POLITECNICO DI TORINO

Biomedical Engineering Master's Degree

Biomedical Instrumentation Specialisation

UCC

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Master's Degree Thesis

"Detecting Congenital Heart Defects on an Edge platform using Phonocardiogram and AI"

Supervisors:

Prof. Danilo Demarchi
Dr. Emanuel Popovici

Candidato:

Giuseppe Caracciolo

Academic Year 2022 – 2023
Italy, Turin, 21st July 2023

Abstract

Congenital Heart Defects (CHD) are heart malformations caused by abnormal heart development. These malformations can induce a wide range of clinical symptoms, indicating that this vital organ is underperforming. The earlier one can detect these malformations, the better patient outcomes. Ultrasound is used for the early detection of CHD.

However, because antenatal and postnatal ultrasounds are in short supply, diagnosis is generally performed based on other solutions which are more easily available. In a resource-constraint context, where ultrasound screening is highly limited, these alternative methods may become even more crucial. Routine CHD screening is executed in such settings by the mean of a multi-dimensional clinical test that includes, among others, pulse oximetry and auscultation. Although auscultation is subject to interpretation, as some cardiac aberrations are not always audible, it facilitates cardiac defects detection.

Some novel Artificial Intelligence (AI) driven methodology for the detection of CHD has been developed at the Embedded.Systems@UCC research group. The purpose of this study is to implement a clinical decision-making device relying on AI to aid in the clinical differentiation of sounds affected by CHDs. This work includes implementation and evaluation for Machine Learning (ML) based Segmentation, Feature Extraction and Classification on a Raspberry Pi device. In fact, validation of these techniques on EDGE IoT (Internet of Things) devices is paramount towards the early detection of CHDs. The final goal is to realise a first Demo of a portable, rapid, and low-cost Phonocardiogram (PCG) signals real-time monitoring system that merges the previous research into a unique device.

The equipment chosen for this purpose is:

- Raspberry Pi model 4 (RPi4): a single board computer that saves the received soundtracks through a jack port and processes them with a Python algorithm.

- Thinklabs One Digital Stethoscope: this medical device records and shares sounds, providing various solutions for Telemedicine, Education, Research and Electronic Medical Records (EMR).

It is important to underline that this implementation serves as the initial demonstration of the entire research project. Indeed, the developed system is optimised only in terms of the execution time of each section. The basic idea and demo will support additional optimisation for the algorithm and the embedded system.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

BPM	Beat Per Minute
AI	Artificial Intelligence
AUC	Area Under the Curve
AV	Atrio Ventricular
BLE	Bluetooth Low Energy
CHD	Congenital Heart Defects
CPU	Central Process Unit
CV	Cross Validation
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
ECG	Electrocardiogram
EMR	Electronic Medical Records
ET	Execution Time
FE	Feature Extraction
FHS	Fundamental Heart Sound
HR	Heart Rate
IoT	Internet of Things
LED	Light Emitting Diode
LPF	Low Pass Filter
MHR	Maximum Heart Rate
ML	Machine Learning

PCG	Phonocardiogram
PDA	Patent Ductus Arteriosus
PP	Pre-Processing
RAM	Random Access Memory
RPi 4	Raspberry Pi 4
RTS	Relative Time of the Silence periods
TFCF	Tonal Deviation from the average Central Frequency
WAV	Wave Audio format
XGBoost	eXtreme Gradience Boosting

List of pubblications

Conference Presentation

S. Gomez-Quintana, V. Shelevytska, **G. Caracciolo**, A. Factor, E. Popovici, A. Temko Hearttone: An accurate, objective decision support tool for detecting congenital heart disease using phonocardiograms and artificial intelligence, oral presentation at the 9th Congress of the European Academy of Paediatric Society, Barcelona, 7-10 Oct 2022;

Journal Publication

S. Gomez-Quintana, V. Shelevytska, **G. Caracciolo**, A. Factor, E. Popovici, A. Temko Hearttone: An accurate, objective decision support tool for detecting congenital heart disease using phonocardiograms and artificial intelligence, Frontiers in Pediatrics; Frontiers Media SA; The European Academy of Paediatrics (EAP), The European Society for Paediatric Research (ESPR), The European Society of Paediatric and Neonatal Intensive Care (ESPNIC). (2022). 9th Congress of the European Academy of Paediatric Societies, ISBN: 978-2-88971-024-9 DOI: 10.3389/978-2-88971-024-9

Chapter 1

Introduction, Motivations and Goals

This thesis aims to implement an embedded solution for a portable device able to merge and put into practice a biomedical research project regarding heart abnormalities detection developed in the Embedded Systems Group at University College Cork and divided into two branches: PCG automated segmentation and CHDs detection assisted by AI.

1.1 Motivations

The heart develops very early in the gestation period, and it is one of the first organs developing in the embryo [1]. Neonatal deaths from congenital heart abnormalities (CHD), which afflict approximately eight newborns out of each 1000, account for 3% of all newborns deaths. Nearly 18 to 25 per cent of the affected newborns die during their first year, whereas 4 per cent of those who survive infancy die by age 16. Consequently, CHDs are among the leading causes of newborns deaths [2]. Sophisticated and well-equipped settings allow identifying the majority of CHDs using prenatal ultrasound, which enables identification of the heart abnormality starting from 12–16 weeks of gestation. Antenatal diagnosis accuracy is still restricted, and a sizable number of heart abnormalities go undiscovered [3].

Auscultation is a component of newborns' clinically essential examination, as well as for adults. Prior to the nineteenth century, a sort of direct auscultation, known as "immediate" and consisting of listening to the heartbeat by placing an ear against the patient's chest, was utilised by clinicians to investigate breath and cardiac sounds [4]. Reports of this technique are dated even to the 15th century BC [5-6]. The stethoscope (Figure 1.1.a) was invented only in 1816 by R T H Laënnec, avoiding the embarrassment of the previous method [7].

The Laënnec stethoscope was simply a tubular structure made of wood which channelled cardiac sounds from the patient to the doctor's ears. The evolution of the stethoscope is shown in Figure 1.1, from the Laënnec's (a) to the classic stethoscope (b), with the newest technology of the digital stethoscope, better described in the next chapter.

