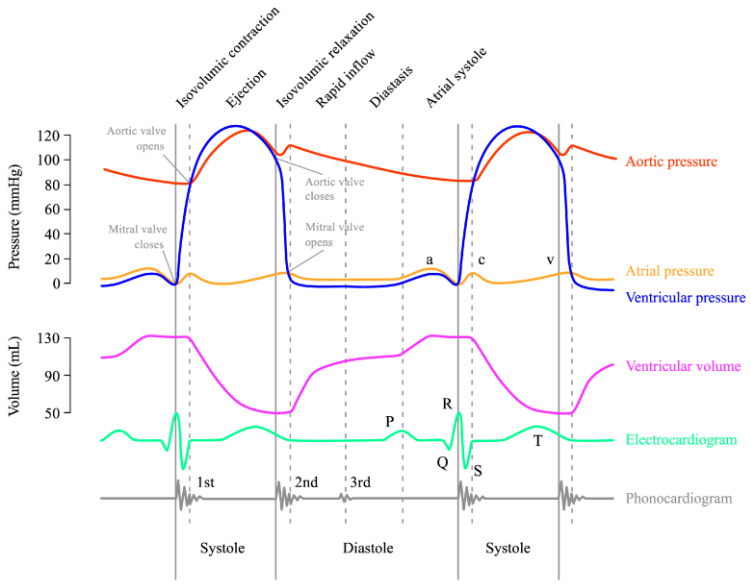

Figure 2.3 - Wiggers diagram (Singh et al., 2024).

The isovolumetric contraction phase marks the onset of ventricular systole, occurring simultaneously with the peak of the R wave of the QRS complex on the electrocardiogram. During this period, the mitral and tricuspid valves (atrioventricular valves) close, giving rise to the first heart sound (S1) recorded on the PCG (Rosas & Ayala, 2014). As all the valves remain closed, the ventricular volume remains unchanged, while intraventricular pressure rises rapidly (Morris et al., 2021).

When ventricular pressure exceeds that in the arterial system (aorta or pulmonary artery), the semilunar valves open, allowing blood to be ejected (Morris et al., 2021). This phase, known as ventricular ejection, can be subdivided into two stages (Sarti, 2023):

- Rapid ejection, characterized by simultaneous increase in ventricular and aortic pressure, accompanied by a sharp reduction in ventricular volume;
- Slow ejection, which follows the peak of the volume curve and coincides with the initial declines in aortic pressure.

This phase of the cardiac cycle concludes with the closure of the semilunar valves (Sarti, 2023), which produces S2. Under normal circumstances, the PCG remains acoustically silent during the ejection phase, except in the presence of pathological conditions that generate additional sounds.

Following this, isovolumetric relaxation begins, occurring immediately after the closure of the

semilunar valves. During this phase, the atrioventricular (AV) valves remain closed, preventing changes in ventricular volume. However, the ventricular pressure decreases rapidly, preparing the heart for the subsequent filling phase. This stage continues until atrial pressure exceeds the ventricular pressure, triggering the opening of the AV valves and allowing blood to flow from the atria into the ventricles (Morris et al., 2021).

Ventricular filling begins with the opening of the mitral and tricuspid valves and progresses through three distinct phases: rapid inflow (RI), diastasis and atrial systole (ASY).

During rapid inflow, the blood accumulated in the atria flows quickly into the relaxed ventricles due to the pressure gradient (Hall & Guyton, 2011). This phase is characterized by a decrease in atrial pressure and a sudden increase in ventricular volume. In specific cases, this flow can give rise to the third heart sound S3, audible on the phonocardiogram (Rosas & Ayala, 2014).

The subsequent phase, known as diastasis, involves slower filling, during which blood from the *vena cavae* and pulmonary veins continues to flow into the ventricles. The AV valves remain open, through changes in pressure and volume are less pronounced during this stage (Hall & Guyton, 2011; Morris et al., 2021).

Finally, ASY occurs because of atrial contraction following the P wave on the ECG. This contraction completes ventricular filling, causing a slight increase in atrial and ventricular pressures, as well as ventricular volume (Hall & Guyton, 2011). This phase signals the end of diastole and prepares the heart for the beginning of a new cardiac cycle (Sarti, 2023).

2.3. Heart Sounds and Auscultation

Cardiac auscultation remains one of the fundamental tools for the physiological assessment of the cardiovascular system. It is a simple, cost-effective, and non-invasive technique that has been employed for over two centuries to gather valuable information about cardiac dynamics (S.Sathyanarayanan et al., 2023). Although advancements in imaging technologies such as echocardiography have transformed diagnostic approaches, auscultation continues to play a key role (Abrams, 2005), particularly in the initial screening and monitoring of various cardiac conditions. Despite requiring continuous training and clinical experience by health professionals, this technique is still used because of its effectiveness and low cost (Chizner, 2008).

2.3.1. Cardiac auscultation

Heart sounds are generated by pressure changes within the heart chambers, the movement (opening and closing) of the cardiac valves, and turbulent blood flow. These sounds propagate through the cardiac and vascular structures until they reach the thoracic surface, where they can be auscultated using a stethoscope. S1 and S2 are the fundamental heart sounds and define the boundaries of systole and diastole. Proper auscultation should be performed in a quiet environment with the patient in a comfortable position - preferably supine, left lateral decubitus or sitting - depending on the area of the heart being assessed (Pazin-Filho et al., 2004).

The stethoscope is used to auscultate the four traditional cardiac valve areas, positioned as

shown in Figure 2.4 (Pazin-Filho et al., 2004; Rosas & Ayala, 2014):

- Aortic valve (A): second right intercostal space, close to the sternum;
- Pulmonary valve (P): second left intercostal space, close to the sternum;
- Tricuspid valve (T): left parasternal line, at the fourth intercostal space;
- Mitral valve (M): cardiac apex, at the fifth intercostal space, in the left midclavicular line.

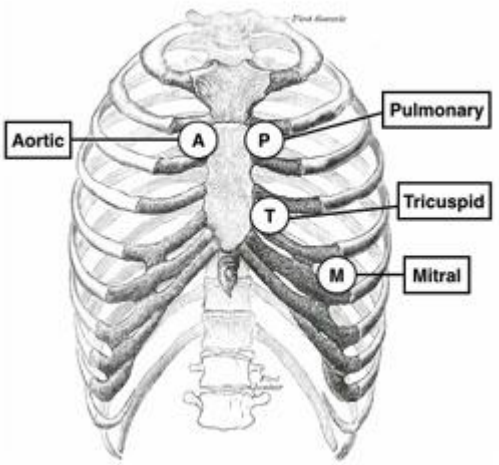

Figure 2.4 - Auscultation of the different heart valves (Mallinson, 2017).

There are currently several types of stethoscopes, from acoustic (traditional) models to the latest electronic ones. However, as there is still no certainty about the accuracy and reliability of electronic models, even though they are more expensive, traditional stethoscopes continue to be the most widely used in clinical diagnosis.

The recurrent auscultation technique of the acoustic stethoscope is based on the choice of the most appropriate auscultation component: the bell, with its hollow structure, is suitable for low-frequency (high-pitched) sounds(Santos, 2023) and should be applied with light pressure to the surface of the chest (Pazin-Filho et al., 2004); the diaphragm, on the other hand, is more effective in detecting high-frequency (low-pitched) sounds (Santos, 2023) and should be pressed firmly against the chest wall. This distinction is based on the acoustic properties of each component, and their choice directly affects the quality of the auscultation (Pazin-Filho et al., 2004).

The main aim of auscultation is to characterize the heart rhythm, identify the main sounds, detect additional sounds (S3 and S4) and detect the presence of murmurs, which may indicate changes in heart function (Pazin-Filho et al., 2004).

2.3.2. Normal heart sounds

Heart sounds reflect the mechanical interactions between blood flow and the heart valves, allowing for the identification of their intensity (amplitude), frequency (pitch), quality (timbre), and timing within the cardiac cycle (duration). These characteristics make it possible to assess the functional status of the heart valves (S. Li et al., 2020; Pazin-Filho et al., 2004). Sounds can be classified as normal or abnormal, including murmurs, gallop rhythms, and clicks (such as rubs

and ejection clicks), which may indicate pathological changes (Dwivedi et al., 2019).

The first heart sound (S1) corresponds to the closure of the mitral (M1) and tricuspid (T1) valves, marking the onset of ventricular systole (Dwivedi et al., 2019). This sound results from the vibration of the valves and adjacent cardiac structures and is characterized by a relatively high frequency. It is generally most audible at the mitral area, as described above in Section 2.3.1, and typically ranges between 10 and 200 Hz, with the M1 and T1 components occurring approximately 20 to 30 milliseconds (ms) apart (Reed et al., 2004).

The second heart sound (S2) marks the end of systole and the beginning of diastole and is produced by the closure of the aortic (A2) and pulmonary (P2) valves (Abrams, 2005). This sound is louder than S1, with frequencies ranging from 20 to 250 Hz and a duration between 20 and 150 ms and is most clearly heard at the base of the heart (Abrams, 2005; Dwivedi et al., 2019). The vibrations of S2 reflect the sequential closure of the A2 and P2 valves, whose individual components have an average duration of less than 60 ms. The interval between the closure of the two valves can vary between 30 and 80 ms, depending on physiological conditions (Dwivedi et al., 2019).

2.3.3. Additional heart sounds

Additional cardiac sounds may be heard throughout the cardiac cycle, in addition to the normal S1 and S2 sounds. These extra sounds may be associated with either physiological variations or pathological conditions such as heart disease. In the literature, gallop rhythms and murmurs are among the most frequently described and clinically relevant additional sounds, not only because of their higher prevalence in patients with cardiac pathologies, but also due to their diagnostic value. For this reason, these sounds are the ones detailed below.

Gallop rhythm

The gallop rhythm is primarily characterized by the presence of the third (S3) and fourth (S4) heart sounds and may be associated with various cardiac alterations. S3 is a low-frequency sound (25-50 Hz) that occurs at the beginning of diastole, immediately after the second heart sound (S2). It is a brief, transient sound, often described as a “ventricular gallop” or “early diastolic gallop.” Its presence is commonly linked to conditions such as heart failure or diastolic dysfunction of the left ventricle, although it may be considered physiological in children, adolescents, and young adults. However, its detection in individuals over 40 years of age may indicate underlying pathology. S4 also presents as a low-frequency sound (20-30 Hz) that occurs before the first heart sound (S1), during the presystolic phase. This sound is typically referred to as an “atrial gallop” or “presystolic gallop” and may be associated with decreased ventricular compliance, which could indicate conditions such as myocardial ischaemia (Pazin-Filho et al., 2004; Pechetty & Nemani, 2020).

Murmurs

Heart murmurs are typically caused by turbulent blood flow, which may result from the narrowing or insufficiency of heart valves, or from the abnormal passage of blood through the heart(Chebil et al., 2007). Murmurs can be classified into three main categories: innocent murmurs (IM), systolic murmurs (SM), and diastolic murmurs (DM). Figure 2.5 illustrates the main types of heart sounds associated with different pathologies, highlighting those that have been most frequently studied and clinically diagnosed (Choi & Jiang, 2010).

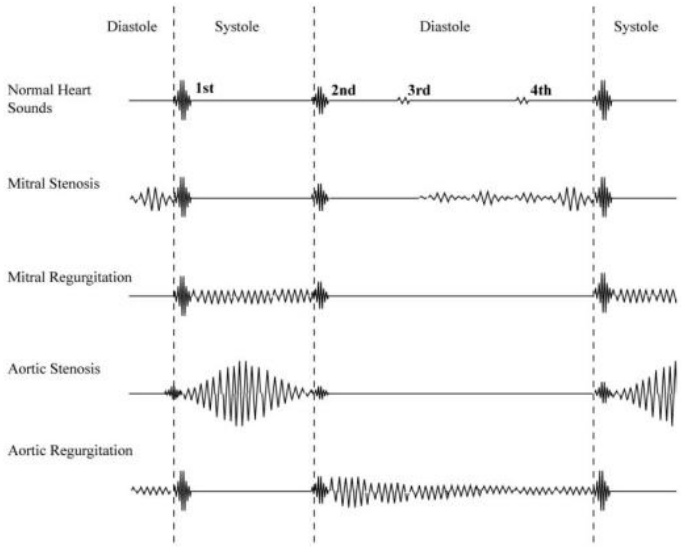

Figure 2.5 - Representation of heart murmurs recorded through the technique of auscultation (Wong, 2014).

Aortic stenosis (AS) refers to the narrowing of the aortic valve. Auscultation reveals a characteristic “crescendo-decrescendo” systolic murmur. Aortic regurgitation (AR) occurs when the aortic valve does not close properly, allowing blood to flow back into the left ventricle. This condition produces a high-pitched sound best heard along the left sternal border, described as a decrescendo diastolic murmur (Thomas et al., 2024). Mitral stenosis (MS) results from the narrowing of the mitral valve, generating a low-pitched sound often accompanied by an opening click. Auscultation is characterized by a diastolic murmur. Mitral regurgitation (MR) occurs when the mitral valve fails to close completely, resulting in a holosystolic murmur that is best heard at the apex of the heart. Pulmonary stenosis (PS) is characterized by an ejection systolic murmur caused by narrowing of the pulmonary valve. This condition is common in infants with tetralogy of Fallot, a congenital heart defect present from birth (José Roquette, 2023). The resulting murmur follows a “crescendo-decrescendo” pattern, similar to that of aortic stenosis, but is more commonly heard at the upper left sternal border. Tricuspid stenosis (TS) is marked by a systolic murmur due to narrowing of the tricuspid valve, often auscultated at the lower left sternal border (Thomas et al., 2024). Tricuspid regurgitation (TR) is a systolic murmur caused by the tricuspid valve’s inability to close properly, allowing blood to flow backward from the right ventricle to the right atrium. The characteristic murmur is typically heard at the lower left sternal border (Thomas

et al., 2024). Finally, there are two main types of septal defects: atrial septal defect (ASD) and ventricular septal defect (VSD). ASD is a congenital defect that allows free communication between the atria and is usually subtle on auscultation. VSD, on the other hand, is characterized by a loud holosystolic murmur, primarily heard during systole, with greater intensity at the apex (Thomas et al., 2024).

2.4. Artificial Intelligence and Neural Networks

This section provides a general overview of the main concepts related to artificial intelligence (AI) and neural networks. It begins with a brief introduction to AI, machine learning (ML) and deep learning (DL), followed by a description of the activation functions commonly used in neural networks and adopted in this work. Finally, the classification algorithm based on convolutional neural networks (CNN) implemented in this study is presented.

2.4.1. Introduction to Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science focused on developing systems capable of performing tasks that simulate human cognitive functions, such as reasoning, pattern recognition and clinical decision support (Russell and Norvig, 2021). AI involves the use of algorithms and large volumes of data to model intelligent behavior, enabling for example the identification of complex patterns in medical imaging or evidence-based decision making (Yu et al., 2018). Within this field, as illustrated in Figure 2.6, we find the subdomains of Machine Learning (ML) and Deep Learning (DL), which have contributed to significant progress in data processing and the development of predictive models.

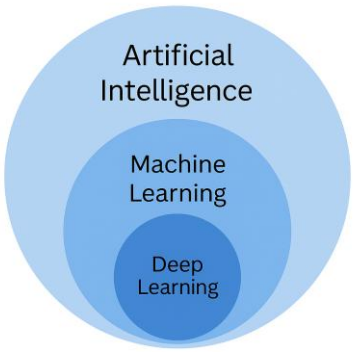

Figure 2.6 - Venn diagram representing the relationships between AI, ML and DL.

Machine Learning

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms capable of learning patterns from data without the need for explicit programming. An ML algorithm can learn from experience by adjusting its internal parameters to improve performance on a specific task (M.Mitchell, 1997).

In traditional ML approaches, feature engineering is often employed, where experts manually select relevant characteristics from the input data. In the context of biomedical signal

analysis, these features can be extracted from different domains: the time domain (such as amplitude, duration, or Shannon entropy), the frequency domain (for example, using the Fast Fourier Transform), or the time-frequency domain (through techniques such as Short-Time Fourier Transform (STFT), Mel-Frequency Cepstral Coefficients (MFCC), or Continuous Wavelet Transform (CWT)) (Rath et al., 2022; Taneja et al., 2023).

The extracted features are then used as input for algorithms that perform classification or regression tasks. Among the most widely used algorithms are Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Hidden Markov Models (HMM), Decision Trees (DT), and Random Forests (RF) (Dwivedi et al., 2019).

There are four main types of learning models, but only the two most applied in this context are considered here. In supervised learning, the training data is labelled, meaning that it includes defined input-output relationships. The model learns to generalize from these examples to predict new, unseen cases. In contrast, unsupervised learning is applied to unlabeled data, where the algorithm independently searches for patterns or underlying structures, such as clusters or distributions (M. Bishop, 2006).

Despite its usefulness, one of the main limitations of ML is its reliance on feature engineering. This process requires domain expertise, involves careful pre-selection of relevant attributes, and may fail to capture the full complexity and variability of the data (LeCun et al., 2015). Additionally, the performance of traditional ML models often degrades when dealing with large-scale datasets, noise, or heterogeneous signals, as is frequently the case in clinical environments (Litjens et al., 2017).

Deep Learning

Deep learning represents an evolution of machine learning and involves the use of artificial neural networks (ANNs), inspired by the functioning of the human brain. In these models, artificial neurons simulate the way biological neurons communicate. These networks are composed of one or more hidden layers that enable the learning of hierarchical data representations, unlike traditional approaches that rely more heavily on manual intervention. This process allows the model to automatically extract discriminative features from the input data, reducing the need for feature engineering and supporting autonomous learning of relevant patterns.

Deep networks are structured into layers of artificial neurons, typically organized into three types: an input layer, one or more hidden layers, and an output layer, as illustrated in Figure 2.7. Information flows through the network in a unidirectional manner, from input to output, in a process known as *feed-forward* (GeeksforGeeks, 2025). In each neuron, a *weighted sum* of the inputs is calculated, where each input is multiplied by its respective weight. A *bias* term is added to this sum, allowing the neuron to produce a non-zero output even when all inputs are zero. The resulting value is then passed through an *activation function*, and the output is transmitted to the next layer (GeeksforGeeks, 2025; Zilliz, 2025).