Figure 1.1: a) Laënnec's stethoscope [Wikipedia] and b) modern standard stethoscope [Wikipedia]

Stethoscopes are a common tool used by healthcare providers to detect various medical conditions, including CHDs. Stethoscopes are an essential tool in the early detection of CHD as they allow healthcare providers to listen to the heart sounds and detect any abnormalities.

While stethoscopes are a useful tool in CHD detection, their accuracy can be limited, and they may not detect all abnormalities. In particular, some types of CHD may not produce audible heart sounds, making detection with a stethoscope difficult. Furthermore, the accuracy of stethoscope detection may depend on the clinician's skill and experience and may be subject to interpretation errors.

However, because of physiological shunt sounds generated across the Ductus Arteriosus, evaluating cardiac sounds through auscultation right after birth might be challenging. The Ductus Arteriosus plays a right-to-left shunting function in the foetus' heart by allowing blood that flows out from the right ventricle to circumvent the pulmonary circulation. Shunt direction changes during the postnatal transition because of the pulmonary circulation starting with the first breath.

The Ductus Arteriosus shuts in the great majority of term newborns within two to three postnatal days [8-10]. In contrast, postponed ductal closure (arising passed three days) is classified as problematic and joins the CHD spectrum with the name of Patent Ductus Arteriosus (PDA) [11-13]. Stethoscopes can be used to detect various types of CHD, including PDA. Different cases of ductal closure are shown in Figure 1.2.

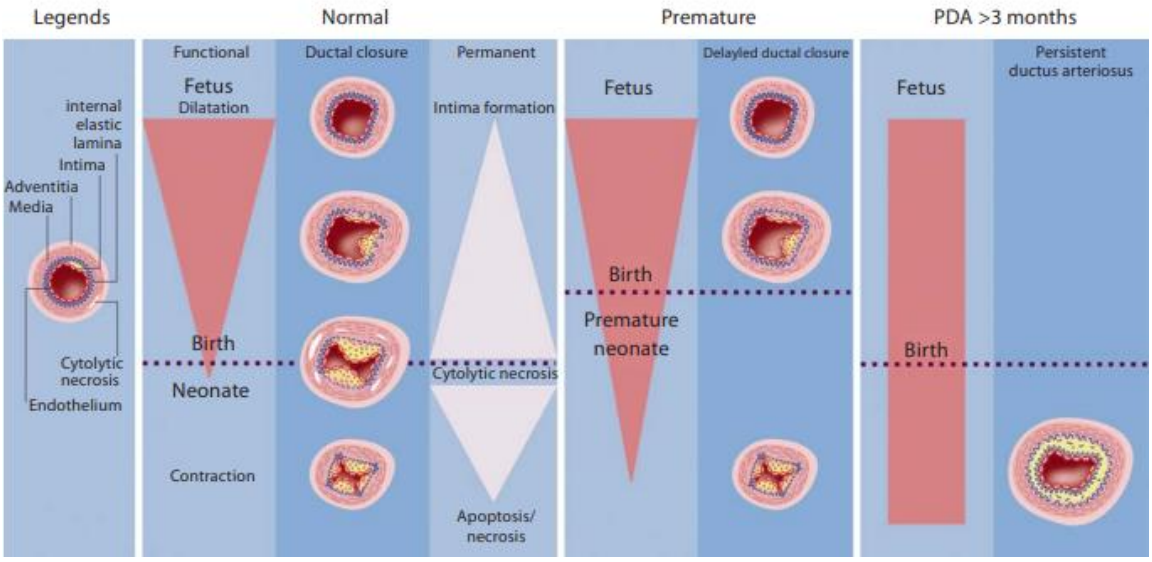

Figure 1.2: Schematic representation of the anatomical and functional closing process of the human Ductus Arteriosus [9].

In agreement with a precedent study [14], the amount of neonatal diagnoses of CHDs over a ten-year time frame is 39%, a rate which is not rising. The number of serious CHDs (causing a potential risk of premature death with the need for medical treatments) is one out of two ranging between 13% and 87% [15]. The sensitivity of complex CHDs detection has grown from 29.8 to 88.3 per cent.[16-17]

1.2 New Concept

Early diagnosis of abnormalities such as PDA and other CHDs during the first few days of birth enables clinicians to make conscious decisions that could save lives. Continuous monitoring is crucial for this, and it presents many difficulties in a clinical context. Indeed, gathering physiological data from newborns is not an easy task.

It sometimes entails drawn-out procedures that necessitate specialised staff. Moreover, sophisticated monitoring instrumentation is usually expensive. Finally, even for experienced staff, who may not be available at any time, the data's complexity may make it hard to comprehend.

Thus, there is a need for alternative techniques and equipment that are cheaper, simple to use, and accurate to the same degree as a specialist medical practitioner. Only a few research works have been focused on a solution for the automated interpretation of PCG in newborns. Starting from two of those, carried out at University College Cork, this research focuses on creating a portable device capable of recording neonatal heart sounds and making predictions about possible diseases and dysfunctions affecting the heart. Thus, the final aim is to prevent neonatal deaths by optimising care for patients whose heart defects are challenging to detect with other methods, such as conventional auscultation and ultrasounds, because of efficiency or availability constraints.

Murmurs are sounds which can be heard between cardiac sounds while auscultating. Some murmurs are pathological, while others are physiological and innocent, like those due to the ductus arteriosus.

To address some of these limitations, new technologies, such as digital stethoscopes and AI algorithms, are being developed to improve CHD detection. Digital stethoscopes can capture and record heart sounds, which can be analysed and visualized by healthcare providers or AI algorithms to improve detection accuracy.

Given the heart murmurs subjective interpretation through auscultation, Artificial intelligence (AI) assistance provides support, offering, as a supplement to the conventional approach, a solution for the objective interpretation of cardiac sounds [18]. Large datasets can be used to train ML, producing objective judgments unaffected by a person's perception, level of exhaustion, or mood.