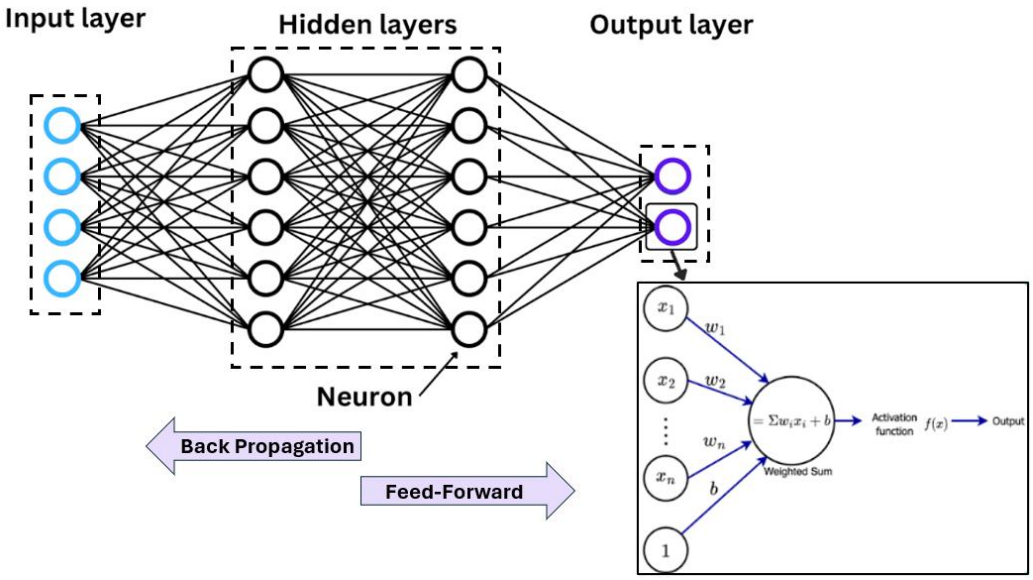

Figure 2.7 - Neural network architecture and inner functioning of a neuron (Zilliz, 2025) [adapted].

During training, the network repeatedly adjusts its internal weights based on the prediction error. This process uses the *backpropagation* algorithm, which propagates the error from the output layer back through the previous layers. To achieve this, the gradient of the loss function is calculated with respect to each weight, applying the chain rule of derivatives. The gradient indicates how and in which direction the weights should be updated to correct the parameters and reduce the loss. This iterative adjustment improves the network's performance and its ability to generalize (Bergmann & Stryker, 2024; LeCun et al., 2015).

2.4.2. Activation function

As described above, activation functions are elements present in neural networks which introduce non-linearity into models, allowing them to learn complex representations and model non-linear relationships between data (Zilliz, 2025). The activation functions adopted in this work are ReLU (Rectified Linear Unit), GELU (Gaussian Error Linear Unit), Sigmoid and Softmax, which are briefly described below.

ReLU

The ReLU function is currently the most common in the hidden layers of neural networks due to its computational simplicity and effectiveness in accelerating the learning process (Elola et al., 2023; F. Li et al., 2020). It cancels all negative activations by returning *"zero"* and keeps positive values unchanged. This function is mathematically expressed as:

$$f(x) = \max\ (0, x) \tag{2.1}$$

During activation, only some of the neurons remain active, which improves the quality of the network. However, a key issue associated with this function is the *"dead neuron"*, whereby units

stop learning because they constantly receive negative values during training, rendering them permanently inactive. Nevertheless, ReLU continues to be used, including in this work, and it is particularly effective in deep architectures, such as convolutional networks (Chaudhary, 2020; GeeksforGeeks, 2025).

GELU

The Gaussian Error Linear Unit (GELU) is a smooth and differentiable activation function that can be interpreted as a continuous and probabilistic approximation to the ReLU function. In contrast to ReLU, which eliminates all negative values, GELU applies a smooth transformation where each input value is weighted by the probability that a random variable with a standard normal distribution takes a value less than the input itself (Hendrycks & Gimpel, 2023; Lee, 2023). This behavior preserves small negative activations and improves gradient flow during network training. The GELU function is mathematically defined as:

$$f(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right) \tag{2.2}$$

Sigmoid

The sigmoid function is used in the output layer of binary classification tasks because it compresses values into the range [0, 1], enabling them to be interpreted as probabilities (Chaudhary, 2020; F. Li et al., 2020). Formally, the function can be expressed as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Softmax

The softmax function is a generalization of the sigmoid function and is used for multiclass problems. This activation function transforms a vector of real values into normalized probabilities, the sum of which is equal to 1, and is the standard choice for the output layer of multiclass classification networks (GeeksforGeeks, 2025; Tariq et al., 2022). The analytical expression of the softmax function is defined as:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}, i = 1,2, \dots N \tag{2.4}$$

where x_i denotes the logit corresponding to the output of the previous layer for the i -th class, and *N* represents the total number of output classes. The e^{x_i} applies the exponential function to each logit, while the $\sum_{j=1}^{N} e^{x_j}$ performs the normalization across all classes, ensuring that the final outputs form a valid probability distribution.

2.4.3. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are one of the most widely used models in deep learning, and are generally trained in a supervised manner, which simplifies the learning process and improves training efficiency. This architecture is particularly relevant in signal and image analysis, as it can exploit data with spatial (2D) or temporal (1D) structure, such as medical images or biomedical signals. Figure 2.8 illustrates the typical architecture of a CNN, where input information is processed by convolution layers, followed by pooling layers, until it reaches a fully connected layer, responsible for generating the final prediction in the output layer.

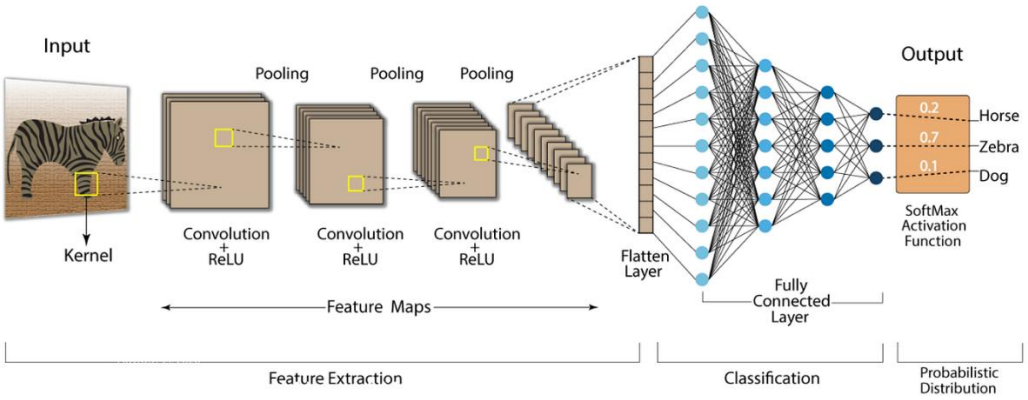

Figure 2.8 - The Architecture of the Convolution Neural Network (Rabiza, 2024).

Convolutional layers apply learned filters (kernels) that slide over the input, performing local weighted sums to produce feature maps that emphasize patterns such as edges or textures .Each layer is controlled by hyperparameters such as *stride* (filter step), *padding* (input edge adaptation), and *depth*, which defines the number of filters applied (Teoh, 2023).

Pooling reduces dimensionality in CNN by condensing the information extracted by convolutions, making the network more efficient and less sensitive to variations, such as pattern positions. Within this technique, there are two possible approaches: *max pooling*, which retains the maximum value of each analyzed region, and *average pooling*, which calculates the average of the values in that same region (Teoh, 2023). A visual example of both pooling operations is provided in Figure A.1, where a) illustrates max pooling and b) represents average pooling.

The *flatten layer* transforms the multidimensional output of the convolution and pooling layers into a one-dimensional vector. The extracted information is compatible with the fully connected layers, allowing the network to learn nonlinear combinations of the detected features (Teoh, 2023).

Finally, the *fully connected layer* combines all the previously extracted features to form a final decision by the network, through a probability distribution (output).

To prevent overfitting[2] and improve generalization, dropout layers are often integrated after the convolutional stages and before the fully connected layer. These layers randomly deactivate

[2] The model memorizes the training data instead of learning new patterns.

the contribution of certain neurons during training, forcing the network to learn features that are less dependent on specific combinations of active units (Srivastava et al., 2014). This behavior can be observed in Figure A.2.

2.5. Evaluations

This section presents and describes the evaluation methods adopted in this project. The techniques detailed below were used to assess and validate the performance of the CNN models.

2.5.1. Hold-out

The hold-out methodology consists of dividing the dataset into independent subsets, typically for training and testing (Bami et al., 2025), and optionally including a validation set. The training set is used to fit the model parameters, the validation set is employed to optimize the hyperparameters, and the test set is reserved exclusively for the final evaluation of the model on unseen data.

2.5.2. Confusion Matrix

The confusion matrix is a requirement used in evaluating the performance of classification algorithms. It is presented in the form of a table (Table 2.1), comparing the actual classes and the classes predicted by the model. It can be applied to binary or multi-class classification problems (eg. 4x4 matrix), allowing the number of correct and incorrect predictions for each class to be identified, to inform the expert about the performance and errors of the model.

Table 2.1 - Confusion Matrix.

		Predicted Class	
		Negative	Positive
Actual Class	Negative	True Negative (**TN**)	False Positive (**FP**)
	Positive	False Negative (**FN**)	True Positive (**TP**)

Based on the confusion matrix, the following terms are defined (Emanuel, 2019):

- True Negative (TN): Corresponds to situations where the model accurately predicts the absence of the class under analysis. *Example:* the heart sound is abnormal, and the model correctly classifies it as abnormal.
- False Positive (FP): Describes cases in which the model erroneously indicates the presence of the target class when it is absent. *Example:* the heart sound is normal, but the model classifies it as pathological.
- True Positive (TP): Refers to instances where the model correctly predicts the presence of the target class. *Example:* the heart sound contains a pathological murmur, and the model classifies it as pathological.
- False Negative (FN): Occurs when the class exists in the actual set, yet the model

fails to identify it. *Example:* the heart sound contains a pathological murmur, but the model classifies it as normal.

2.5.3. Performance Metrics

Accuracy

Accuracy is defined as the proportion of correct classifications made by the model relative to the total number of predictions. The formula is shown in Equation 2.5.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

Precision

Precision is a measure of the confidence placed in the model's positive predictions. It measures the proportion of positive cases that were correctly classified. Equation 2.6 describes this calculation.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.6}$$

Recall or Sensitivity

Recall, also known as sensitivity, is a metric that can correctly identify positive cases. It shows the proportion of positive cases that were correctly identified out of the total number of cases in that class. The corresponding formula can be found in Equation 2.7.

$$\text{Recall/Sensitivity} = \frac{TP}{TP + FN} \tag{2.7}$$

Specificity

Specificity measures the proportion of truly negative cases that the model correctly identifies as negative. The mathematical expression is presented in Equation 2.8.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.8}$$

F1-score

The F1-score metric is the weighted average of precision and recall, as shown in Equation 2.9.

$$\text{F1} - \text{Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.9}$$

Chapter 3

Proposed Methodology

This chapter describes the methodology developed for the implementation of the heart sound classification system. It presents the procedures adopted from data preparation to model construction and training. These include the characterization of the database used, the applied data augmentation techniques, the implementation plan, the preprocessing and segmentation stages, the feature extraction process, and finally, the proposed classification models.

3.1. Database Description

The database used in this dissertation comes from the PhysioNet/CinC Challenge 2016, which provided a public repository aimed at promoting the development of automatic algorithms for heart sound analysis and classification. This repository has become one of the most comprehensive references in the field, due to the diversity, volume, and quality of the collected recordings (Liu et al., 2016).

The dataset comprises 3 240 phonocardiogram (PCG) recordings acquired from 764 patients (with and without heart disease) in both clinical and non-clinical environments (e.g., home settings). These recordings were collected across six independent databases, provided by research teams from various countries. Data acquisition took place at different auscultation sites (e.g., aortic, pulmonary, tricuspid, and mitral areas) using a variety of electronic stethoscopes (e.g., 3M Littmann 3200, Allyn Meditron, among others), and under varying noise conditions, thereby enhancing representativeness and clinical realism (Liu et al., 2016).

The recordings range from 5 seconds to just over 120 seconds and all were resampled to 2000 Hz, stored in .wav format, containing only the PCG signal (except for the training-a subset, which also includes ECG). The participants range in age from children to elderly individuals and include clinically validated diagnoses such as mitral valve prolapse, mitral regurgitation, aortic stenosis, prior valve surgery, and coronary artery disease, among others (Liu et al., 2016).

The database is structured into six training subsets (A-F), comprising:

- A total of 3 240 recordings, of which 2 575 are classified as normal and 665 as abnormal;
- Detailed annotations of cardiac states (S1, systole, S2, diastole), initially obtained using the algorithm proposed by (Springer et al., 2015) and manually corrected by a cardiologist, resulting in a total of 84 467 annotated cardiac cycles.

The recordings present varying levels of noise, including stethoscope movement, respiratory and intestinal sounds, which reinforces the suitability of this database for real-world scenarios with limited clinical supervision. As such, it stands out for the volume and heterogeneity of its data (Liu et al., 2016).

Table 3.1 shows the distribution of the different data subsets provided by the PhysioNet/CinC Challenge 2016. For each subset (training-a to training-f), we list the numbers of normal and abnormal recordings and the subset total; we also provide the overall total across all subsets. This overview contextualizes the PCG dataset (Physionet/CinC Challenge 2016) employed in this dissertation.

Table 3.1 - Description of the PhysioNet/CinC 2016 challenge dataset.

Data	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	1 958	183	2 141
Training-f	80	34	114
Total	2 575	665	3 240

3.2. Data Augmentation

The use of deep learning models requires large and diverse datasets to ensure robust pattern learning and to reduce overfitting. In the context of heart sound classification, this requirement becomes even more relevant due to the limited size and class imbalance present in PhysioNet datasets. Therefore, data augmentation is not merely a theoretical recommendation but an empirically validated approach that enhances model performance by artificially expanding the diversity of training samples (Jain, 2024; Tariq et al., 2022)

Based on this evidence, data augmentation was integrated from the outset rather than training a baseline model without it, since the literature consistently demonstrates that augmented datasets significantly improve generalization in audio-based classification tasks (Tariq et al., 2022; Torres, 2021). In this work, two techniques were applied, additive noise and pitch shifting, both considered suitable for doubling the number of training samples while introducing meaningful acoustic variations without compromising the integrity of the cardiac signals.

3.2.1. Additive Noise

To mitigate overfitting, we added controlled noise to the PhysioNet heart-sound recordings during training. Each original segment signal window was linearly combined with a randomly selected noise segment. As proposed by (Tariq et al., 2022), this combination was performed using a random coefficient α, extracted from a uniform distribution $\cup$ [0.001, 0.005], as shown in Equation (3.1).

$$\alpha \sim \cup [0.001, 0.005]$$
$$x' = (1 - \alpha).x + \alpha . x_{noise} \quad (3.1)$$

Where, x' represents the new signal resulting from the combination of the original signal x and a noise segment x_{noise}, with the proportion determined by the value of α.

Figure 3.1 illustrates the effect of the random noise on both normal and abnormal signals, enhancing the model's robustness to real-world variations in the input data. Additionally, a 1-second sample image was included to better visualize the subtle noise additions applied to the signals, which are very small and not easily perceptible in shorter time frames.

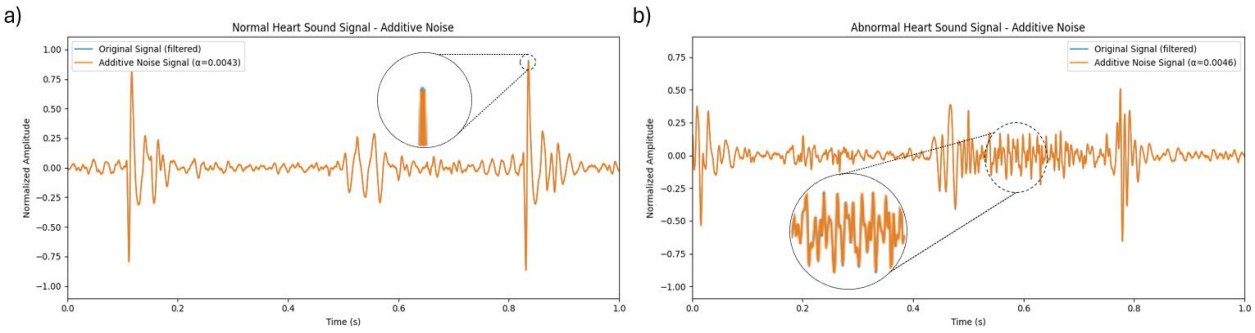

Figure 3.1 - Application of the additive noise technique to two cardiac sound signals from the PhysioNet database. Subfigure a) shows the normal signal (a0080.wav) before and after the addition of noise, while subfigure b) presents the abnormal signal (a0002.wav) subjected to the same process.

3.2.2. Pitch Shifting

Additionally, we used pitch shifting, further enriching the training data. This technique makes a slight adjustment to pitch (raising or lowering the signal's fundamental frequency) without changing its timbre, duration, or cardiac content (Tsang,2023).

In this case, a semitone[3] of +1 was applied, increasing the frequency of the original signal while keeping its duration unchanged. As shown in Figure 3.2, this results in a new sample with a distinct tonal quality, while preserving all relevant features of the original phonocardiogram (PCG) signal.

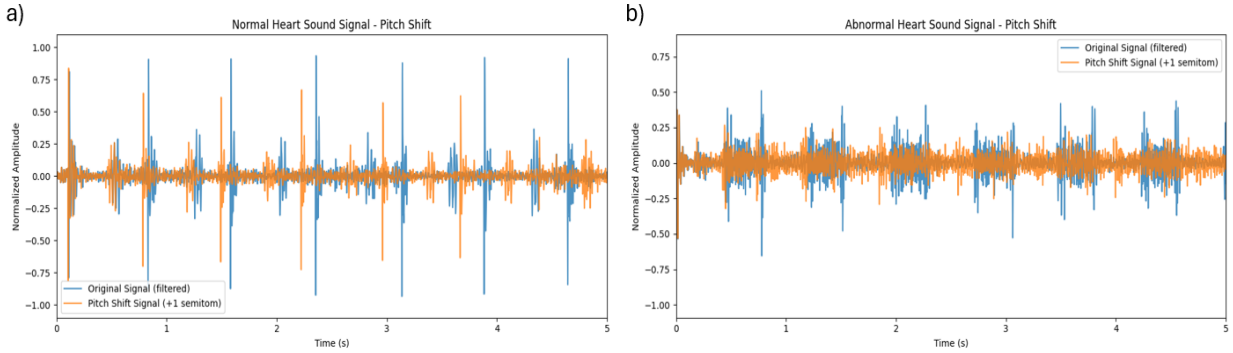

Figure 3.2 - Application of the pitch shift technique to two cardiac sound signals from the PhysioNet database. Subfigure a) shows the normal signal (a0080.wav) before and after the application of pitch shift, while subfigure b) presents the abnormal signal (a0002.wav) processed in the same way. The transformation was performed with a +1 semitone increase, resulting in a slight upward shift of the signal's frequency components while maintaining its temporal duration.

[3] Corresponds to a scaling factor of $2^{\frac{1}{12}} \approx 1.059$, increasing all frequencies by 5.94%.

3.3. Implementation and System Design Plan

This section describes the development environment and tools used throughout the work, as well as the execution plan and steps followed in implementing the system. It provides the reader with a structured overview of the methods applied and methodological decisions adopted.

3.3.1. Implementation language

Python

The Python programming language was used in this project for signal preprocessing, segmentation, feature extraction, construction of visualization tools, and cardiac pathology prediction. Python provides a wide range of modules and frameworks that support signal processing, machine learning, and deep learning algorithms, making it an appropriate choice for projects of this nature. Visual Studio Code (VS Code)[4] with the python extension was employed to develop the scripts.

Python Libraries

Several Python libraries were used to develop the scripts implemented in this project. The integration of these libraries allowed for the reuse of functions, simplification of repetitive tasks, and acceleration of the development process. The installation was performed in an environment created using Anaconda software, using the Anaconda PowerShell Prompt terminal.

The Pandas library was used for reading, manipulating, and transforming tabular data, namely for files in *.csv* format. It also allowed operations such as filtering, removing missing values, and organizing data sets into data frames (Chugh, 2025). In addition, the NumPy library was responsible for supporting multidimensional array data structures (Mckinney, 2017), which are important for performing mathematical operations and numerical calculations on vectors and matrices.

Signal analysis was supported by the SciPy library, which provides advanced tools, particularly for signal filtering. The Librosa library was used to manipulate and extract features from audio signals, such as phonocardiograms. This also allowed *.wav* files to be loaded, representations such as spectrograms to be produced, and data augmentation techniques such as pitch shift to be applied.

Data and results were visualized using Matplotlib, a library that allows the creation of static and interactive graphs (Hunter, 2007). Scikit-learn (sklearn) was used for operations in the deep learning pipeline, such as data normalization and balancing, division into training, validation, and test sets, and calculation of model evaluation metrics.

Finally, OpenCV (cv2) was used to process images in *.png* format generated from the signals, which was relevant in modifying spectrograms into visual representations for later use in convolutional neural networks.

[4] Developed by Microsoft.

Deep Learning Frameworks

In implementing deep learning models, the TensorFlow and Keras frameworks were used, which are known in the literature for their effectiveness in building and training deep neural networks.

TensorFlow, developed by Google, was used as a scalable numerical computing back-end, responsible for the optimized execution of models on Central Processing Unit (CPU) and Graphics Processing Unit (GPU), with GPU being used in this work. This framework stands out for its performance in machine learning tasks and its computational flexibility (Keeton, 2016).

Keras was used as a high-level Application Programming Interface (API) for defining convolutional neural network (CNN) architectures, allowing for the modular construction of layers and experimentation with different hyperparameters (TensorFlow Core, 2023). It was designed to accelerate the creation and execution of complex models, maintaining direct integration with TensorFlow as the execution agent.

3.3.2. Project workflow

The project development followed different approaches, consisting of different design and validation phases, until the final architecture was achieved. This section describes the main methodological steps, as well as the two distinct approaches (attempts) implemented to identify the phases of the cardiac cycle (S1, systole, S2, and diastole) and the subsequent detection of pathology. The process is represented by flowcharts that summarize the steps performed in each phase of the system.

A. Identification of the phase of the cardiac cycle

First approach

The initial phase of the project aimed to segment the cardiac cycle by detecting characteristic peaks in the PCG signal. In this sense, to ensure greater clarity and facilitate the analysis of patterns, only signals without pathology were used. To this end, signal pre-processing was implemented, followed by the application of Shannon energy envelope, allowing the temporal structure of the S1 and S2 sounds to be highlighted. Based on these detections, the aim was to delimit the systole and diastole phases, as can be seen in Figure 3.3.

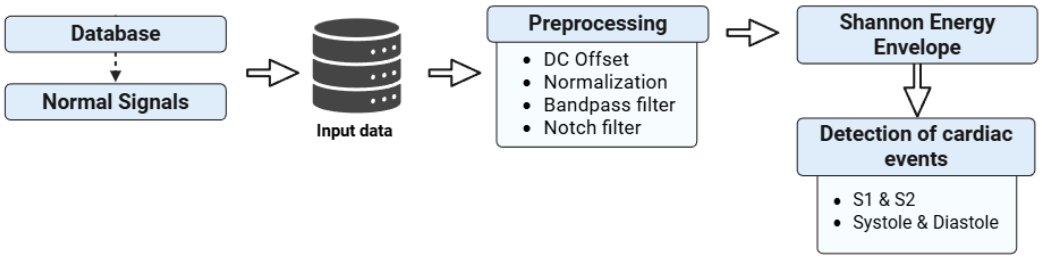

Figure 3.3 - Diagram of the first approach applied to cardiac cycle segmentation.

Second approach

The architecture of the system developed for the multiclass classification of the cardiac cycle components is shown in Figure 3.4. The process begins with the PhysioNet datasets, which are subjected to preprocessing and segmentation based on the temporal annotations of each cardiac event. Each interval corresponding to an event is cropped and adjusted to a 1-second window to ensure that all samples have the same temporal length.

The data are then divided into training, validation and test sets, following the *hold-out* methodology. In each of these subsets, four independent folds are created, containing the corresponding cardiac events. To increase the diversity of the samples, data augmentation is applied to the training set. The processed audio segments are then converted into spectrograms, which serve as input to different CNN models, responsible for the image-based classification of the cardiac cycle events.

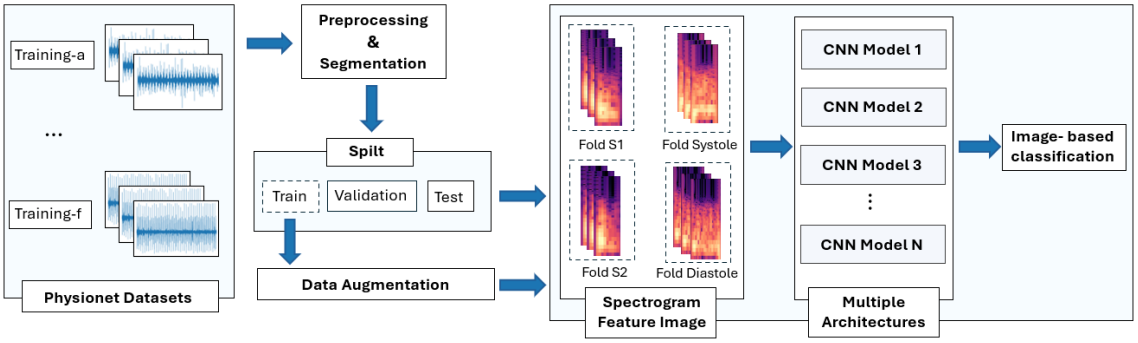

Figure 3.4 - Proposed system architecture for the multi-class classification of heart sound components (S1, systole, S2, and diastole).

B. Classification of pathologies (Normal vs. abnormal)

The overall architecture of the system implemented in this study is presented in Figure 3.5, which illustrates the workflow from data acquisition to the final classification stage. The process begins with the PhysioNet datasets, which are subjected to preprocessing and segmentation into 2-second windows. The data are then divided into training, validation and test sets, following the hold-out methodology. To balance the classes and improve the model's generalization capability, data augmentation is applied to the training set. The audio segments are converted into spectrograms, which serve as input for different CNN models, with two types of folds (normal and abnormal) for each of the training, validation and test subsets. These models perform image-based classification of heart sound recordings.

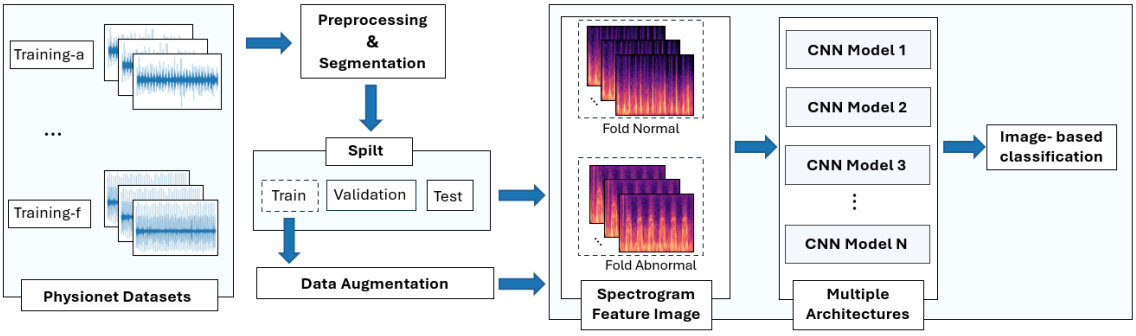

Figure 3.5 - Proposed system architecture for normal versus pathological heart sound classification.

3.4. Preprocessing

Signal noise is a phenomenon that affects both analog and digital systems, compromising the quality and integrity of the transmitted information. During the acquisition and recording of medical signals, such as phonocardiograms (PCG) and electrocardiograms (ECG), it is common for the signal to be contaminated by different types of noise. Among the factors included in this group are environmental noise, internal body sounds (such as respiratory and digestive sounds), and artifacts resulting from body movement, such as limb movement (Azam et al., 2022).

In this sense, preprocessing proved to be an essential step in preparing the data for the subsequent phases of the analysis. The operations performed were meticulously described and illustrated, ensuring the cleanliness, consistency, and suitability of the data for the study's objective. The main phases of preprocessing adopted in this investigation are presented below.

3.4.1. Characterization and Analysis of Filter behavior

In the initial stage of selecting filters for the preprocessing of cardiac sounds, the behavior of Butterworth and Bessel filters was characterized and analyzed. This evaluation involved applying a discrete unit impulse to the input of each filter, as shown in Equation (3.2), before applying them to the actual PhysioNet data. The aim was to observe and compare the impulse responses to gain a detailed insight into the dynamic behavior of each filter, particularly in terms of attenuation, frequency response and stability.

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{3.2}$$

According to the literature, the heart cycle frequency ranges from 25 Hz to 400 Hz, encompassing murmurs with higher frequency components (Schmidt et al., 2010; SINGH & MAJUMDER, 2020). To attenuate unwanted noise, Butterworth and Bessel bandpass filters of orders 2, 4 and 6 (type I) with cut-off frequencies between 25 Hz and 400 Hz were tested.

Figure 3.6 and Figure 3.7 illustrate the application of Butterworth and Bessel filters to a unit impulse, highlighting differences in their frequency responses.

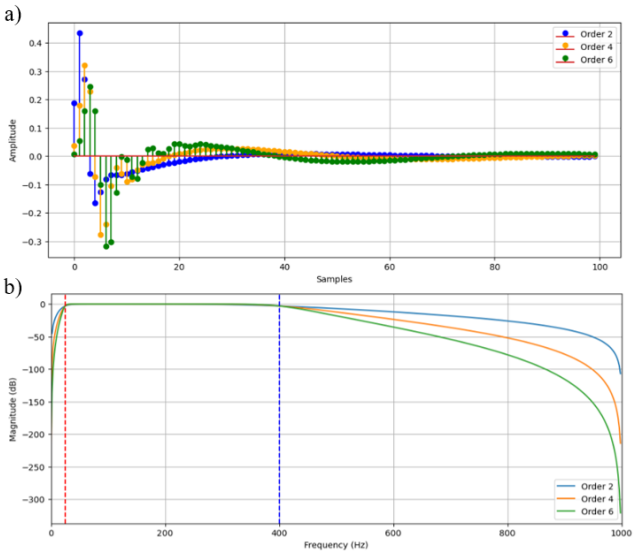

Figure 3.6 - Butterworth Filter Response – a) Impulse response; b) frequency response for different orders (2,4 and 6).

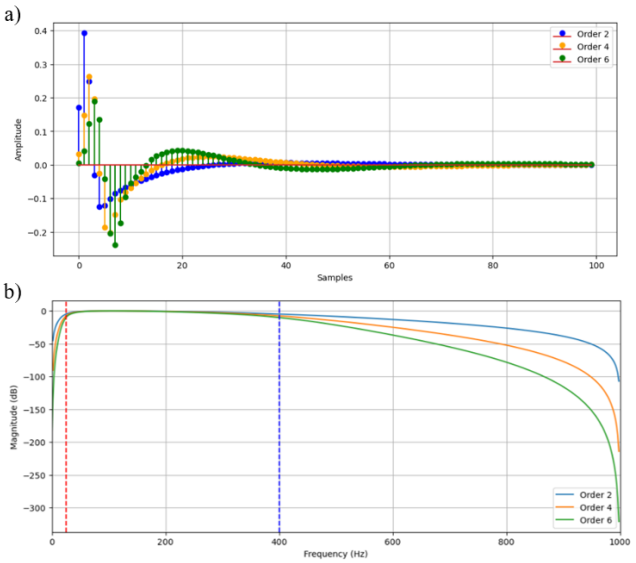

Figure 3.7 - Bessel Filter Response – a) Impulse response; b) frequency response for different orders (2,4 and 6).

A fourth-order Bessel filter was selected for preprocessing the phonocardiographic signal. This choice was based on the filter's characteristics, namely its nearly linear phase response and consistent group delay in the passband. As a result, the waveform is preserved with minimal temporal distortion, which is crucial in the analysis of cardiac sounds, where the accurate identification of pathologies depends on maintaining temporal integrity. Although the Butterworth filter provides stronger attenuation in the transition region, it introduces phenomena such as overshoot and ringing in the impulse response, which can distort short-duration signal components. In contrast, while the Bessel filter exhibits a more gradual attenuation outside the band of interest, it minimizes these undesirable effects and ensures a more faithful representation of the cardiac sounds (C.Baker, 2016).

Therefore, a fourth-order configuration was adopted as an appropriate compromise between attenuation performance and phase linearity, preserving the temporal characteristics of the phonocardiographic signal.

3.4.2. Direct Current Offset and Normalization

The first stage of preprocessing consisted of removing the DC component (also known as the continuous component or direct current offset) from the signal. This component corresponds to the average value of the signal over time, which can arise during sound capture, more specifically in the analog-to-digital (A/D) conversion process, introducing a baseline deviation that does not represent useful information (Alves de Brito, 2012). Removing this component allows the signal to be centered around zero, which facilitates the application of filters and signal analysis, avoiding distortions. This process was performed according to Equation (3.3):

$$x_n(t) = x(t) - \text{mean}(x(t)) \tag{3.3}$$

Where x_n (t) represents the signal without the continuous component, while x(t) corresponds to

the original signal.

The signal was then normalized relative to its maximum absolute value to restrict it to a common range between -1 and +1. This step is important, since the signals were acquired from different devices (Chakir et al., 2016), which can introduce scale variations. Normalization was performed using the equation (3.4):

$$x_n(i) = \frac{x(i)}{\max(|x(i)|)}, i = 1,2, \dots N \quad (3.4)$$

Where x_n (i) represents the corresponding normalized signal, while x (i) represents the original signal.

3.4.3. Bandpass Filter and Electrical Noise Removal

The resulting signal was filtered to eliminate noise and unwanted artifacts. A 4 th-order Bessel bandpass filter was applied, with cutoff frequencies set between 25 Hz and 400 Hz. This filter preserved the characteristic frequency components of heart sounds while attenuating low-frequency components, such as motion artifacts and respiratory noise, as well as high-frequency components related to ambient interference. Additionally, a notch filter with a cutoff frequency of 50 Hz was used to remove power line interference and other noise introduced by electronic equipment during signal acquisition.

Figure 3.8 presents two cardiac signals, the first from a healthy individual and the second from a patient with pathology. These figures illustrate the effect of the filtering process, comparing the raw signals with noise (in blue) to the filtered signals (in orange).

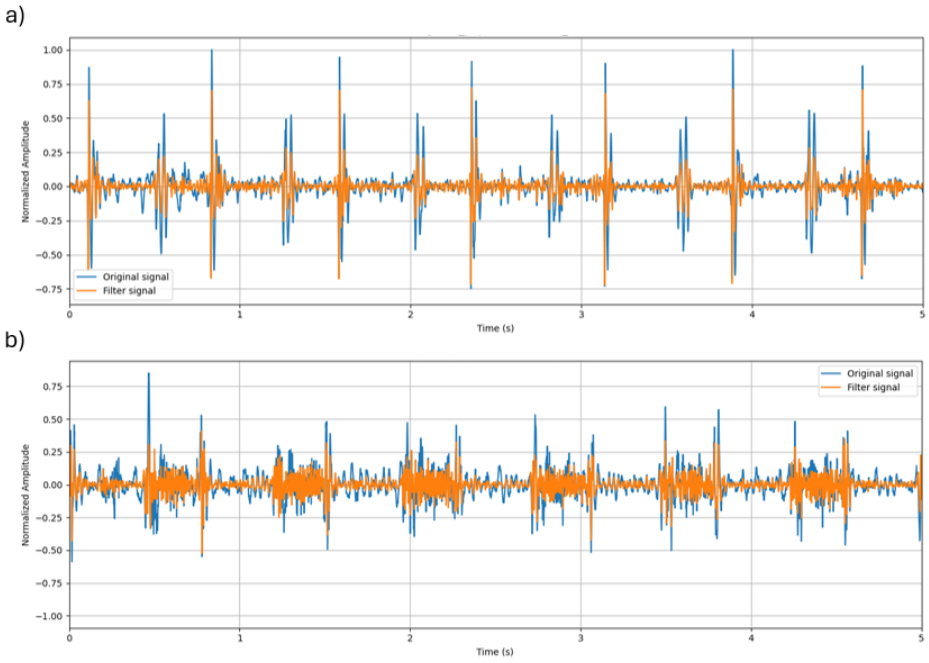

Figure 3.8 - Application of the filter to a) the normal signal a0080 and b) the pathological signal a0002.

3.4.4. Shannon Energy Envelope

To detect the main heart sounds (S1 and S2), Shannon energy was employed as the main technique. This approach enhances low-amplitude oscillations, facilitating the distinction between heart sounds and ambient noise (Meziani et al., 2012). Shannon energy was calculated according to Equation (3.5):

$$ES\ (i) = -{x_n}^2(i) \times \log({x_n}^2(i)) \tag{3.5}$$

where ES (i) represents the energy output of the filtered signal at time i and x_n refer to the normalized and filtered signal.

Following the computation of Shannon energy, its envelope was extracted by applying a Butterworth low-pass filter (LPF) with a cutoff frequency of 10 Hz (Jaros et al., 2023), as illustrated in Figure 3.9. This step aimed to smooth the energy curve, reduce abrupt variations, and improve the detection of local maxima associated with heart sounds. Finally, to further enhance the envelope, the resulting signal was normalized once again using Equation (3.4).

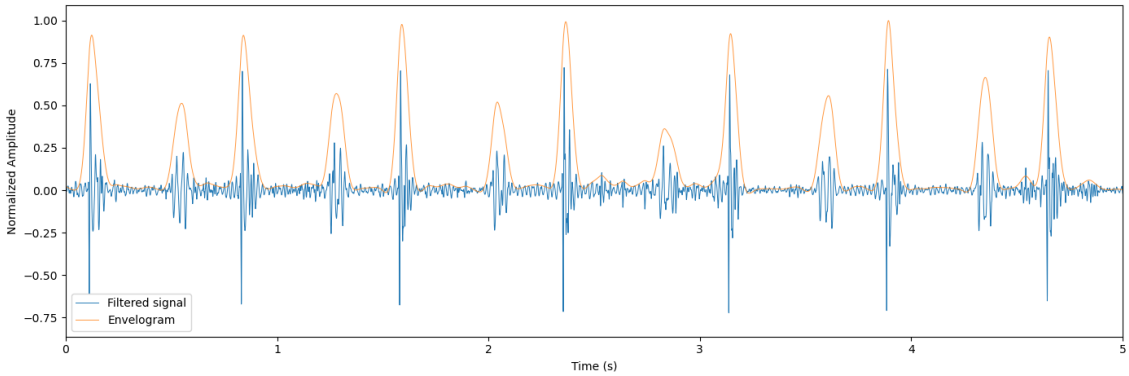

Figure 3.9 - Smoothed PCG (a0080) waveform envelope.