AI has revolutionized various fields of healthcare, including neonatology, by providing accurate and efficient interpretation of various medical data, such as PCG recordings. AI algorithms can be trained to analyse these recordings and provide clinicians with insights into the infant's cardiac health.

One of the main advantages of using AI in neonatal PCG interpretation is the speed and accuracy of diagnosis. In a traditional setting, PCG interpretation requires a skilled clinician to listen to and analyse the heart sounds, which can be time-consuming and subject to interpretation errors. AI algorithms, on the other hand, can analyse vast amounts of PCG data in a quick and accurate way, providing clinicians with a reliable and objective assessment of the infant's cardiac function. Finally, AI can help standardize CHD detection across different healthcare settings, reducing the risk of misdiagnosis and ensuring consistency in patient care.

Another benefit of using AI in neonatal PCG interpretation is the ability to detect subtle abnormalities in the heart's function that may go unnoticed by even the most experienced clinicians. AI algorithms can be trained to identify even the smallest variations in heart sounds, which can help detect early signs of cardiac dysfunction and lead to timely interventions.

However, it's important to note that AI is not a replacement for clinical expertise, but rather a tool that can support and enhance clinical decision-making. AI algorithms should be developed and validated using large datasets to ensure their accuracy and reliability. Clinicians should also be involved in the interpretation of AI-generated results to ensure that any abnormalities detected are properly assessed and treated. Overall, the integration of AI into neonatal PCG interpretation has the potential to improve clinical outcomes for infants with cardiac conditions, by providing faster and more accurate diagnoses.

1.3 First end-to-end Demo

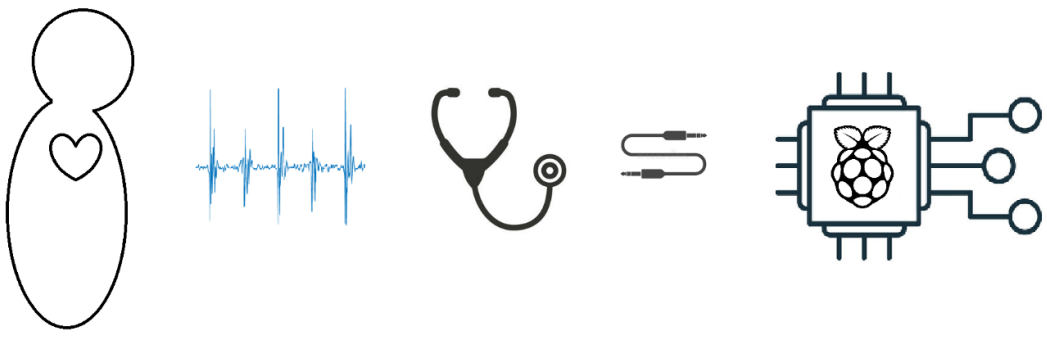

Figure 1.3: First Demo concept.

The embedded system designed, conceptualised in Figure 1.3, allows recording through the Thinklabs One digital stethoscope a sequence of 5 soundtracks, one for each of the auscultation points (Figure 1.4). The five auscultation points are important because each one highlights different aspects of the cardiac rhythm [19].

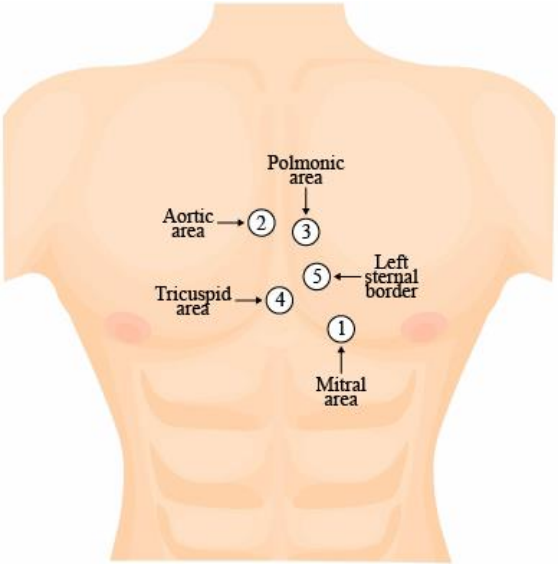

Figure 1.4: Auscultation Points.

These five auscultation points commonly used by healthcare providers correspond to the positions of the heart valves. The five auscultation points are:

- Aortic area: located in the second right intercostal space at the sternal border. This is where the aortic valve can be heard, which is responsible for preventing blood from flowing back into the heart's left ventricle.
- Pulmonic area: located in the second left intercostal space at the sternal border. This is where the pulmonic valve can be heard, which is responsible for preventing blood from flowing back into the heart's right ventricle.
- Tricuspid area: located in the fourth left intercostal space at the sternal border. This is where the tricuspid valve can be heard, which separates the right atrium and right ventricle.
- Mitral area: centred in the fifth left intercostal area at the mid-clavicular line. This is where the mitral valve can be heard, which separates the left atrium and left ventricle.
- Erb's point: found in the third left intercostal area at the sternal border. This point is not a specific valve listening area but rather a site where both the

aortic and pulmonic sounds can be heard equally, and it is useful for detecting murmurs and other abnormalities.

Auscultation at these five points is critical for assessing the heart's function and detecting any abnormalities, such as heart murmurs or valve regurgitation. Understanding the location and characteristics of the heart sounds heard at these points is essential for accurate diagnosis and effective treatment of various heart conditions.

The five recordings acquired by the digital stethoscope are then sent and stored into the RPi4 board using the Thinklabs Link adapter and a stereo jack cable to be successively processed from a computation algorithm previously uploaded into the RPi4. This system, at last, returns a final decision about the possible presence of abnormalities in the heart sounds, a signature of PDA or CHDs.

A first boosted decision tree classifier receives the features to assign the cycle phase to each sound and silence. A second boosted decision tree classifier uses the averaged and rearranged features to calculate the likelihood of PDA or other CHDs.