3.5. Segmentation

Due to the limitations identified in the initial approach, a second methodology was developed for the segmentation and classification of cardiac events, as well as for pathology classification, as described in Section 3.3.2.

In the second approach (Section 3.3.2 A), the different states of the cardiac cycle were segmented based on the temporal annotations indicating the onset of each event. Events with unavailable annotations (*"NA"*) were ignored. Since only the onset times were provided, the end of each event was defined as the onset of the subsequent one. Consequently, intervals of the form [start, end] were created, corresponding to each phase of the cardiac cycle. Each extracted segment was centered and adjusted to a fixed 1-second window through the application of *zero-*

padding [5]. The use of this uniform window allowed for the normalization of event durations and ensured that all samples shared the same input size for the network, given that each cardiac cycle component presents distinct temporal variations.

For pathology classification (Section 3.3.2 B), an alternative method was adopted in which the signals were divided directly into fixed-length segments, eliminating the need for prior detection of cardiac events. This strategy simplified the feature extraction process and increased the amount of data available for model training.

Considering that the average duration of a cardiac cycle ranges from 0.6 to 0.8 seconds (Zhou et al., 2023), a fixed window of 2 seconds was defined to ensure that each segment contained at least one to two complete cardiac cycles. Segments that did not meet this criterion were discarded, specifically those corresponding to signals with durations of 15 or 35 seconds, as illustrated in Figure 3.10.

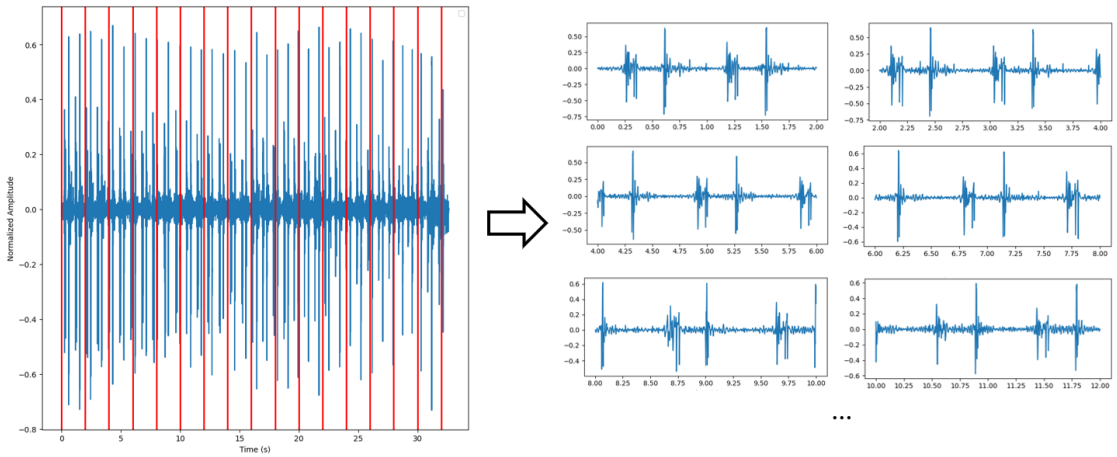

Figure 3.10 - Process of segmenting the normal signal “c0003.wav” into 2-second windows. Red lines mark the 2-second boundaries applied to the original PCG signal (left), producing the individual 2-second segments shown on the right.

3.6. Feature Extraction

For feature extraction, the datasets corresponding to the cardiac cycle events and pathology classification were previously segmented, as described in Section 3.5.

The analysis was performed using the Short Time Fourier Transform (STFT), which represents cardiac signals in the time–frequency domain. This technique provides information that complements the time-domain representation, allowing the observation of frequency components, signal sequences, and their temporal evolution.

Figure A.3 and Figure A.4 illustrates the spectrograms obtained for both classification

[5] Consists of adding zeros to a signal to extend its length to a fixed or predefined size. This operation doesn’t alter the original information but ensures uniform input dimensions and facilitates computational processing, such as convolution.

approaches, where the intensity and color indicate the distribution of energy in decibels (dB) over time and frequency (Hz). These representations highlight the distinct patterns and the modifications introduced by the data augmentation techniques (see Section 3.2).

The STFT was calculated identically for both cases, following the approach of (Tariq et al., 2022), according to Equation (3.6):

$$\mathrm{STFT}_x^f\ (t,f) = \int_{\infty}^{\infty} [x(t)\omega(t-\tau)e^{-j2\pi ft}\,dt] \tag{3.6}$$

where $x(t)$ represents the time-domain signal, τ denotes the time localization of the STFT, and $\omega(t-\tau)$ is the window function applied to segment the signal within the analysis interval and thereby shape its spectrum.

For the implementation of the spectrograms, the parameters proposed by (Salman Khan et al., 2021) were adopted: an FFT length (*n_fft*) of 128, a Hamming window of 64 ms, and a 75% overlap (*hop_length*). The Hamming window was employed to smooth transitions between consecutive frames and to attenuate spectral leakage effects caused by discontinuities at segment boundaries, thus providing a clearer time-frequency representation. This level of overlap was selected to increase the temporal resolution of the spectrogram, ensuring that short-duration cardiac events are consistently captured across successive frames and not lost between windows. Using smaller overlaps, such as 25%, would increase the temporal spacing between consecutive frames, reducing continuity in the time-frequency representation and leading to potential loss of relevant transient information.

Subsequently, the spectrogram values were linearly normalized to the range [0–255] to convert them into grayscale images and stored in *.png* format, with dimensions of 63 x 65 (height x width), for the identification of cardiac cycle phases and 65 x 126 for pathology classification. This conversion ensures compatibility and preserves image quality during compression, making them suitable for input into the deep learning model.

3.7. Classification Model

The proposed model is a two-dimensional convolutional neural network (2D-CNN), composed of an input convolutional layer followed by pooling layers and fully connected layers. Several CNN architectures were tested, varying their hyperparameters to evaluate performance in classifying normal versus pathological heart sounds.

In a preliminary exploratory phase, ten different CNN configurations were prototyped by changing the number of convolutional layers, the number of filters per layer and the pooling strategy. These models were trained and evaluated on the validation set, and their performance was compared using accuracy and validation loss. Based on these results, four representative architectures were selected for systematic study. They were chosen because they showed the best trade-off between classification performance and computational cost, and because they allowed analyzing the impact of network depth and filter allocation while keeping the remaining

hyperparameters fixed.

A general overview of the proposed CNN model architecture is illustrated in Figure 3.11, which depicts the sequential processing flow from input to classification. The different hyperparameter configurations considered in the experimental phase are summarized in Table 3.2 whereas the detailed description of each architecture is presented in Sections 3.7.1 - 3.7.4. The results of these experiments are presented in Chapter 4, while their discussion is provided in Chapter 5.

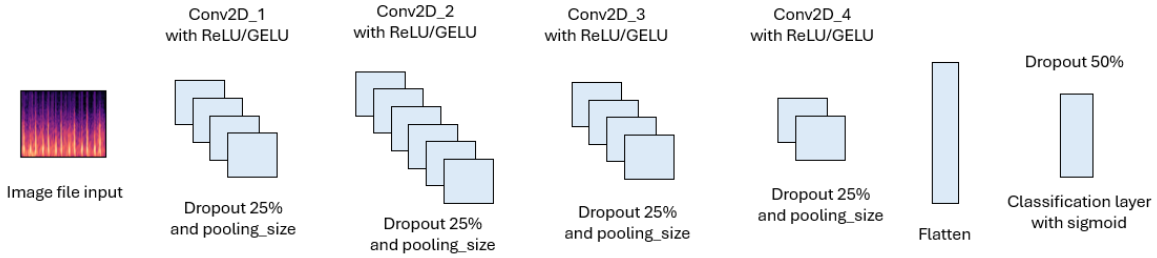

Figure 3.11 - Schematic representation of the proposed CNN architecture.

Table 3.2 - Experiments with different Architectural variants of CNN.

Experiment No.	Architectures of different investigated CNN models
1	Convolutional layers: 3
	$[(128 \times 2 \times 2) + maxpool\ (3 \times 3)], [(256 \times 2 \times 2) + maxpool\ (3 \times 3)], [(128 \times 2 \times 2) + maxpool\ (3 \times 3)]$
2	Convolutional layers: 3
	$[(128 \times 3 \times 3) + maxpool\ (3 \times 3)], [(256 \times 3 \times 3) + maxpool\ (3 \times 3)], [(128 \times 3 \times 3) + maxpool\ (3 \times 3)]$
3	Convolutional layers: 3
	$[(128 \times 3 \times 3) + maxpool\ (3 \times 3)], [(512 \times 3 \times 3) + maxpool\ (3 \times 3)], [(128 \times 3 \times 3) + maxpool\ (3 \times 3)]$
4	Convolutional layers: 4
	$[(128 \times 3 \times 3) + maxpool\ (2 \times 2)], [(256 \times 3 \times 3) + maxpool\ (2 \times 2)], [(128 \times 3 \times 3) + maxpool\ (2 \times 2)], [(64 \times 3 \times 3) + maxpool\ (2 \times 2)]$

The choice of small kernels (2×2 e 3×3), as presented in Table 3.2, is justified by their effectiveness in capturing local patterns within the STFT spectrograms, where short time-frequency variations characteristic of the cardiac cycle occur (Salman Khan et al., 2021; Tariq et al., 2022). Using larger kernels would considerably increase the processing time and the number of floating-point operations performed in each layer, making the training process slower, particularly on systems without dedicated computational units. In addition, larger kernels tend to introduce redundancy because they cover excessively wide regions of the spectrogram and often capture very similar information (Lau et al., 2024). At the same time, these kernels lose accuracy in identifying short and specific structures, as they aggregate overly broad areas and fail to focus on the local variations that are relevant to cardiac sound analysis (Lau et al., 2024). The use of smaller kernels therefore reduces the number of parameters, mitigates the risk of overfitting, and preserves the model's ability to detect meaningful details within heart sound recordings (Salman Khan et al., 2021; Tariq et al., 2022).

Three aspects are noteworthy for the functioning of the CNN model:

(1) Audio signals were converted into STFT spectrograms and subsequently transformed into images *“png”*, providing rich visual representations of temporal and spectral characteristics;
(2) Each CNN architecture was adjusted to better accommodate the characteristics of the spectrogram images, ensuring that the convolutional filters could extract relevant temporal a spectral feature;
(3) Training was fundamental to enable the models to correctly classify the proposed categories, and to identify the hyperparameter configuration that yielded the best classification performance.

3.7.1. Model 1

The proposed Model 1 consists of three convolutional layers configured according to the hyperparameters presented in Table 3.2. Between layers, the ReLU activation function was used, and the GELU function was also tested to compare their impact on the model’s performance. Max pooling was applied to reduce dimensionality, and dropout was introduced at specific points in the network to mitigate the risk of overfitting and improve generalization capability. Training was performed over 110 epochs, with a batch size of 32, using the Adam optimization algorithm with a learning rate of 0.001. In the binary classification setup, the model comprises 263 681 (0.26 million) trainable parameters. A summary of the defined hyperparameters is provided in Table 4.1.

The first convolutional layer is composed of 128 filters with a 2×2 kernel size, followed by a max pooling operation with a 3×3 stride, application of the ReLU (or GELU) activation function, and a dropout rate of 0.25. The second layer includes 256 filters, maintaining the same kernel size and activation and pooling structure. The third and final convolutional layer contains 128 filters, following the same configuration as the previous layers and enabling the gradual extraction of higher-level features.

After the convolutional phase, the extracted features are flattened, and a dropout rate of 0.5 is applied before the final dense layer. This dense layer performs the classification, using the Sigmoid activation function for binary classification (normal vs. pathological).

3.7.2. Model 2

Model 2 retains the same pipeline described for Model 1. It uses convolutional blocks followed by activation, max pooling and dropout, then flattening and a final dense layer. The architecture comprises three convolutional layers with 128, 256 and 128 filters and 3×3 kernels in all layers. Each convolution is followed by max pooling with 3×3 pool size and stride and a dropout rate of 0.25. After feature extraction the feature maps are flattened and a dropout rate of 0.5 is applied before the final dense layer, which performs the classification using Sigmoid for

binary problems.

The training protocol mirrors that of Model 1 with 110 epochs, a batch size of 32 and the Adam optimizer with a learning rate of 0.001. Owing to the 3×3 kernels the model yields a lower parameter count, with 591 873 (0.59 M) trainable parameters in the binary setup with one output unit. The full set of hyperparameters and the per-layer parameter counts are provided in Table A.2.

3.7.3. Model 3

Model 3 follows the pipeline defined in the previous models and adopts a deeper configuration in terms of filters. The architecture comprises three convolutional layers with 3×3 kernels and 128, 512, and 128 filters, respectively. Each convolution is followed by max-pooling with a 3×3 window and stride, and by dropout of 0.25, which reduces the spatial dimension and helps control overfitting. ReLU and GELU activations were tested between layers.

After feature extraction, the feature map is flattened and dropout of 0.5 is applied before the final dense layer. This layer performs the classification using Sigmoid for the binary task (normal versus pathological).

The training regimen mirrors that of the other models, comprising 110 epochs, a batch size of 32, and the Adam optimizer with a learning rate of 0.001. In the binary scenario, with one output unit and Sigmoid activation, the model totals 1 181 953 (1.18 M) trainable parameters. The full per-layer specification is provided in Table A.3.

3.7.4. Model 4

Model 4 follows the same pipeline as the previous models but adopts four convolutional layers. It uses 3×3 kernels with 128, 256, 128 and 64 filters, respectively each convolution is followed by 2×2 max-pooling and dropout of 0.25. ReLU and GELU activations were tested between layers. After feature extraction, feature map is flattened and dropout of 0.5 is applied before the final dense layer, which performs the classification for the binary problems. The training regimen mirrors that of the earlier models. This configuration yields 666 049 (0.66 M) trainable parameters in Sigmoid activation. Full per-layer details are provided in Table A.4.

.

Chapter 4

Results and Evaluation

Several experiments were conducted for the developed models using cardiac sound datasets. Each model was evaluated according to the one classification approach described in Section 3.3.2. The following sections present the results obtained for the four models, considering the augmented dataset for binary classification tasks. The results include the main evaluation metrics such as accuracy, loss, precision, recall, specificity, F1-score and the corresponding confusion matrix. The experimental results were analyzed in parallel with recent methods to assess the relative effectiveness of the proposed architecture.

4.1. Experimental Procedure/Setup

For the first approach, which corresponds to binary classification, a workstation equipped with an Intel Core i7 octa-core CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 3070 GPU was used. For the second approach, related to multiclass classification (S1, S2, systole, and diastole), a different computational configuration was planned to be used in future work, namely the Vision Supercomputer, composed of two compute nodes and one management node. Each compute node consists of an NVIDIA DGX A100 system equipped with dual AMD Rome 7742 processors (128 cores in total), 1 TB of system memory, and 8 NVIDIA A100 Tensor Core GPUs with 40 GB of GPU memory each, providing a total of 320 GB of GPU memory per node. The two DGX A100 systems are interconnected through 8 × 200 Gb/s HDR InfiniBand links, ensuring high-bandwidth and low-latency communication between nodes. The second approach was not experimentally evaluated due to time and resource constraints.

The dataset was divided into 80% for training, 10% for validation and 10% for testing. This proportion was chosen to maximize the amount of data available for training, which is essential given the limited size and class imbalance of the PhysioNet dataset, while also ensuring independent subsets for hyperparameter tuning (validation) and for an impartial final evaluation of the model (test). This split follows widely adopted practices in deep learning studies that use datasets of similar scale.

During the training process, the model performance (accuracy and loss) was continuously monitored on independent validation data that had never been previously observed. This procedure allowed the evaluation of the model's generalization ability, the detection of possible signs of overfitting, and the adjustment of hyperparameters to improve performance before proceeding to the next stage.

Data augmentation was also applied during training to balance the datasets, enhance the learning capacity of the models and mitigate overfitting. In the binary classification, augmentation was applied only to the minority class (abnormal), whereas in the multiclass classification,

corresponding to S1, systole, S2 and diastole, it was applied to the entire training set.

The test set, consisting of data unseen by the model, was used only at a later stage for the final evaluation. Several performance metrics were reported for both the binary and multiclass approaches, considering the results obtained from the training and test sets. The metrics for each class were computed for the four proposed models using the augmentation cardiac datasets.

Finally, the activation functions used in the convolution layers (ReLU and GELU) were analyzed and compared, as well as the output activation function, which was selected according to the binary classification approach, using a sigmoid activation.

4.2. Dataset

In this study, two datasets were used, both of which underwent preprocessing and segmentation, followed by the application of data augmentation techniques to increase the number of training samples and improve class balance. The following subsections describe the characteristics of each dataset and the increase in data volume resulting from this process.

4.2.1. Cardiac sound dataset for normal and abnormal classes

The cardiac sound dataset used in this study originates from the PhysioNet repository and contains a total of 3 240 audio recordings. These recordings are divided into two categories, with 2 575 corresponding to normal heart sounds and 665 to abnormal heart sounds.

The raw signals underwent a preprocessing stage, and the recordings were subsequently segmented into two-second clips, a procedure considered relevant for model training. To increase the diversity and representativeness of the data, data augmentation techniques were applied to the preprocessed training set. This procedure allowed the creation of additional samples from the minority class, thereby balancing the dataset and reducing the risk of overfitting.

After applying data augmentation, the total number of training samples increased from 28 187 to 41 303. Table 4.1 presents the detailed distribution of the samples across the training, validation and test sets, before and after the application of data augmentation. The augmentation process was applied exclusively to the training set, while the validation and test sets remained unchanged.

Table 4.1 - Distribution of the cardiac sound dataset samples by class and subset before and after data augmentation.

Class	Training (Original Data)	Training (Augmented Data)	Validation	Test	No. of Spectrogram Images/class
Normal	21 629	21 629	2 519	2 648	26 796
Abnormal	6 558	19 674	750	878	21 302
Total	28 187	41 303	3 269	3 526	48 098

4.2.2. Cardiac cycle dataset

This section describes the annotations of the cardiac cycle, corresponding to each individual cardiac event. The annotations were obtained using the algorithm proposed by (Springer et al., 2015) and were later manually reviewed by cardiologists, resulting in a total of 84 467 annotated cardiac cycles.

From the complete set of annotations, unidentified elements were automatically excluded, that is, those without temporal position information in the signal.

The signals underwent a preprocessing stage and were subsequently segmented according to the specific events of the cardiac cycle (S1, systole, S2 and diastole), considering their respective temporal intervals. This procedure enabled the creation of distinct folds for each event, ensuring a structured and consistent division of the data.

To improve the model accuracy and increase the diversity of the training set, data augmentation techniques were applied to the preprocessed signals. The augmentation process was performed on all cardiac events of the training set, while the validation and test sets remained unchanged.