The algorithm was implemented in Python 3.6 programming language. In particular, the contributions to this study are:

- Implemented a Python algorithm for heart sound automated segmentation starting from an existing Matlab script.
- Adjusted and optimised a Feature Extraction algorithm for the first classification with an XGBoost classifier – the cycle phase classification.
- Implemented an algorithm to compute features average over a single patient data and rearrange the resulting table in order to obtain a single raw of features per patient.
- Adjusted and optimised an algorithm for the second XGBoost classifier, which returns the Normal-Abnormal heart sounds final prediction.
- Merged all the above sections to get a complete pipeline algorithm that, having a heart sound as input, returns its final prediction.
- Implemented a code to interact with the RPi4, which allows to record and store the audio signal received by the digital stethoscope, ready to be processed by the software.

Chapter 2

Methods

The following chapter discusses the research projects that are the basis of this thesis. The primary qualities and features of the algorithms and formulas that underlie the entire application are explained, with an overview of the instrumentation in use.

For the implementation of the system under study, two approaches are available: cloud-based and edge devices. They both have advantages and disadvantages. The fundamental difference lies in the location of computing resources and where data processing occurs.

The cloud-based solution works with information transmission from the source to a computational device, which frees the source of the computing load, as shown in [20]. These systems offer scalability, flexibility, and accessibility since users can access their applications and data from anywhere with an internet connection. Cloud-based systems excel in handling large-scale data processing, complex computations, and resource-intensive tasks. They leverage the power of centralized processing and storage capabilities, enabling organizations to offload computational burdens and reduce hardware costs. Additionally, cloud-based systems often provide advanced features such as automated backups, high availability, and seamless software updates. Bandwidth and memory capacity are the principal advantages of this technique, and possible optimisation methods related to these issues are illustrated in [21].

On the contrary, edge devices are localized computing devices situated closer to the source of data generation or user interaction. These devices, which can range from smartphones and laptops to Internet of Things (IoT) devices, possess their own processing capabilities and store data locally.

The primary advantage of edge devices is low latency and real-time processing. Since data is processed locally, there is minimal delay in transmitting information to a remote server and receiving a response. This attribute is crucial for time-sensitive applications, such as autonomous vehicles, industrial automation, and remote healthcare monitoring, where immediate decisions or actions are required. Edge devices also provide offline functionality, allowing them to continue operating even when internet connectivity is limited or unavailable. An example is illustrated in [22].

In terms of data security and privacy, cloud-based systems typically implement robust security measures and rely on specialized teams to manage and safeguard data stored in centralized locations. Edge devices, on the other hand, do not involve sensitive data transmission over servers because the source of information is kept near the computational device, decreasing the possibility of cyber-attacks and breaches. This decentralized approach can enhance privacy and compliance with data protection regulations. However, edge devices may be more susceptible to physical theft or local security vulnerabilities.

Both cloud-based systems and edge devices have their place in today's computing landscape, and often a combination of the two is employed to optimize performance and efficiency. Cloud-based systems excel in handling large-scale data analysis, AI-driven applications, and resource-intensive workloads, while edge devices provide real-time processing, offline capabilities, and enhanced data privacy.

The choice between these approaches depends on factors such as the nature of the application, required latency, data volume, connectivity availability, and security requirements.

The whole analysis process of the algorithm has been performed in a cloud-based mode, which means that all the PCG signals are previously made available on a cloud, where the algorithm will process them and return a prediction to the clinician (Figure 3.1).

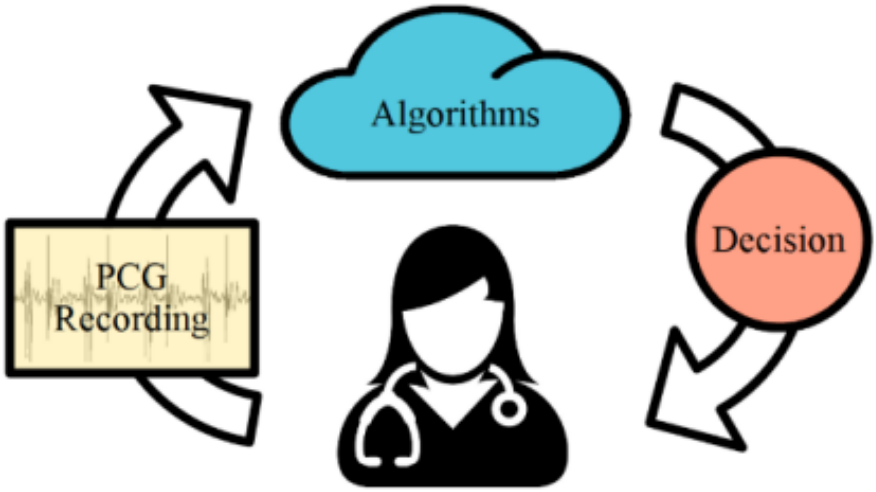

Figure 2.1: Cloud-based system scheme [19].

Only the final implementation (Figure 2.2), simulating a real application, is an edge-based implementation based on real-time acquisition and further digital signal processing (DSP) supported by AI.

Figure 2.2: Edge-based system scheme.

2.1 Instrumentation

The edge device implemented during this study mainly consists of a microcontroller, described in the next chapter, and a digital stethoscope.

The digital stethoscope is the ThinkLabs One shown in Figure 2.3, characterised by 16 bits of resolution, a sampling frequency of 44100Hz and a rechargeable Lithium Ion Battery. It features a LED display with a battery level and volume indicator and two buttons, which allow setting volume and frequency range of filtering. Finally, it presents on the side a Jack port to connect the charger and headphones to listen to the recording, other than linking the stethoscope to a computer for saving the audio files and afterwards processing them.

Figure 2.3: ThinkLabs One Digital Stethoscope.

The ThinkLabs Thinklink Mobile Kit shown in Figure 2.4 is a necessary accessory to the stethoscope provided by the manufacturer to connect the stethoscope to a mobile or a computer. It presents three Jack ports: the left side has the headphones connector on the top and the input from the stethoscope at the bottom, while on the right side, it features the output port that goes to the computer.

Figure 2.4: ThinkLabs Thinklink Mobile kit.

One more tool is necessary to connect the digital stethoscope to the computer and to process the audio recordings: a Jack to USB adapter. For this project, the uGo Sound Card with a Jack Splitter, shown in Figure 2.5, was used. The Jack Splitter input on the right is connected to the output port of the Thinklink module, which is divided into two channels "microphone" and "speaker". The two Jack channels are finally converted into USB by the Sound Card.