After applying data augmentation, the total number of training samples increased from 272 383 to 817 149. Table 4.2 presents the detailed distribution of the samples across the training, validation and test sets, before and after the application of the augmentation process.

Table 4.2 - Distribution of the segmented cardiac cycle samples (S1, systole, S2 and diastole) across the training, validation and test sets before and after data augmentation.

Class	Training (Original Data)	Training (Augmented Data)	Validation	Test	No. of Spectrogram Images/class
S1	63 877	203 631	8 359	8 190	220 180
Systole	68 059	204 177	8 382	8 222	220 781
S2	67 766	203 298	8 346	8 169	219 813
Diastole	68 681	206 043	8 457	8 276	222 776
Total	272 383	817 149	33 544	32 857	883 550

4.3. Classification Results and Evaluation

This section presents the results obtained from the experiments conducted using the four CNN architectures described in Chapter 3. Each model was evaluated under two activation functions, ReLU and GELU, and using one classification binary (normal vs. abnormal) strategy.

The evaluation includes the performance metrics obtained for each model, as well as the corresponding confusion matrices.

The results obtained for models in the binary classification task (normal vs. abnormal) are presented in this section. Two activation functions, ReLU and GELU, were tested in the convolutional layers, while the sigmoid activation was used in the output layer.

Model 1

The detailed numerical metrics for training, validation and testing phases are provided in Table A.5, where the performance of both activation functions can be compared in detail.

Figure 4.1 illustrates the evolution of accuracy and loss across epochs for both ReLU and GELU activations.

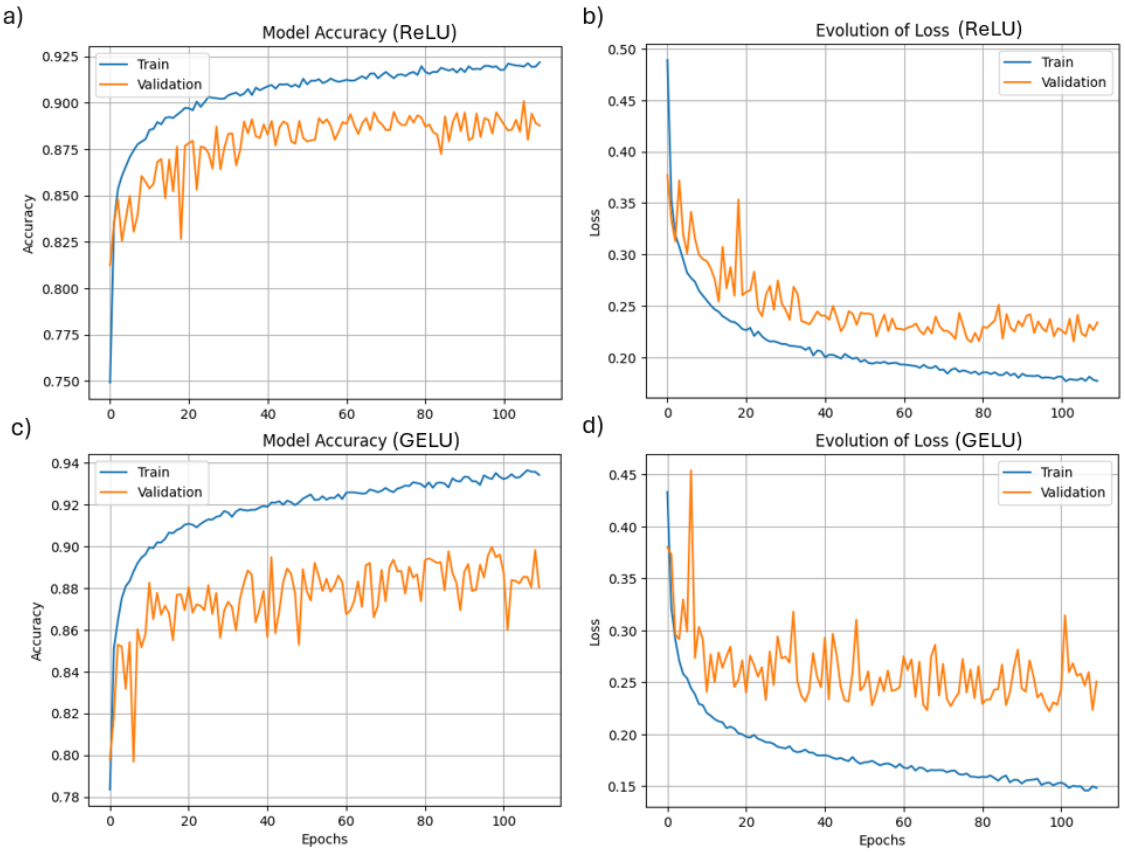

Figure 4.1 - Evolution of accuracy and loss for model 1 with ReLU and GELU activations in the convolutional layers. a) ReLU – accuracy; b) ReLU – Loss; c) GELU – accuracy; and d) GELU – loss.

Figure 4.2 presents the confusion matrices computed for each activation function, enabling a direct comparison of their performance in distinguishing normal versus abnormal samples.

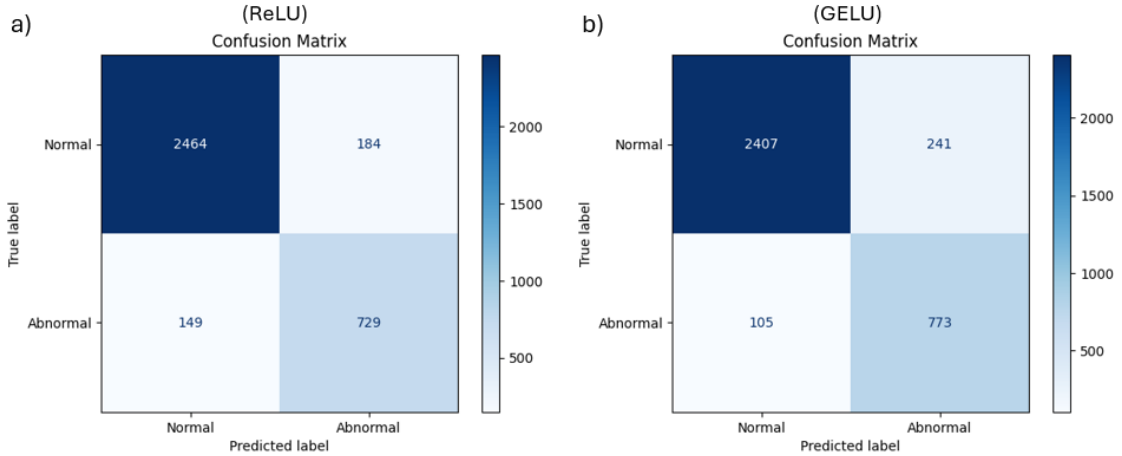

Figure 4.2 - Confusion matrix for Model 1 with a) ReLU and b) GELU activations.

Model 2

The detailed numerical results for model 2, including training, validation and testing metrics, are provided in Table A.6, where the performance of both ReLU and GELU activations can be compared in detailed.

Figure 4.3 illustrates the evolution of accuracy and loss throughout the training epochs for both activation functions.

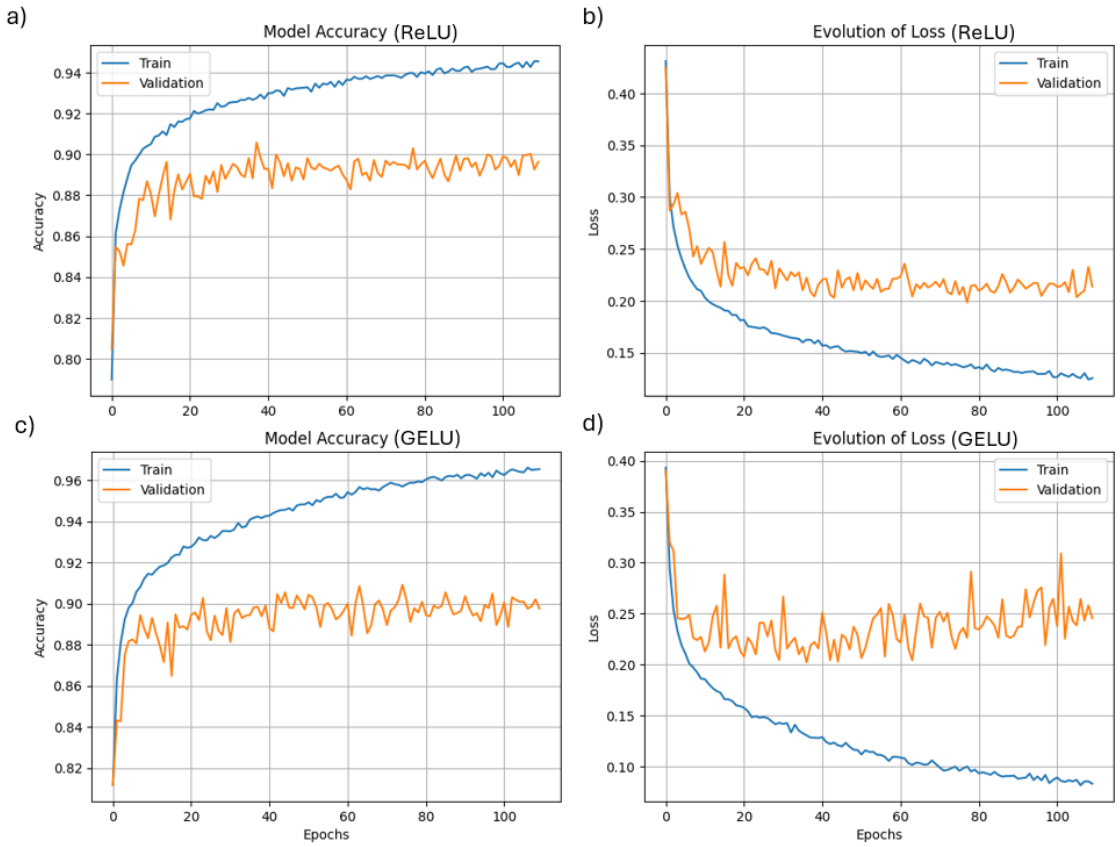

Figure 4.3 - Evolution of accuracy and loss for model 2 with ReLU and GELU activations in the convolutional layers. a) ReLU – accuracy; b) ReLU – Loss; c) GELU – accuracy; and d) GELU – loss.

Figure 4.4 present the confusion matrices obtained for the ReLU and GELU activations, highlighting the correct and incorrect classifications between normal and abnormal heart sound classes. Both activations produced similar classification distributions, with GELU showing a slightly higher number of false negatives compared to ReLU.

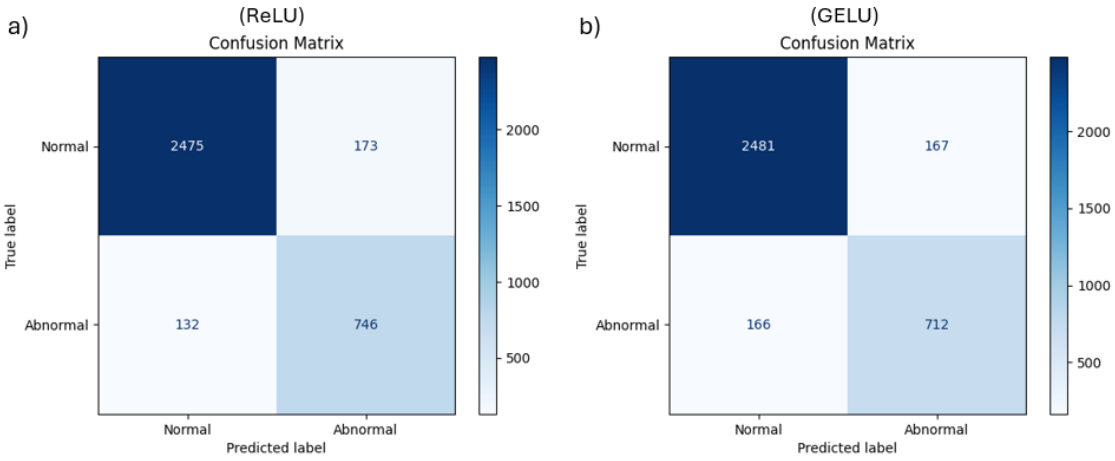

Figure 4.4 - Confusion matrix for Model 2 with a) ReLU and b) GELU activations.

Model 3

The detailed numerical metrics for training, validation and testing phases are provided in Table A.7, where the performance of both activation functions can be compared in detail.

Figure 4.5 illustrates the evolution of accuracy and loss across epochs for both ReLU and GELU activations.

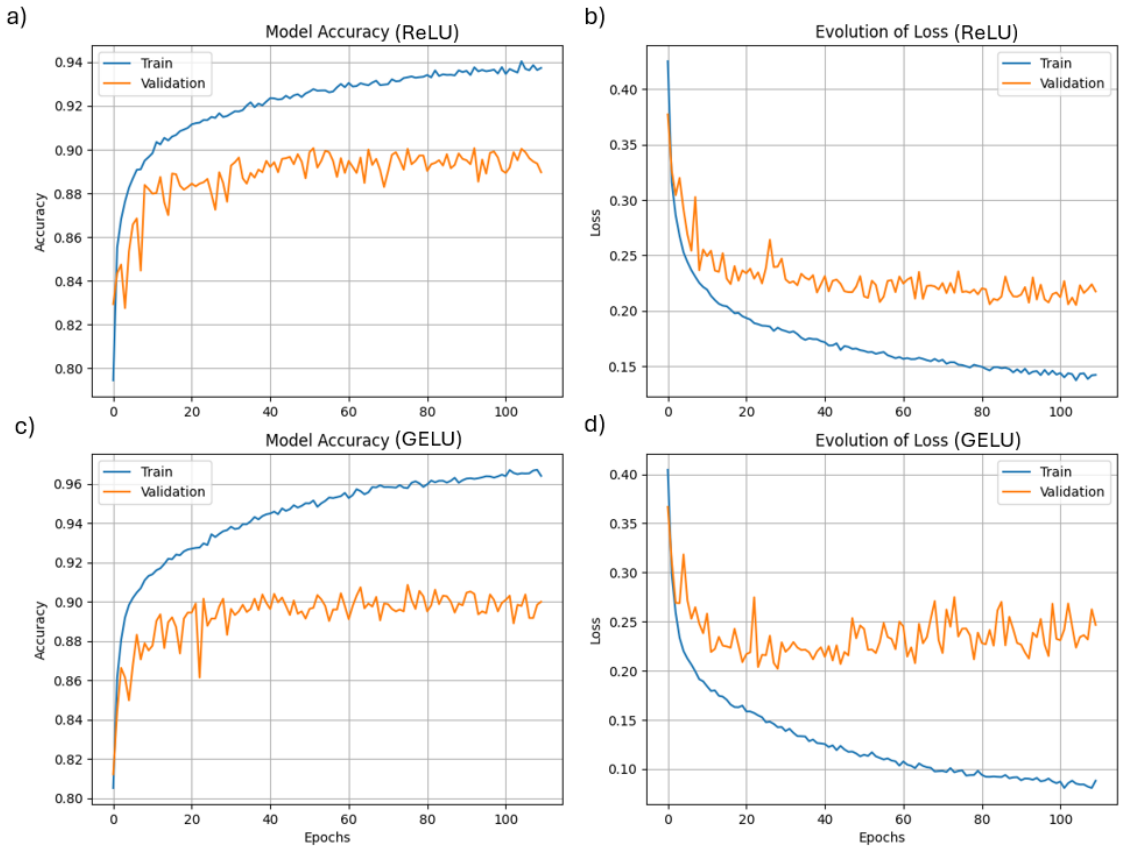

Figure 4.5 - Evolution of accuracy and loss for model 3 with ReLU and GELU activations in the convolutional layers. a) ReLU – accuracy; b) ReLU – Loss; c) GELU – accuracy; and d) GELU – loss.

Figure 4.6 presents the confusion matrices obtained for both activation functions, highlighting the separation between normal and abnormal classes.

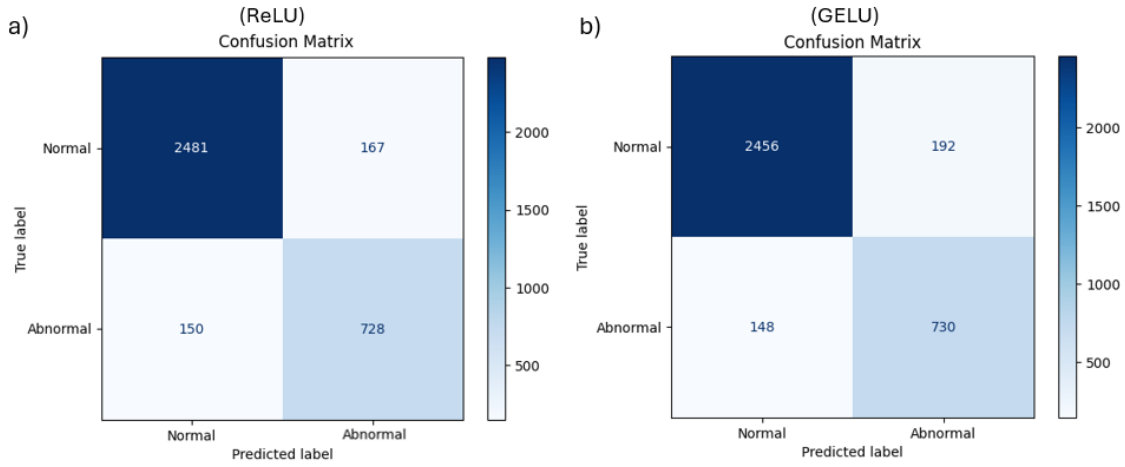

Figure 4.6 - Confusion matrix for Model 3 with a) ReLU and b) GELU activations.

Model 4

The detailed numerical metrics for training, validation and testing phases are provided in Table A.8, where the performance of both activation functions can be compared in detail.

Figure 4.7 illustrates the evolution of accuracy and loss across epochs for both ReLU and GELU activations.