Figure 2.5: uGo external Sound Card and Jack Splitter.

The digital stethoscope does not feature a built-in speaker. Therefore, the use of headphones appears to be essential for the clinician to directly listen to the Heart Sound if not linking the stethoscope to a mobile. For the current application, headphones are necessary mainly for checking the correct position of the stethoscope before starting a recording.

For this study, the Thinklabs Earbud Headphones displayed in Figure 2.6 are used. They provide a deep bass which is extremely useful for the low frequencies of the heartbeat.

Figure 2.6: Thinklabs Earbud Headphones.

The Raspberry Pi Sense HAT shown in Figure 2.7 is a supplementary board that enhances Raspberry Pi functionalities. It includes a wide range of built-in sensors:

- Temperature
- Humidity
- Barometric Pressure
- Magnetometer
- Accelerometer
- Gyroscope
- Colour and brightness

None of these measurements is required for this study, but the important features of the board for this application are the five-button joystick on the left-bottom part and the 8x8 RGB LED matrix, which works as a display for the outputs of the algorithm.

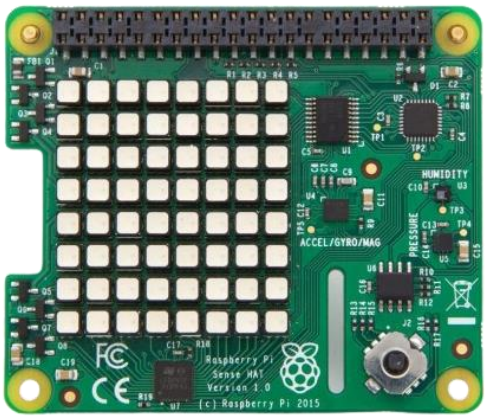

Figure 2.7: Pi Sense Hat module.

2.2 Full Pipeline

This section aims to give an overview of the full algorithm pipeline, i.e. all the sequential steps that, from a given PCG signal, finally return the prediction regarding the presence of CHDs. These sections of the main algorithm (Figure 2.8) and their functions will be described following in this chapter.

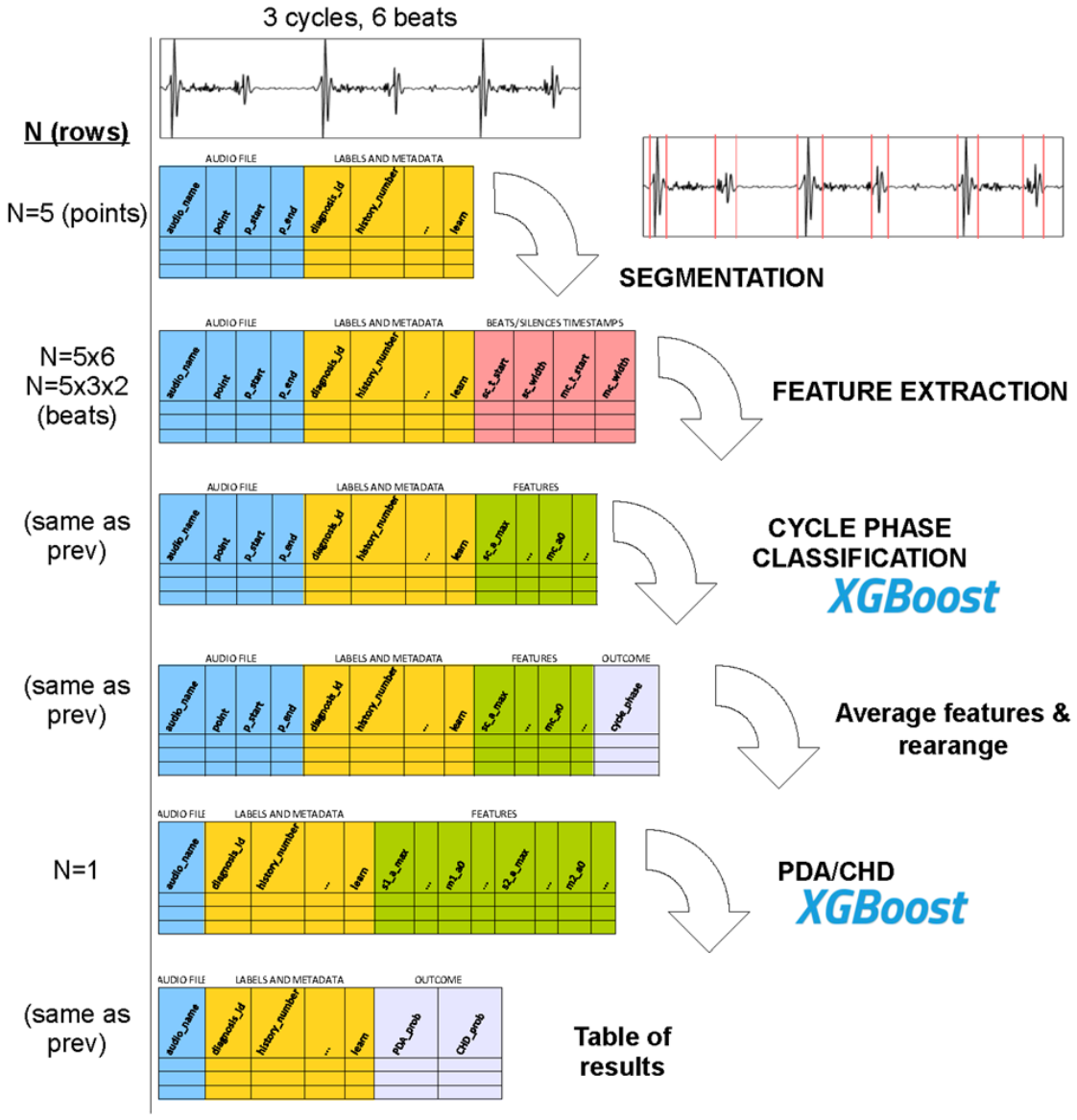

Figure 2.8: Final prediction algorithm chart.