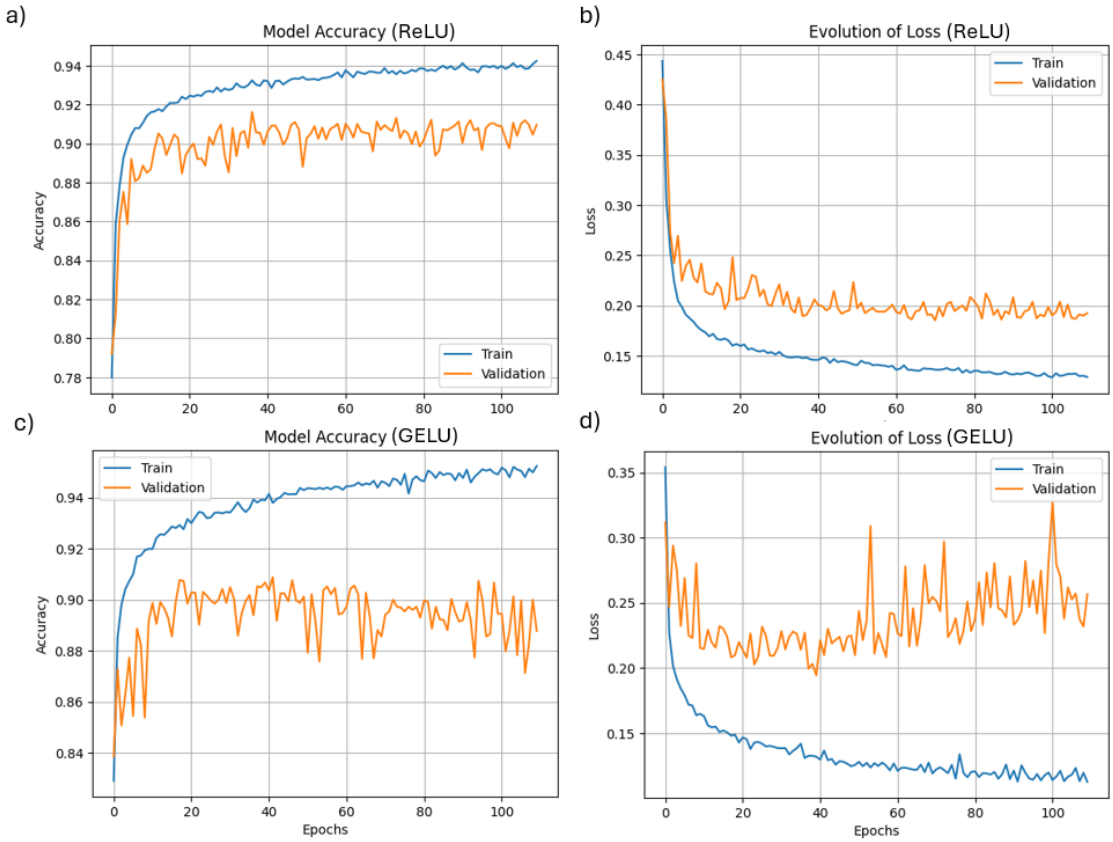

Figure 4.7 - Evolution of accuracy and loss for model 4 with ReLU and GELU activations in the convolutional layers. a) ReLU – accuracy; b) ReLU – Loss; c) GELU – accuracy; and d) GELU – loss.

Figure 4.8 illustrates the confusion matrices generated for the two activation functions, revealing the classification performance across normal and abnormal classes.

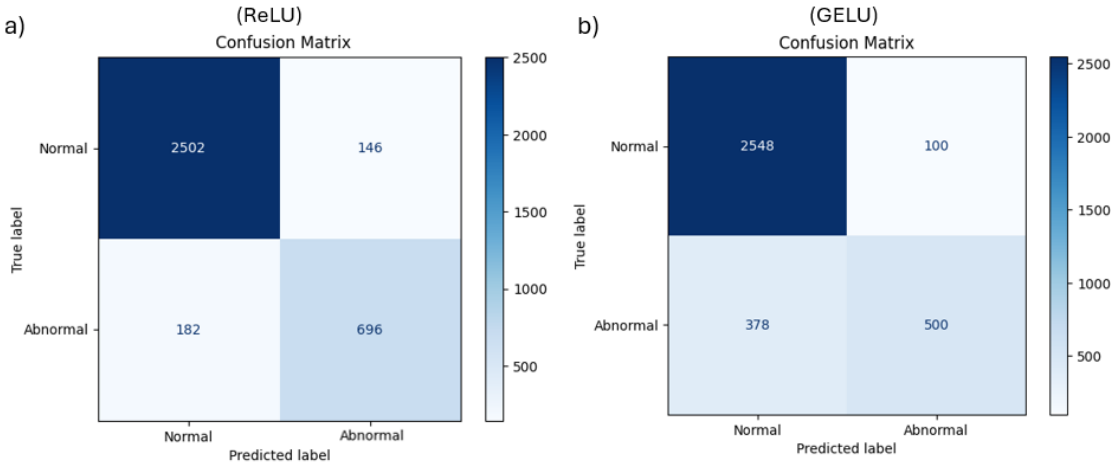

Figure 4.8 - Confusion matrix for Model 4 with a) ReLU and b) GELU activations.

Chapter 5

Discussion Results

This chapter presents the discussion of the results obtained from the different models developed throughout this work. The objective is to analyze, interpret and understand the performance of the proposed architecture in relation to the goals defined at the beginning of the study. The analysis focuses on the main performance metrics, the influence of the methodological approaches adopted, and the identification of potential signs of overfitting.

In addition, a comparative analysis with related studies from the literature is presented, allowing the positioning of the proposed models within the context of the current state of the art. The discussion highlights the advantages and limitations of each model and outlines potential directions for future improvements, providing a comprehensive understanding of the models' effectiveness in heart sound classification.

5.1. Analysis of results

This section presents the analysis of the results obtained for the different implemented architectures of the binary classification, as well as their behavior throughout the learning process.

Model 1, using the ReLU activation function, achieved satisfactory results, reaching an *accuracy* of 94.53% on the training set and 90.56 % on the test set. The difference of approximately 3.97% between both sets indicates a good generalization capability, although with slight signs of overfitting, as reflected by the validation performance (88.77%). The *accuracy* and *loss* curves (Figure 4.1 (a-b)) confirm this trend, showing a stable evolution: the training curve (blue) continues to improve progressively, while the validation curve (orange) tends to stabilize with small oscillations. This behavior reveals overall stability but also some discrepancy between training and validation. Regarding the complementary metrics on the test set (Table A.5), the model achieved a *recall* of 83.03 %, demonstrating good sensitivity in detecting abnormal heart sounds, and a *precision* of 79.85 %, indicating the presence of some false positives (FP). The *specificity* (93.05%) and *F1-score* (81.41%) reinforce the model's overall consistency. The confusion matrix (Figure 4.2 (a)) reflects this behavior, showing 184 normal samples (FP) incorrectly classified as abnormal, compared with 149 false negatives (FN). When the activation function was replaced with GELU, a slight increase in training *accuracy* (96.57%) and *recall* was observed, but no significant improvement was achieved on the test set (90.19%). The model became more sensitive to the abnormal class achieving a *recall* of 88.04 %, at the expense of a reduction in precision (76.23 %), indicating a higher number of false positives, as illustrated in Table A.5. The learning curves (Figure 4.1 (c-d)) also exhibit stronger oscillations in the validation set and greater divergence from the training curve, suggesting a higher tendency toward overfitting. In addition, the *loss* curve tends to stagnate at higher values compared to the ReLU-

based model, further reinforcing this tendency.

Model 2, using the ReLU activation function, presented consistent results, demonstrating good generalization capability. It achieved an *accuracy* of 97.63 % on the training set and 91.35 % on the test set, with a difference of approximately 6.3%. The *accuracy* on the validation set (89.63%) confirms this trend, showing stable performance but slightly lower than that of the training set. The *accuracy* and *loss* curves (Figure 4.3 (a-b)) show a regular evolution, similar to that observed in Model 1. In the test set, the metrics (Table A.6) achieved a *recall* of 84.97 %, *precision* of 81.18 %, *specificity* of 93.47 %, and an *F1-score* of 83.03 %, indicating a solid balance between sensitivity and precision. The confusion matrix (Figure 4.4 (a)) shows 173 FP and 132 FN, with 2,475 TN and 746 TP, consistent with the reported metrics. When the GELU activation was applied, the model reached an *accuracy* of 99.53 % on the training set and 90.56 % on the test set, showing a more pronounced difference of 9 %. The validation set (89.75 %) exhibited more noticeable fluctuations, and as the model continued to train, the *loss* values increased over the epochs. This indicates that the model began to fit excessively to the training data, ceasing to improve on the validation set. In the test set, the *recall* (81.09 %) and *precision* (81.00 %) remained close, resulting in an *F1-score* of 81.05 %, which is slightly lower than that obtained with ReLU. The confusion matrix shows 167 FP and 166 FN.

With ReLU, model 3 maintained a balanced performance between the training and testing phases, achieving accuracy values of 97.13% and 91.01%, respectively. The difference between both indicates an adequate adjustment without significant signs of overfitting, a behavior similar to that already observed in previous models. The precision (81.34%) and recall (82.92%) metrics demonstrate a good ability to identify abnormal sounds without compromising precision. The confusion matrix, presented in Figure 4.6 (a), confirms this behavior by showing reduced proportions of false positives and false negatives. When the GELU activation function was applied, the model reached slightly higher training values, as shown in Table A.7, but this was accompanied by a decrease in validation accuracy to 80% and a slight drop in the test set compared with the model using ReLU. This behavior is consistent with that observed in other models employing the same activation function. As depicted in Figure 4.6 (c-d), the accuracy and loss curves of the validation phase (orange line) reveal that after a certain number of epochs, the model starts fitting excessively to the training data, reducing its ability to generalize.

Finally, the last model using the ReLU activation maintained a stable and balanced performance across the different data partitions, achieving an accuracy of 96.31% in training and 90.70% in testing, present in Table A.8. The small difference between both confirms a good model adjustment without signs of overfitting. The validation accuracy reached 90.95%, and the loss curve remained almost stagnant after the first epochs, as illustrated in Figure 4.7 (a-b). The confusion matrix shown in Figure 4.8 (a) indicates that the number of false negatives (182) out of a total of 878 abnormal signals was slightly higher compared with the previous models (1, 2, and 3). When the activation was changed to GELU, the model achieved a slightly higher training accuracy (96.96%), but lower results in validation (88.77%) and testing (86.44%). The reduction in generalization performance, together with the greater gap between the training and validation

curves observed in Figure 4.7 (c–d), suggests that after a certain number of epochs the model starts to fit excessively to the training data, losing generalization ability and memorizing patterns. This activation also caused a marked decrease in recall (56.95%), despite a slight improvement in precision (83.33%). The confusion matrix in Figure 4.8 (b) confirms this behavior, showing a higher number of false negatives (378) compared with the model using ReLU.

Considering these results, the model with GELU activation exhibited the lowest overall performance among all models evaluated (1, 2, 3, and 4), showing a significant loss of sensitivity in the detection of abnormal signals and a stronger tendency toward overfitting.

5.2. Interpretation of results

The results obtained from the binary classification between normal and abnormal heart sounds demonstrated that the convolutional neural network (CNN) architecture was able to effectively learn discriminative patterns from the spectrograms generated by the Short-Time Fourier Transform (STFT). These findings confirm that time and frequency representations enable the model to capture both spectral and temporal variations within the signal, which are key characteristics of cardiac cycles, thereby improving classification performance when compared to approaches that rely solely on numerical or statistical signal features (Nogueira et al., 2019; Zhang & Han, 2017).

The data preprocessing stage also had a direct impact on model performance, as the absence of this step would lead the network to learn noise components and irrelevant patterns (Liu et al., 2016). The application of balancing and data augmentation techniques proved essential to mitigate the class imbalance present in the dataset, resulting in a more stable and reproducible training process, particularly for models using the ReLU activation function.

Regarding the network architecture, it was observed that the progressive increase in the number of convolutional layers and filters contributed to a more detailed extraction of the features present in the spectrograms. However, from Model 3 onward, the improvements became less significant, suggesting that excessively deep architectures with a higher number of parameters do not necessarily translate into better generalization capabilities, especially considering the limited size of the dataset. Consequently, Model 2, employing the ReLU activation, stood out among the others by presenting the best balance between performance, stability, and parameter count. Table 5.1 summarizes the metrics obtained for each model and the total number of parameters used during training.

Table 5.1 - Summary of test performance metrics obtained for CNN architectures using ReLU activation in convolutional layers and a sigmoid output function.

Metrics	Model 1	Model 2	Model 3	Model 4
Accuracy	90.56%	91.35%	91.01%	90.70%
Precision	79.85%	81.18%	81.34%	82.66%
Recall	83.03%	84.97%	82.92%	79.27%
Specificity	93.05%	93.47%	93.69%	94.49%
F1-score	81.41%	83.03%	82.12%	80.93%
Total Parameters	263 681	591 873	1 181 953	666 049

Experiments with different activation functions showed that ReLU achieved more stable and consistent results compared to GELU. Although GELU reached higher training accuracies, it exhibited poorer performance during validation and testing. This behavior confirms that ReLU remains a more robust option for small datasets, while GELU tends to favor model overfitting to the training data.

Overall, the results confirm the initial hypotheses of this study. The combination of CNNs with spectrogram-based representations enhanced the model's ability to discriminate between normal and abnormal heart sounds, demonstrating the effectiveness of this approach in cardiac signal analysis. Furthermore, increasing the number of layers or filters improved performance only up to a certain point, after which the gains became marginal. Finally, the findings reinforce the suitability of the ReLU activation function for this type of problem, providing a more stable and generalizable learning process than GELU, particularly in data-limited scenarios.

5.3. Comparison with State-of-the-Art

This subsection presents a brief review of relevant state-of-the-art approaches, followed by a discussion comparing their results with those obtained in this study. The main limitations identified during the work are also summarized.

5.3.1. State-of-the-Art

To validate the proposed model, a comparative analysis was conducted with several recent and relevant studies published between 2016 and 2022, as summarized in Table 5.2. These works focus on the automatic classification of heart sounds, considering aspects such as data augmentation, feature selection, network architecture, number of trainable parameters, and achieved accuracy.

In Potes et al. (2016), a hybrid model combining *feature engineering* and *deep learning* was proposed, being one of the first approaches to integrate traditional machine learning with CNNs for heart sound classification. The authors extracted 124 time-frequency features from the PhysioNet/CinC Challenge 2016 dataset and tested three approaches: AdaBoost alone, CNN alone, and an ensemble of both. The CNN architecture consisted of two convolutional layers with

8 and 4 filters (kernel size 5), followed by ReLU activation, 2×2 max pooling, and 0.25 dropout. The outputs were fed into a multilayer perception (MLP) with 20 hidden neurons, 0.5 dropout, and L2 regularization. The model was trained using the Adam optimizer with a batch size of 1024, a learning rate of 0.0007, and 200 epochs. The ensemble between AdaBoost and CNN achieved the best performance on the PhysioNet/CinC Challenge 2016 dataset, with sensitivity of 94.24%, specificity of 77.81%, and an overall accuracy of 86.02% in binary classification between normal and abnormal heart sounds.

A deeper CNN architecture was later developed by Rubin et al. (2017), applied to the same task and dataset. In this approach, the audio signals were converted into 3-second time–frequency segments using Mel-Frequency Cepstral Coefficients (MFCC), which were then used as input to the network. The CNN consisted of two convolutional layers with 64 filters (2×20 and 2×10), followed by max pooling operations and two fully connected layers with 1024 and 512 hidden units, and an output layer with sigmoid activation. L2 regularization and 0.86 dropout were applied. The model was trained using the Adam optimizer with a learning rate of 0.0001 and a batch size of 256. Despite the class imbalance, the network achieved a recall of 72.78%, specifically of 95.21%, and an overall accuracy of 83.99%.

In contrast, Li et al. (2020) proposed a hybrid approach that combined feature engineering with a compact 1D-CNN architecture. The signals were first segmented using a Hidden semi-Markov Model (HSMM) to identify the cardiac states S1, systole, S2 and diastole. From each recording, 497 handcrafted features were extracted across eight domains (time, amplitude, energy, higher-order statistics, cepstrum, frequency, cyclostationarity and entropy), capturing detailed morphological and spectral characteristics of the PCG. Two CNN configurations were evaluated. The main model consisted of three convolutional blocks with 32, 64 and 128 filters, each using kernel size 3, followed by 2×2 max pooling and 25% dropout. This version ended in a fully connected layer, achieving 88.6% accuracy, 76.4% recall, 91.9% specificity and 84.1% Macc. A second configuration replaced the dense layers with a Global Average Pooling (GAP) layer, simplifying the architecture and improving generalization, with 86.7% accuracy and 87% recall. The model was trained using binary cross-entropy, the Adam optimizer with a learning rate of 0.001, and five-fold stratified cross-validation. To address class imbalance, class weighting was applied (0.63 for normal and 2.37 for abnormal samples). The inclusion of the GAP layer reduced overfitting and provided a more stable performance compared with traditional fully connected layers.

Salman Khan et al. (2021) proposed a custom, lightweight CNN architecture for the automatic classification of unsegmented phonocardiograms, using the PhysioNet/CinC 2016 and PASCAL 2011 datasets. The signals were converted into STFT-based spectrograms and segmented into 8-second windows. After testing seven architectural variants, the final CNN architecture consisted of four convolutional layers with 128, 256, 128, and 64 filters (3×3), followed by (2×2) max pooling and 0.25 dropout, culminating in a fully connected layer with 50% dropout and a sigmoid output activation for binary classification (normal vs. abnormal). The model was trained using binary cross-entropy loss, the Adam optimizer with a learning rate of 0.001, and

ten-fold cross-validation. On the PhysioNet dataset, it achieved 95.4% accuracy, 96.3% recall, 92.4% specificity, 97.6% precision, and an F1-score of 96.98%. When trained with the combined PhysioNet-PASCAL dataset, the accuracy remained high (94.2%). Using transfer learning from the pre-trained PhysioNet model, the approach achieved a precision of 98.29% on the PASCAL dataset.

Later, Tariq et al. (2022) introduced an innovative model named Feature-Based Fusion Disease Classification (FDC-FS), developed for the automatic classification of heart and lung sounds into six classes. This study stands out for combining *transfer learning* and *feature-level fusion*, integrating three independent convolutional neural networks, FDC1, FDC2, and FDC3, each trained on a distinct audio representation: STFT spectrograms, MFCC, and chromagrams. To address data scarcity and class imbalance, several data augmentation techniques were applied, including white noise addition, pitch shifting, and time stretching, enhancing variability and model robustness. Each CNN was trained using the Adam optimizer with an L2 regularization factor of 0.0005, categorical cross-entropy loss, and ReLU activation between layers. The batch size was set to 64 for the original dataset and 128 for augmented data. Training was carried out for 50 epochs on the original dataset and 30 additional epochs after augmentation. To prevent overfitting, a dropout layer (rate 0.5) was applied before the final layer. The three CNNs were subsequently combined through a fusion layer that integrated the feature vectors from each model. This fusion process, supported by transfer learning, enabled the final model to learn complementary relationships between different audio representations. The resulting architecture comprised 13 convolutional layers and 3 dense layers, with a total of 1.48 million trainable parameters. On the heart sound dataset, the FDC-FS model achieved 97% accuracy, outperforming most previous approaches and demonstrating the effectiveness of multimodal feature fusion in biomedical sound classification.

Table 5.2 - State-of-the-art Heart Classification Models.

Author [Year]	Classes	Balancing	Feature Extraction	Classification Algorithm	Performance Metrics
Potes et al. (2016)	2	No	124 time-Frequency features (MFCC)	CNN	Accuracy: 86.02% Recall: 94.24% Specificity: 77.81%
Rubin et al. (2017)	2	No	3-s overlapping segments, MFCC spectrograms	CNN	Accuracy: 83.99% Recall: 72.78% Specificity: 95.21%
Li et al. (2020)	2	Yes (Class weighting)	497 features (8 domains)	1D-CNN with Fully Connected layer	Accuracy: 88.6% Recall: 76.4% Specificity: 91.9% Macc[6]: 84.1%
				1D-CNN with Global Average Pooling	Accuracy: 86.7% Recall: 87% Specificity: 86.6% Macc: 86.8%
Salman Khan et al., (2021)	2	No	8-s segments, STFT spectrogram	CNN	Accuracy: 95.4% Recall: 96.3% Specificity: 92.4% Precision: 97.6% F1-score: 96.98%
Tariq et al. (2022)	6	Yes (Data Augmentation)	3-s segments, STFT spectrogram, MFCC, Chromagram	Fusion CNN	Accuracy: 97%

5.3.2. Discussion

Table 5.3 presents a summary of the results comparing the proposed model with several reference studies in the field of heart sound classification. Unlike the state-of-the-art review, which focused on the methodological aspects of each study, this section provides a direct comparison of the obtained results to contextualize the effectiveness of the developed approach.

The proposed model achieved an accuracy of 91.35%, recall of 84.97%, specificity of 93.47%, precision of 81.18%, and an F1-score of 83.43%. When compared with the study by Potes et al. (2016), which combined manually extracted time-frequency features with a CNN, the proposed model demonstrated a better overall balance between sensitivity and specificity. Although Potes et al. achieved a higher sensitivity (94.24%), their specificity was considerably lower (77.81%), indicating a greater occurrence of false positives. In contrast, the proposed model attained a substantially higher specificity (93.47%), demonstrating a more robust ability to

[6] Mean Accuracy

discriminate between normal and abnormal heart sounds while maintaining good generalization.

Regarding Rubin et al. (2017), who applied a similar criterion for spectrogram analysis but based on MFCCs, their model was trained using the same optimizer adopted in this work. The proposed model achieved superior results, with an improvement of 7.36% in accuracy and 12.19% in sensitivity. These results confirm that STFT-based representations preserve richer spectral and temporal information, enabling more effective recognition of cardiac cycle components.

In the case of Li et al. (2020), the authors trained a one-dimensional CNN (1D-CNN) using 497 manually extracted features from eight different domains, obtaining accuracy between 86% and 88%. Despite the extensive feature engineering and the use of balancing techniques, their performance remained below that of the proposed model, which relies exclusively on spectrograms and automatically learns its own features. This demonstrates that a purely deep learning-based approach can replace manual feature extraction while achieving higher and more generalizable performance.

The study by Salman Khan et al. (2021) shows the greatest methodological similarity to this dissertation, as it also employs STFT spectrograms as input to a two-dimensional CNN. Their model achieved an accuracy of 95.4% and a recall of 96.3%, slightly higher than those obtained in this work. This difference is mainly attributed to the use of 8-second windows, which provide more temporal information, the application of transfer learning techniques (pre-training of the model), and the combination of multiple datasets (PhysioNet and PASCAL), which increases variability and reduces overfitting. Even so, the proposed model, using only 2-second windows, achieved comparable specificity (93.47%) with lower computational cost and simpler architecture.

Tariq et al. (2022) achieved 97% accuracy using a feature fusion model that combines spectrograms, MFCCs, and chromograms, along with two data augmentation techniques, additive noise and pitch shifting, like those applied in the present work. Although this multimodal strategy provides the highest overall performance, it also introduces greater architectural complexity and higher computational cost. The proposed model, based solely on a single-stream CNN with STFT spectrograms, achieved competitive results with a lighter and more efficient architecture, making it more suitable for real-time implementations or systems with limited computational resources.

In summary, the proposed model presents a balanced trade-off between performance and architectural complexity, clearly outperforming traditional feature-based approaches such as those of Potes et al. (2016) and Li et al. (2020). Furthermore, its stability, generalization capability and low computational cost, reflected in the use of a single-stream CNN with short 2-second inputs and without pre-training or feature-fusion modules, demonstrate its suitability for integration into automatic heart-sound classification systems, especially in scenarios with limited processing resources.

Table 5.3 - Comparative evaluation of CNN models with PhysioNet Dataset (normal vs abnormal)

Author (year)	Accuracy	Recall	Specificity	Precision	F1-Score	Macc
Potes et al. (2016)	86.02 %	94.24 %	77.81 %	-	-	-
Rubin et al. (2017)	83.99 %	72.78 %	95.21 %	-	-	-
Li et al. (2020)	88,6 %	76.4 %	91.9 %	-	-	84.1 %
	86.7 %	87 %	86.6 %	-	-	86.8 %
Salman Khan et al., (2021)	95.4 %	96.3 %	92.4 %	97.6 %	96.98 %	-
Tariq et al. (2022)	97 %	-	-	-	-	-
Proposed Approach	91.35 %	84.97 %	93.47 %	81.18 %	83.43 %	-

5.4. Limitations

Despite the solid performance achieved by the proposed model, several limitations must be acknowledged. Firstly, the modelling process does not yet establish a definitive or systematic procedure that guarantees the selection of optimal architecture or hyperparameters. The results obtained, although encouraging, still rely on empirical tuning and may not represent the upper bound of performance achievable with this type of approach.

Secondly, the study relied exclusively on STFT-based spectrograms for feature representation. While this choice proved effective, it leaves open the possibility that alternative representations such as MFCCs, wavelet scalograms, Mel spectrograms or hybrid time–frequency descriptors could further improve classification performance. Without exploring these additional feature domains, it cannot be concluded that the proposed architecture constitutes the best performing configuration among all potential alternatives.

Chapter 6
Conclusion

In this study, two distinct classification approaches were developed: one aimed at identifying the different events of the cardiac cycle and another focused on the classification of cardiac pathologies. For both approaches, several convolutional neural network (CNN) architectures were explored with the objective of determining the most suitable configuration for each task. The STFT spectrograms were generated from the segmented PCG signals, corresponding to event-based cardiac windows and 2-second signal segments. These spectrograms were then used as input to the tested architectures, enabling the network to automatically learn the most relevant features.

The experiments were conducted using the publicly available PhysioNet 2016 heart sound dataset, which presents an uneven distribution of samples across classes. To mitigate this imbalance and increase model robustness, data augmentation techniques were applied, specifically additive noise and pitch shifting. These methods also contributed to reducing the risk of overfitting and improving the model's generalization capability.

Throughout multiple iterations of training and validation, different activation functions (ReLU and GELU) were tested, and the CNN hyperparameters were tuned to achieve an optimized configuration. This process involved progressive parameter refinement and comparative analysis of the models' performance.

The final selected model demonstrated competitive results, achieving an accuracy of 91.35%, recall of 84.97%, precision of 81.18%, specificity of 93.47%, and an F1-score of 83.43%. When compared with the studies by Potes et al. (2016), Rubin et al. (2017), and Li et al. (2020), the proposed model achieved superior performance. In relation to Khan et al. (2021) and Tariq et al. (2022), the results were slightly lower, which is expected since those authors employed more complex models involving multimodal fusion architectures, pre-trained networks, and higher computational power.

Overall, the results demonstrate that deep learning stands out as a more effective approach than traditional machine learning, as it eliminates the need for manual feature extraction and allows the model to autonomously learn discriminative representations directly from the data.

For future work, it is proposed to continue the approach based on classifying the events within each cardiac cycle, whose initial implementation was developed in the context of this thesis but could not be experimentally validated due to time and resource constraints. In addition, it is intended to explore deeper models with a greater number of convolutional layers and more advanced regularization mechanisms (such as batch normalization and adaptive dropout), as well as the use of pre-trained networks and transfer learning techniques applied to larger databases (Salman Khan et al., 2021; Torres, 2021). Another promising direction is the integration of different signal representations (time, frequency, and time-frequency) through multimodal fusion or hybrid convolutional neural network-recurrent neural network (CNN-RNN) architectures, such as Bidirectional Long Short-Term Memory (BiLSTM)-CNN models, which have shown strong

potential for capturing complex temporal dependencies in cardiac sound classification (Alam et al., 2018). Additionally, the adaptation of bio-inspired optimization methods, including Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), may further enhance model design by enabling automated hyperparameter tuning, architecture search and efficient feature selection, contributing to more robust and computationally efficient solutions (Rahman et al., 2021; Yang, 2021).

References

Abrams, J. (2005). Physical Examination of the Heart and Circulation. In *Essential Cardiology* (pp. 99–115). Humana Press. https://doi.org/10.1007/978-1-59259-918-9_7

Alam, S., Banerjee, R., & Bandyopadhyay, S. (2018). *Murmur Detection Using Parallel Recurrent & Convolutional Neural Networks*. http://arxiv.org/abs/1808.04411

Alhussein, M., Muhammad, G., & Hossain, M. S. (2019). EEG Pathology Detection Based on Deep Learning. *IEEE Access*, *7*, 27781–27788. https://doi.org/10.1109/ACCESS.2019.2901672

Alves de Brito, R. (2012, July 20). *DC Offset*. [Accessed June 2025]

Azam, F. B., Ansari, Md. I., Nuhash, S. I. S. K., McLane, I., & Hasan, T. (2022). Cardiac anomaly detection considering an additive noise and convolutional distortion model of heart sound recordings. *Artificial Intelligence in Medicine*, *133*, 102417. https://doi.org/10.1016/j.artmed.2022.102417

Bami, Z., Behnampour, A., & Doosti, H. (2025). *A New Flexible Train-Test Split Algorithm, an approach for choosing among the Hold-out, K-fold cross-validation, and Hold-out iteration*. https://doi.org/10.48550/arXiv.2501.06492

Benjamin, E. J., Virani, S. S., Callaway, C. W., Chamberlain, A. M., Chang, A. R., Cheng, S., Chiuve, S. E., Cushman, M., Delling, F. N., Deo, R., de Ferranti, S. D., Ferguson, J. F., Fornage, M., Gillespie, C., Isasi, C. R., Jiménez, M. C., Jordan, L. C., Judd, S. E., Lackland, D., … Muntner, P. (2018). Heart disease and stroke statistics - 2018 update: A report from the American Heart Association. https://doi.org/10.1161/CIR.0000000000000558 Circulation, 137(12), E67–E492.

Bergmann, D., & Stryker, C. (2024, July 2). *What is backpropagation?* IBM . https://www.ibm.com/think/topics/backpropagation. [Accessed in August 2025]

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

C. Potes, S. Parvaneh, A. Rahman and B. Conroy, "Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds," *2016 Computing in Cardiology Conference (CinC)*, Vancouver, BC, Canada, 2016, pp. 621-624.

C.Baker, B. (2016). How to compare your circuit requirements to active-filter approximations. *Analog Applications Journal, 1–6.*

Chakir, F., Jilbab, A., Nacir, C., Hammouch, A., & Hajjam El Hassani, A. (2016). Detection and identification algorithm of the S1 and S2 heart sounds. *2016 International Conference on Electrical and Information Technologies (ICEIT)*, 418–420. https://doi.org/10.1109/EITech.2016.7519633

Chaudhary, M. (2020, August 28). *Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax*. Medium. [Accessed in August 2025]

Chebil, J., Al, J., & Al-Ahliyya, N. (2007). Classification of heart sound signals using discrete wavelet In Article in International Journal of Soft Computing. https://www.researchgate.net/publication/287592634

Chen, W., Sun, Q., Chen, X., Xie, G., Wu, H., & Xu, C. (2021). Deep Learning Methods for Heart Sounds Classification: A Systematic Review. *Entropy*, *23*(6), 667. https://doi.org/10.3390/e23060667

Chizner, M. A. (2008). Cardiac Auscultation: Rediscovering the Lost Art. Current Problems in Cardiology, 33(7), 326–408. https://doi.org/10.1016/j.cpcardiol.2008.03.003

Choi, S., & Jiang, Z. (2010). Cardiac sound murmurs classification with autoregressive spectral analysis and multi-support vector machine technique. Computers in Biology and Medicine, 40(1), 8–20. https://doi.org/10.1016/j.compbiomed.2009.10.003

Chugh, V. (2025). *Tutorial do Python pandas: O guia definitivo para iniciantes*. DataCamp. https://www.datacamp.com/pt/tutorial/pandas . [Accessed August 2025]

Cowan, D. (2025). *Confusion Matrix*. ML-Science. https://www.ml-science.com/confusion-matrix. [Accessed in August 2025]

Di Cesare, M., McGhie, D. V., Perel, P., Mwangi, J., Taylor, S., Pervan, B., Kabudula, C., Narula, J., Bixby, H., Pineiro, D., Gaziano, T. A., & Pinto, F. J. (2024). The Heart of the World. Global Heart, 19(1). https://doi.org/10.5334/gh.1288

Dwivedi, A. K., Imtiaz, S. A., & Rodriguez-Villegas, E. (2019). Algorithms for Automatic Analysis and Classification of Heart Sounds–A Systematic Review. IEEE Access, 7, 8316-8345. https://doi.org/10.1109/ACCESS.2018.2889437

Emanuel, G. de S. (2019, March 27). *Entendendo o que é Matriz de Confusão com Python*. Medium. [Accessed in August 2025].

Fattah, S. A., Rahman, N. M., Maksud, A., Foysal, S. I., Chowdhury, R. I., Chowdhury, S. S., & Shahanaz, C. (2017). Stetho-phone: Low-cost digital stethoscope for remote personalized

healthcare. *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 1–7. https://doi.org/10.1109/GHTC.2017.8239325

GeeksforGeeks. (2025). *What is Forward Propagation in Neural Networks?* GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/what-is-forward-propagation-in-neural-networks/. [Accessed in August 2025]

Guissous, A. E. (2019). *Skin Lesion Classification Using Deep Neural Network*.

Hall, J. E., & Guyton, A. C. (2011). *Tratado de Fisiologia Médica* (R. Guedes, Ed.; 12º Edição). Elsevier.

Hendrycks, D., & Gimpel, K. (2023). *Gaussian Error Linear Units (GELUs)*. ArXiv:1606.08415

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, and K. Sricharan, "Recognizing abnormal heart sounds using deep learning," arXiv preprint arXiv:1707.04642, 2017.

Jain, A. (2024, February 16). *Data Augmentation*. Medium. [Accessed August 2025]

Jaros, R., Koutny, J., Ladrova, M., & Martinek, R. (2023). Novel phonocardiography system for heartbeat detection from various locations. *Scientific Reports*, *13*(1), 14392. https://doi.org/10.1038/s41598-023-41102-8

Keeton, Kimberly. (2016). *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*. USENIX Association.

Lau, K. W., Po, L.-M., & Rehman, Y. A. U. (2024). Large Separable Kernel Attention: Rethinking the Large Kernel Attention design in CNN. *Expert Systems with Applications*, *236*, 121352. https://doi.org/10.1016/j.eswa.2023.121352

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, M. (2023). *GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance*. ArXiv:2305.12073

Li, F., Tang, H., Shang, S., Mathiak, K., & Cong, F. (2020). Classification of Heart Sounds Using Convolutional Neural Network. *Applied Sciences*, *10*(11), 3956. https://doi.org/10.3390/app10113956

Li, S., Li, F., Tang, S., & Xiong, W. (2020). A Review of Computer-Aided Heart Sound Detection Techniques. BioMed https://doi.org/10.1155/2020/5846191 Research International, 2020, 1–10.

Liao, X., Wu, Y., Jiang, N., Sun, J., Xu, W., Gao, S., Wang, J., Li, T., Wang, K., & Li, Q. (2023). Automated detection of abnormal respiratory sound from electronic stethoscope and mobile phone using MobileNetV2. *Biocybernetics and Biomedical Engineering*, *43*(4), 763–775. https://doi.org/10.1016/j.bbe.2023.11.001

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, C., Springer, D., Li, Q., Moody, B., Juan, R. A., Chorro, F. J., Castells, F., Roig, J. M., Silva, I., Johnson, A. E. W., Syed, Z., Schmidt, S. E., Papadaniil, C. D., Hadjileontiadis, L., Naseri, H., Moukadem, A., Dieterlen, A., Brandt, C., Tang, H., … Clifford, G. D. (2016). An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, *37*(12), 2181–2213. https://doi.org/10.1088/0967-3334/37/12/2181

Mckinney, W. (2017). *Python for Data Analysis*.

Meziani, F., Debbal, S. M., & Atbi, A. (2012). Analysis of phonocardiogram signals using wavelet transform. *Journal of Medical Engineering & Technology*, *36*(6), 283–302. https://doi.org/10.3109/03091902.2012.684830

Miguel, J., Pereira, S., Manuel, J., & Tavares, R. S. (2011). Analysis of Structures in Medical Images: Application to the Cardiovascular System.

Morris, R., Ahmed, M., Drotman, S., & Salamanca-Padilla, Y. Y. (2021). Basic Cardiovascular Physiology. In *Cardiac Anesthesia* (pp. 21–35). Springer International Publishing. https://doi.org/10.1007/978-3-030-51755-7_2

Nogueira, D. M., Ferreira, C. A., Gomes, E. F., & Jorge, A. M. (2019). Classifying Heart Sounds Using Images of Motifs, MFCC and Temporal Features. *Journal of Medical Systems*, *43*(6), 168. https://doi.org/10.1007/s10916-019-1286-5

Pazin-Filho, A., Schmidt, ; André, Benedito, &, & Maciel, C. (2004). *AUSCULTA CARDÍACA: BASES FISIOLÓGICAS-FISIOPATOLÓGICAS CARDIAC AUSCULTATION: PHYSIOLOGICAL AND PHYSIOPATHOLOGICAL MECHANISMS*.

Pechetty, R., & Nemani, L. (2020). Additional Heart Sounds—Part 1 (Third and Fourth Heart Sounds). Indian Journal of Cardiovascular Disease in Women WINCARS, 5(02), 155–164. https://doi.org/10.1055/s-0040-1713828

Rabiza, M. (2024). *A Mechanistic Explanatory Strategy for XAI*. https://orcid.org/0000-0001-6217-6149

Rahman, M. A., Sokkalingam, R., Othman, M., Biswas, K., Abdullah, L., & Abdul Kadir, E. (2021). Nature-Inspired Metaheuristic Techniques for Combinatorial Optimization Problems: Overview and Recent Advances. *Mathematics*, *9*(20), 2633. https://doi.org/10.3390/math9202633

Rath, A., Mishra, D., Panda, G., & Pal, M. (2022). Development and assessment of machine learning based heart disease detection using imbalanced heart sound signal. *Biomedical Signal Processing and Control*, *76*, 103730. https://doi.org/10.1016/j.bspc.2022.103730

ReLU Activation Function in Deep Learning. (2025, July 23). GeeksforGeeks. [Accessed in August 2025]

Ribeiras, R. (2022, May 27). *Válvulas cardíacas e valvulopatias*. Hospital Da Luz. https://www.hospitaldaluz.pt/pt/dicionario-de-saude/valvulas-cardiacas-e-valvulopatias [Accessed in April 2025]

Ríos-Prado, R., Anzueto-Ríos, Á., & Tovar-Corona, B. (2019). Metodología para discernir entre sonido cardiaco no patológico de regurgitación y estenosis aórtica, empleando DTW. Revista de La Facultad de https://doi.org/10.15446/rev.fac.cienc.v8n1.74802 Ciencias, 8(1), 138–155.

Roquette, J. (2023, January 19). Tetralogia de Fallot. (2023). https://hospitaldaluz.pt/pt/dicionario-de-saude/tetralogia-fallot [Accessed in November 2024]

Rosas, A. E., & Ayala, G. G. (2014). *Fisiología - Cardiovascular, Renal y Respiratioria* (1ºEdición). Editorial El Manual Moderno.