2.3 Segmentation

The aim of this first block of code is to detect the Fundamental Heart Sounds (FHS) in a PCG Signal and to classify them by timestamps. At this stage, the two different sounds S1 and S2, as well as for the two silences m1 and m2, are not distinguished. Unlike auscultation, consisting of the clinician listening to the patient's cardiac sound, phonocardiography enables graphical visualisation of the heartbeat waveform and its subsequent analysis [23]. A single heartbeat consists of a sequence of two sounds separated by silences.

The vibrations produced by the closure of the heart valves are what cause the sounds identified by PCG: the first sound, defined as S1, is produced when the atrioventricular (AV) valves (also known as tricuspid and mitral) shut at the beginning of systole, while the second sound, defined as S2, is produced when the aortic and pulmonary valves (also known as semilunar valves) shut at the end of systole. Figure 2.9 shows an excerpt of a neonatal PCG.

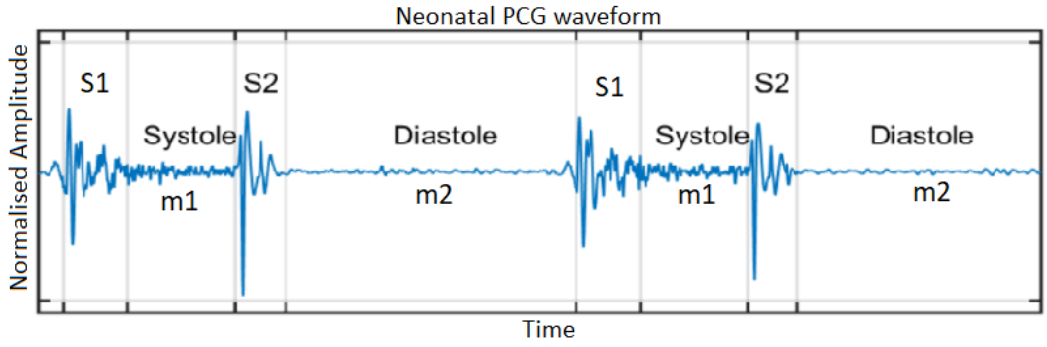

Figure 2.9: Sound phases – PCG waveform of two consecutive cardiac cycles and the related sound phases: Sound 1 (S1), Silence 1 (m1), Sound 2 (S2), Silence 2 (m2).

PCG and ECG signals are related: the R-peak represented in the ECG trace corresponds to the start of S1 in the PCG trace, while the T-wave corresponds to the final instant of S2 [24].

The automated segmentation of a PCG signal consists exactly of identifying those two fundamental heart sounds (S1 and S2) and their related silences (m1 and m2) to annotate their start and final time/sample. This enables the availability of 5 times the number of cardiac cycles obtained with the manual segmentation, which is a tedious task to execute.

A bigger availability of data results in more trained ML models. This breakthrough potentially improves the performance of the algorithm in detecting CHDs [25].

There is scientific evidence of the fact that the Heart Rate decreases with age. This applies to the Maximum Heart Rate as well, the only free variable of the segmentation algorithm, which is modelled in relation to age by the following formula empirically derived in previous studies [26-29]:

$$MAX\_HearRate = 220 - age\ [Beat\ Per\ Minute]$$

This parameter is considered the limit of the expected Heart Rate for a correct estimation of the PCG periodicity and the subsequent classification of the fundamental cardiac sounds.

The automated segmentation block has been translated into Python and customised for the current application from the Matlab algorithm implemented in [25]. It consists, as shown in Figure 2.10, of:

- Low Pass Filter (LPF), with a cut-off frequency of 1000Hz.
- A compressor that decreases the amplitude of the PCG waveform and normalises it in relation to its peak magnitude.
- Fast Envelope: a double-sided exponential impulse response with a time constant τ of 15ms, which is the twentieth part of the maximum HR (MHR), considered equal to 200 Beats Per Minute.
- Slow Envelope: it consists of an LPF with a cut-off frequency of 3.3Hz, equal to the MHR, applied to the output of the fast envelope function.
- Identify slow-envelope peak values as local maxima, which can be considered in first approximation as the Fundamental Heart Sounds location. By extracting the position of these peaks in terms of samples, an initial estimation of the Heart Rate is determined as the reciprocal of the median of the difference between each peak start time and the second next peak (because each Sound 1 alternates a Sound 2).

$$Heart\ Rate = \frac{1}{Median[s(i) - s(i-2)]_1^N}$$

Where s is the array containing the location of each peak.

- Find a more accurate position of the peaks in the fast envelope by identifying its maxima within a half-cycle frame (considering the previous Heart Rate estimation).
- Identify the heartbeats' initial and final samples as the samples in which the fast envelope decreases by 12.5% from the peak magnitude.
- Find the next zero-crossing location in the initial Phonocardiogram (in both directions) to fine-tune initial and final samples.

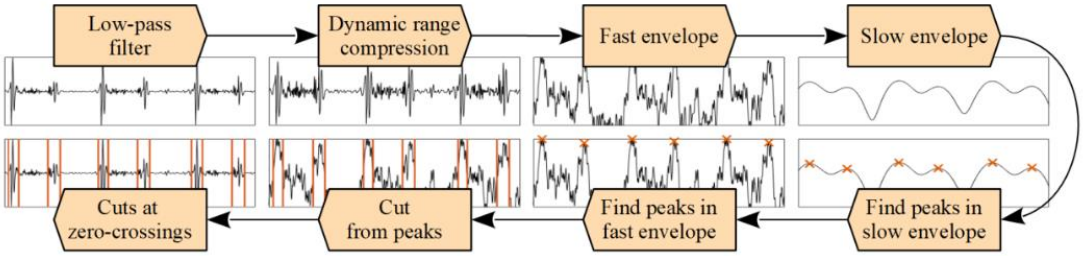

Figure 2.10: Flowchart of the automatic segmentation block [25].

The result of the automatic segmentation is represented in Figure 2.11, which is obtained by overlaying the original PCG signal and the segmented signal, i.e. the sequence containing the initial and final points of each FHS. This sequence is obtained by creating an array of the same length as the PCG and assigning the value 1 to the samples that are located between the initial and final samples identified for each of the peaks and 0 to the remaining samples.

This plot shows how the implemented algorithm is able to identify and annotate each beat.

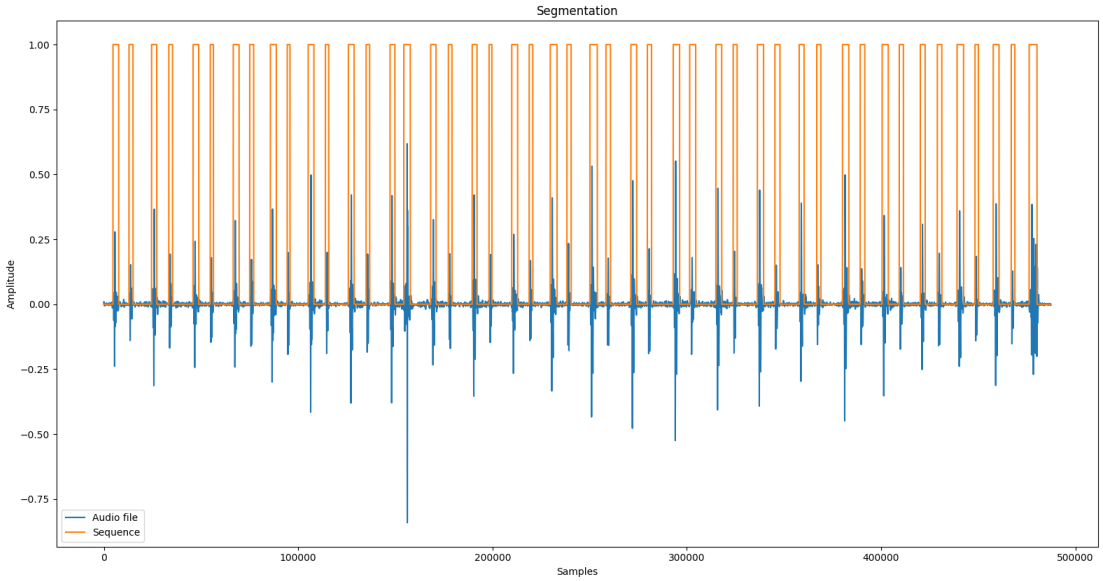

Figure 2.11: Result of the automatic segmentation.

In contrast to manual segmentation, which typically consists of the selection of 2-5 cycles per auscultation point (10-25 per patient), automated segmentation makes available all the heartbeats detected from each of the recordings.

2.4 Feature extraction

Once the PCG has been segmented, and all the beats have been identified, those need to be classified into Sound 1 and sound 2. Afterwards, a further classification for cardiac defects will be performed. The whole classification procedure begins with the extraction of 101 features by each segment [19].

2.4.1 Dataset

A sizable clinical set of cardiac sound recordings from 265 babies with a gestation period of 35 to 42 weeks, characterised by the presence of different CHDs and acquired using the ThinkLabs One digital stethoscope during their initial six days following birth, has been used to train and evaluate the system. Patients' clinic details are listed in Table 1.1.

Table 2.1: Dataset's breakdown [30].

Sex	
Males	137 (52%)
Gestational age [weeks]	**Median 39, IQR 38-40**
Less than 37 weeks (preterm)	23 (9%)
37 weeks or more (term)	242 (91%)
Postnatal age [hours]	**Median 48, IQR 30-64**
Less than 24h	29 (11%)
From 24 to 48h	95 (36%)
From 48h to 72h	87 (33%)
More than 72h	54 (20%)
Diagnostics	
Healthy	137 (52%)
PDA	89 (33%)
CHD	39 (15%)
Total number of patients	**265**

The dataset, for a total of 468 minutes of recording, was gathered from two Ukrainian Hospitals in 2018, and every participant was asked to provide informed consent from parents. All newborns analysed were apparently healthy and had no evident symptoms of CHDs or high blood pressure in the lungs caused by PDA, but they all had a diagnosis validated by ECG [19].

To simplify, each patient is assigned to a single diagnosis class. Patients suffering from both PDA and CHD were categorised as belonging to the CHD class, which is a diagnosis of primary concern for newborns.

The audio files are saved in a Wave Audio format (WAV), which provides enough audible frequency content to convey the information of interest for PCG signals. This format is preferred to the MP3 one, a well-known standard for lossy audio compression (i.e. the encoding implies a particular loss/distortion of the original information). [19]

2.4.2 Features

Because of the differences in the PCG waveform's amplitude and structure at each of the four cardiac sound phases (s1, s2, m1, m2), distinct features are retrieved from every single phase [31]. A total of 101 features are extracted by each of the segments regarding their aspects in terms of time, frequency, and energy. A total of 200 features are finally obtained for each of the cardiac cycles. Some of the features are obtained from each of the cardiac intervals (Table 2.2), while other features are extracted only from the two sounds intervals (Sound 1 and Sound 2) (Table 2.3) or only from the two silences (m1 and m2) related to the systolic and diastolic intervals (Table 2.4). Furthermore, the average BPM and the related cycle duration complete the features set.

A group of the extracted features was previously recognised as crucial to characterise newborns' PCG [32], while others were proposed in [19] for the first time.

Because the band frequency of the entire feature set is enclosed with 1000 Hz, the FE process was preceded by a downsampling operation to 2000 Hz.

For simplicity, the cycle phase classification, as well as the CHD/PDA classification, is carried out using models previously trained and tested in [19,25].

Table 2.2: Features extracted from all four FHS [19].