Russel, S., & Norvig, P. (2022). Artificial Intelligence: A Modern Approach. In *PEARSON SERIES IN ARTIFICIAL INTELLIGENCE* (fourth). Pearson Education Limited.

S.Sathyanarayanan, Sanjay Chitnis, & Srikanta Murthy. (2023). A Comprehensive Survey of Analysis of Heart Sounds using Machine Learning Techniques to Detect Heart Diseases. Journal of Population Therapeutics https://doi.org/10.47750/jptcp.2023.30.11.038 and Clinical Pharmacology, 30(11).

Salman Khan, M., Nawaz Khan, K., Ahmad Khan, F., Abid, A., Olmez, T., Dokur, Z., Khandakar, A., & H Chowdhury, M. E. (2021). *Deep Learning Based Classification of Unsegmented Phonocardiogram Spectrograms Leveraging Transfer Learning*. https://doi.org/10.48550/arXiv.2012.08406

Santo, A. (2016). Caraterização funcional cardíaca por fonocardiografia.

Santos, M. V. (2023, February 28). *Estetoscópio*. Med Estratégia. https://med.estrategia.com/portal/conteudos-gratis/resumo-de-estetoscopio-diagnostico-tratamento-e-mais/

Santos, M., & Lucindo, J. (n.d.). FISIOLOGIA DO SISTEMA CARDIOVASCULAR.

Sarti, G. (2023, July 5). *Ciclo Cardíaco*. Med Estratégia. https://med.estrategia.com/portal/conteudos-gratis/ciclo-cardiaco/ [Accessed in April 2025]

SEELEY, Rod R., STEPHENS, Trent D., TATE, Philip (2004) – Anatomia e Fisiologia- 6º edição. Loures: Lusociência – Edições Técnicas e Científicas, Lda. ISBN 972- 8930-07-0

Softmax Activation Function in Neural Networks. (2025). GeeksforGeeks. [Accessed in August 2025]

Springer, D., Tarassenko, L., & Clifford, G. (2015). Logistic Regression-HSMM-based Heart Sound Segmentation. *IEEE Transactions on Biomedical Engineering*, 1–1. https://doi.org/10.1109/TBME.2015.2475278

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

T. M. Mitchell, *Machine learning*, vol. 1, no. 9. McGraw-hill New York, 1997

T. R. Reed, N. E. Reed e P. Fritzson, (2004) "Análise do som do coração para deteção de sintomas e diagnóstico assistido por computador", Simul. Model. Pract. Theory, vol. 12, no. 2, pp. 129-146.

Taneja, K., Arora, V., & Verma, K. (2023). Classifying the heart sound signals using textural-based features for an efficient decision support system. *Expert Systems*, *40*(6). https://doi.org/10.1111/exsy.13246

Tariq, Z., Shah, S. K., & Lee, Y. (2022). Feature-Based Fusion Using CNN for Lung and Heart Sound Classification. *Sensors*, *22*(4), 1521. https://doi.org/10.3390/s22041521

TensorFlow Core. (2023). *Keras: The high-level API for TensorFlow*. https://www.tensorflow.org/guide/keras

Teoh, T. T. (2023). *Convolutional Neural Networks for Medical Applications* (S. Zdonik, S. Shekhar, X. Wu, L. C. Jain, D. Padua, X. Sherman Shen, B. Furht, V. S. Subrahmanian, M. Hebert, K. Ikeuchi, B. Siciliano, S. Jajodia, & N. Lee, Eds.). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-8814-1

Thomas, S. L., Heaton, J., & Makaryus, A. N. (2024). Physiology, Cardiovascular Murmurs. https://pubmed.ncbi.nlm.nih.gov/30247833/

Torres, J. (2021). *Deteção de patologia em sons cardíacos usando deep learning*.

Trifunović-Zamaklar, D., Jovanović, I., Vratonjić, J., Petrović, O., Paunović, I., Tešić, M., Boričić-Kostić, M., & Ivanović, B. (2022). The basic heart anatomy and physiology from the cardiologist's perspective: Toward a better understanding of left ventricular mechanics, systolic, and diastolic function. In Journal of Clinical Ultrasound (Vol. 50, Issue 8, pp. 1026 1040). John Wiley and Sons Inc. https://doi.org/10.1002/jcu.23316

Tsang, S.-H. (2023, December 19). *Brief Review — CNN and Bidirectional GRU-Based Heartbeat Sound Classification Architecture for Elderly People*. Medium.

Wright, B. E., Watson, G. L., & Selfridge, N. J. (2020). The Wright table of the cardiac cycle: a stand-alone supplement to the Wiggers diagram. Advances in Physiology Education, 44(4), 554–563. https://doi.org/10.1152/advan.00141.2019

Yang, X.-S. (2021). Genetic Algorithms. In *Nature-Inspired Optimization Algorithms* (Second Edition, pp. 91–100). Elsevier. https://doi.org/10.1016/B978-0-12-821986-7.00013-5

Yu, K.-H., Beam, A. L., & Kohane, I. S. (2018). *Artificial intelligence in healthcare*. Nature Biomedical Engineering, 2(10), 719–731.

Zhang, W., & Han, J. (2017, September 14). *Towards Heart Sound Classification Without Segmentation Using Convolutional Neural Network*. https://doi.org/10.22489/CinC.2017.254-164

Zhou, X., Guo, X., Zheng, Y., & Zhao, Y. (2023). Detection of coronary heart disease based on MFCC characteristics of heart sound. *Applied Acoustics*, *212*, 109583. https://doi.org/10.1016/j.apacoust.2023.109583

Zilliz. (2025). *Activation Functions in Neural Networks*. Zilliz. https://zilliz.com/glossary/activation-functions. [Accessed in August 2025]

Annex I

Operations in CNNs

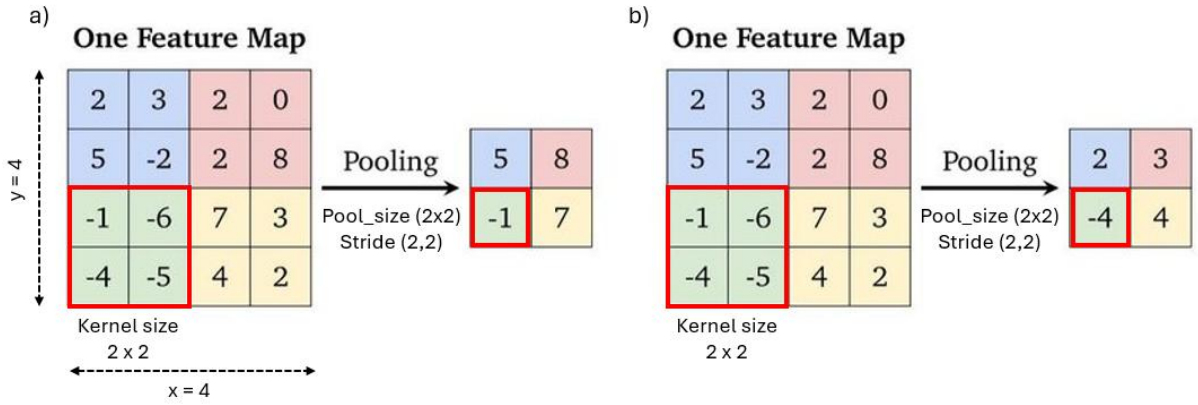

Figure A.1 - Illustration of pooling operations in Convolutional Neural Networks (CNNs). a) Max pooling and b) Average pooling, showing the dimensionality reduction of feature maps (Guissous, 2019) [adapted].

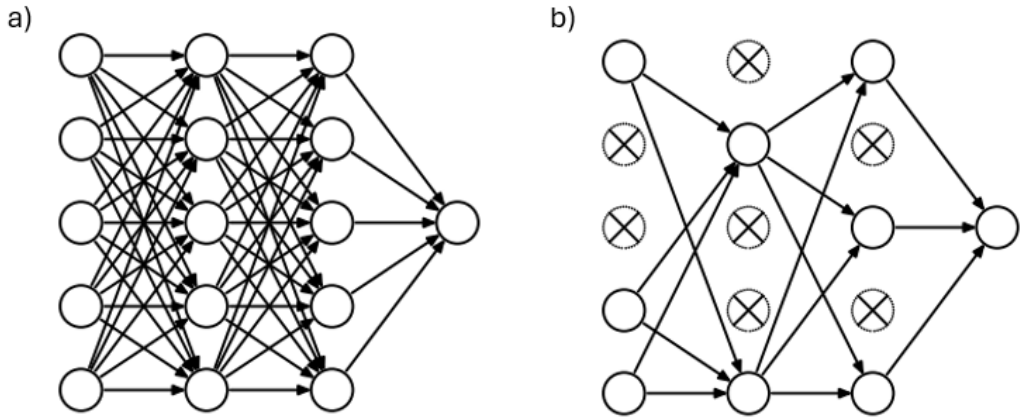

Figure A.2 - Represents of the dropout technique in neural networks: a) Standard fully connected network and b) the same network after applying dropout, where a subset of neurons is randomly deactivated during training (Srivastava et al., 2014) [adapted].

Spectrogram (STFT)

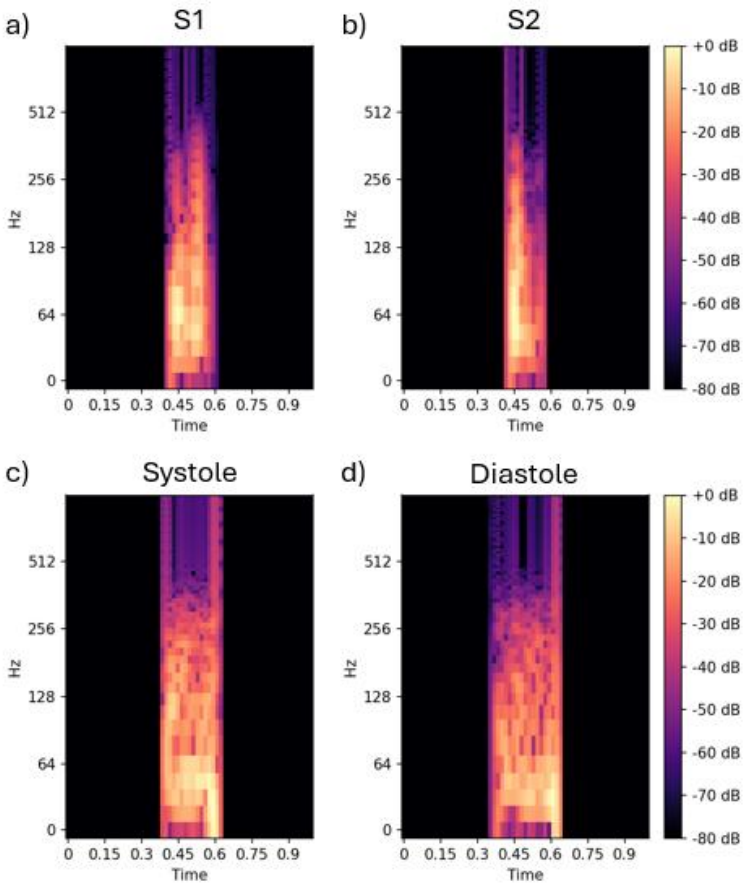

Figure A.3 – Spectrogram representation of the fundamental cardiac events (c0003.wav) from the original database, without data augmentation.

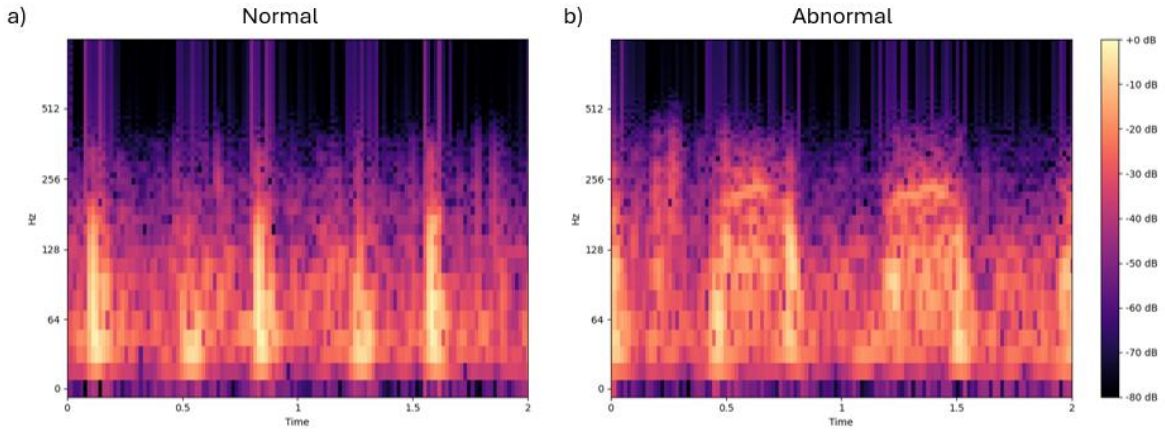

Figure A.4 – Spectrograms of normal and abnormal heart sounds (a0080.wav and a0002.wav) from the original database, without data augmentation.

Summary of model hyperparameters

Table A.1 - Parameters of model 1 (Sigmoid output).

Layers (type)	Output Shape	Parameters
Conv2D	(None, 64, 125,128)	640
MaxPooling2D	(None, 21, 41, 128)	0
Dropout	(None, 21, 41, 128)	0
Conv2D	(None, 20, 40, 256)	131328
MaxPooling2D	(None, 6, 13, 256)	0
Dropout	(None, 6, 13, 256)	0
Conv2D	(None, 5, 12, 128)	131200
MaxPooling2D	(None, 1, 4, 128)	0
Dropout	(None, 1, 4, 128)	0
Flatten	(None, 512)	0
Dropout	(None, 512)	0
Dense	(None, 1)	513
Total Parameters:	263 681	
Trainable Parameters:	263 681	
Non-Trainable Parameters:	0	

Table A.2 - Parameters of model 2 (Sigmoid output).

Layers (type)	Output Shape	Parameters
Conv2D	(None, 63,124,128)	1280
MaxPooling2D	(None, 21, 41 ,128)	0
Dropout	(None, 21,41 ,128)	0
Conv2D	(None,19, 39,256)	295168
MaxPooling2D	(None,6 ,13 ,256)	0
Dropout	(None,6 ,13 ,256)	0
Conv2D	(None,4 ,11 ,128)	295040
MaxPooling2D	(None,1 ,3 128)	0
Dropout	(None,1 ,3 128)	0
Flatten	(None, 384)	0
Dropout	(None, 384)	0
Dense	(None, 1)	385
Total Parameters:	591 873	
Trainable Parameters:	591 873	
Non-Trainable Parameters:	0	

Table A.3 - Parameters of model 3 (Sigmoid output).

Layers (type)	Output Shape	Parameters
Conv2D	(None, 63, 124, 128)	1280
MaxPooling2D	(None, 21, 41, 128)	0
Dropout	(None, 21, 41, 128)	0
Conv2D	(None, 19, 39, 512)	590336
MaxPooling2D	(None, 6, 13, 512)	0
Dropout	(None, 6, 13, 512)	0
Conv2D	(None, 4, 11, 128)	589952
MaxPooling2D	(None, 1, 3, 128)	0
Dropout	(None, 1, 3, 128)	0
Flatten	(None, 384)	0
Dropout	(None, 384)	0
Dense	(None, 1)	385
Total Parameters:	1 181 953	
Trainable Parameters:	1 181 953	
Non-Trainable Parameters:	0	

Table A.4 - Parameters of model 4 (Sigmoid output).

Layers (type)	Output Shape	Parameters
Conv2D	(None, 63, 124, 128)	1280
MaxPooling2D	(None, 31, 62, 128)	0
Dropout	(None, 31, 62, 128)	0
Conv2D	(None, 29, 60, 256)	295168
MaxPooling2D	(None, 14, 30, 256)	0
Dropout	(None, 14, 30, 256)	0
Conv2D	(None, 12, 28, 128)	295040
MaxPooling2D	(None, 6, 14, 128)	0
Dropout	(None, 6, 14, 128)	0
Conv2D	(None, 4, 12, 64)	73792
MaxPooling2D	(None, 2, 6, 64)	0
Dropout	(None, 2, 6, 64)	0
Flatten	(None, 768)	0
Dropout	(None, 768)	0
Dense	(None, 1)	769
Total Parameters:	666 049	
Trainable Parameters:	666 049	
Non-Trainable Parameters:	0	

Performance metrics model

Table A.5 - Performance metrics of model 1 (Sigmoid output).

Model 1	Metrics (%)	ReLU	GELU
Training	Accuracy	94.53	96.57
	Precision	92.66	93.65
	Recall	96.12	99.54
	Specificity	93.08	93.86
	F1-score	94.36	96.51
Validation	Accuracy	88.77	88.04
	Precision	71.74	68.30
	Recall	84.27	89.33
	Specificity	90.12	87.65
	F1-score	77.50	77.41
Test	Accuracy	90.56	90.19
	Precision	79.85	76.23
	Recall	83.03	88.04
	Specificity	93.05	90.90
	F1-score	81.41	81.71

Table A.6 - Performance metrics of model 2 (Sigmoid output).

Model 2	Metrics (%)	ReLU	GELU
Training	Accuracy	97.63	99.53
	Precision	96.03	99.36
	Recall	99.13	99.65
	Specificity	96.27	99.42
	F1-score	97.55	99.51
Validation	Accuracy	89.63	89.75
	Precision	73.06	75.09
	Recall	86.80	82.80
	Specificity	90.47	91.82
	F1-score	79.34	78.76
Test	Accuracy	91.35	90.56
	Precision	81.18	81.00
	Recall	84.97	81.09
	Specificity	93.47	93.69
	F1-score	83.03	81.05

Table A.7 - Performance metrics of model 3 (Sigmoid output).

Model 3	Metrics (%)	ReLU	GELU
Training	Accuracy	97.13	99.41
	Precision	95.69	99.14
	Recall	98.40	99.63
	Specificity	95.96	99.21
	F1-score	97.03	99.38
Validation	Accuracy	88.96	80.00
	Precision	72.28	90.83
	Recall	84.13	87.20
	Specificity	90.39	73.90
	F1-score	77.76	90.00
Test	Accuracy	91.01	90.36
	Precision	81.34	79.18
	Recall	82.92	83.14
	Specificity	93.69	92.75
	F1-score	82.12	81.11

Table A.8 - Performance metrics of model 4 (Sigmoid output).

Model 4	Metrics (%)	ReLU	GELU
Training	Accuracy	96.31	96.96
	Precision	95.66	99.44
	Recall	96.64	94.16
	Specificity	96.01	99.51
	F1-score	96.14	96.73
Validation	Accuracy	90.95	88.77
	Precision	77.68	81.86
	Recall	84.93	65.60
	Specificity	92.74	95.67
	F1-score	81.15	72.83
Test	Accuracy	90.70	86.44
	Precision	82.66	83.33
	Recall	79.27	56.95
	Specificity	94.49	96.22
	F1-score	80.93	67.66