Index	Type	Tag	Description
1	(A)	energy	sum of squared values
2	(A)	n_zero	number of zero-crossings
For filtered signal 25-1000Hz:			
3	(B)	bw_en_lin	energy as sum of squared values
4	(B)	bw_en_db	energy in dB scale
5	(B)	bw_rms_lin	RMS
6	(B)	bw_rms_db	RMS in dB scale
For subands (k) 25-45, 45-80, 80-200, 200-400 & 400-1000Hz:			
7-10	(B)	bk_en_lin	energy as sum of squared values
11-14	(B)	bk_en_dB	energy in dB scale
15-18	(B)	bk_rms_lin	RMS
19-22	(B)	bk_rms_dB	RMS in dB scale
23	(C)	fc	central frequency (<200Hz)
24	(C)	oct	frequency deviation from average fc (per point), in octaves
25	(C)	trel	the relative length of the interval over the average length of the full cycle

Table 2.3: Features extracted only from cardiac sounds [19].

Index	Type	Tag	Description
Absolute extrema:			
26	(A)	a_max	maximum value (positive amplitude)
27	(A)	t_max	relative time location of a_max
28	(A)	a_min	the minimum value (negative amplitude)
29	(A)	t_min	relative time location of a_min
30	(A)	max_a	maximum absolute value
31	(A)	max_t	relative time location of max_a
Local extrema:			
32	(A)	mean_t_max	mean time across all relative maxima
33	(A)	mean_dt_max	mean-time difference across all relative maxima
34	(A)	std_t_max	SD over the time of all relative maxima
35	(A)	std_dt_max	SD over the time difference between all relative maxima
36	(A)	n_max	number of local maxima
37	(A)	mean_t_min	mean-time across all relative minima
38	(A)	mean_dt_min	mean-time difference across all relative minima
39	(A)	std_t_min	SD over the time of all relative minima
40	(A)	std_dt_min	SD over the time difference between all relative minima
41	(A)	n_min	number of local minima
42	(A)	mean_t_zero	mean-time across all zero-crossing
43	(A)	std_t_zero	SD over the time of all zero-crossing
44	(A)	mean_dt_zero	mean-time difference across all zero-crossing
45	(A)	std_dt_zero	SD over the time difference between all zero-crossing
46	(A)	mean_t_max	mean-time across all relative maxima
Structural:			
47	(A)	n_broken	number of discontinuities on the derivative of the signal
48	(A)	skewness	the relative position of the maximum absolute value of the signal around the middle point of the interval

Table 2.4: Features extracted only from cardiac silences [19].

Index	Type	Tag	Description
Envelope approximation using 2nd order polynomial coefficients:			
49	(A)	a0	constant term of the polynomial
50	(A)	a1	linear term of the polynomial
51	(A)	a2	quadratic term of the polynomial
Energy distributed across time (4 quarters):			
52	(A)	en_1/4	1st quarter energy
53	(A)	en_2/4	2nd quarter energy
54	(A)	en_3/4	3rd quarter energy
55	(A)	en_4/4	4th quarter energy
Statistics:			
56	(A)	mean	mean value
57	(A)	std	SD
Statistics distributed across time (4 quarters):			
58	(A)	mean_1/4	1st part mean
59	(A)	mean_2/4	2nd part mean
60	(A)	mean_3/4	3rd part mean
61	(A)	mean_4/4	4th part mean
62	(A)	std_1/4	1st part SD
63	(A)	std_2/4	2nd part SD
64	(A)	std_3/4	3rd part SD
65	(A)	std_4/4	4th part SD
Structural:			
66	(A)	frq_zero	number of zeros per second
67	(A)	skewness	relative position of the maximum absolute value of the signal around the middle point of the interval.

2.5 Cycle Phase Classification

The two most relevant features for distinguishing between the first and the second beat of a cardiac cycle are the Relative Time of the Silence periods (1) and the Tonal Deviation from the average Central Frequency (2) [25]. For each segment *i:*

$$RTS(i) = HR \cdot T_{silence}(i) \tag{1}$$

$$TFCF(i) = \log_2 \frac{f_c(i)}{avg[f_c(i)]_1^N} \tag{2}$$

where the Central Frequency is

$$f_c = \frac{f \cdot |X(f)|^2}{\sum |X(f)|^2}$$

$X(f)$ is the Discrete Fourier Transform (DFT) of each segment calculated over the frequency band 5-200Hz.

Using a first XGBoost (Extreme Gradient Boosting Decision Tree) algorithm, all the sounds and silences are classified as related to the first or second cycle phase and marked with the cycle phase adding an extra column. This algorithm, as well as the algorithm for the second classification routine, has been used to train and test a logistic regression model. The model section procedure used is a stratified subject-independent 10-fold CV process (Figure 2.12), and the models' parameters are set as follows: objective = binary: logistic, eval_metric = auc, eta = 0.03 (learning rate).

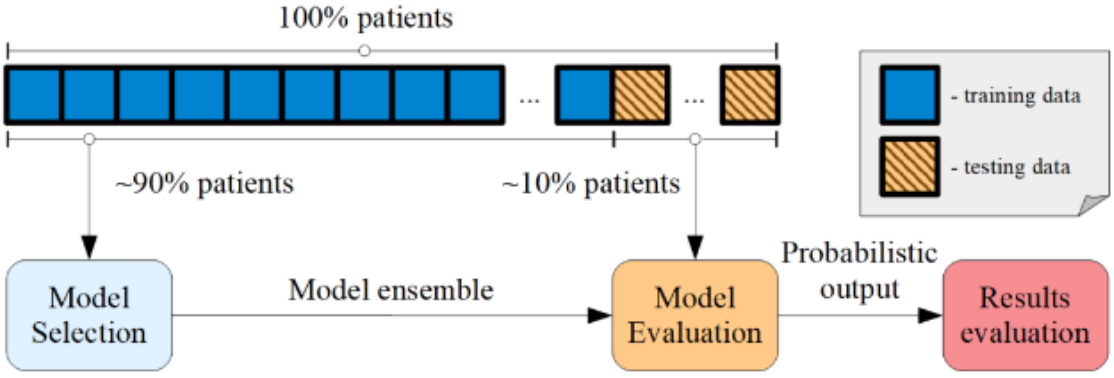

Figure 2.12: Model Evaluation procedure [19].

Using ML algorithms necessarily involves a well-defined and independent model selection process [33] to optimise a model across an array of hyper-parameters, ensuring its accuracy over an unknown test dataset. The nested Cross Validation routine shown in Figure 2.13 is used to tune these hyper-parameters.

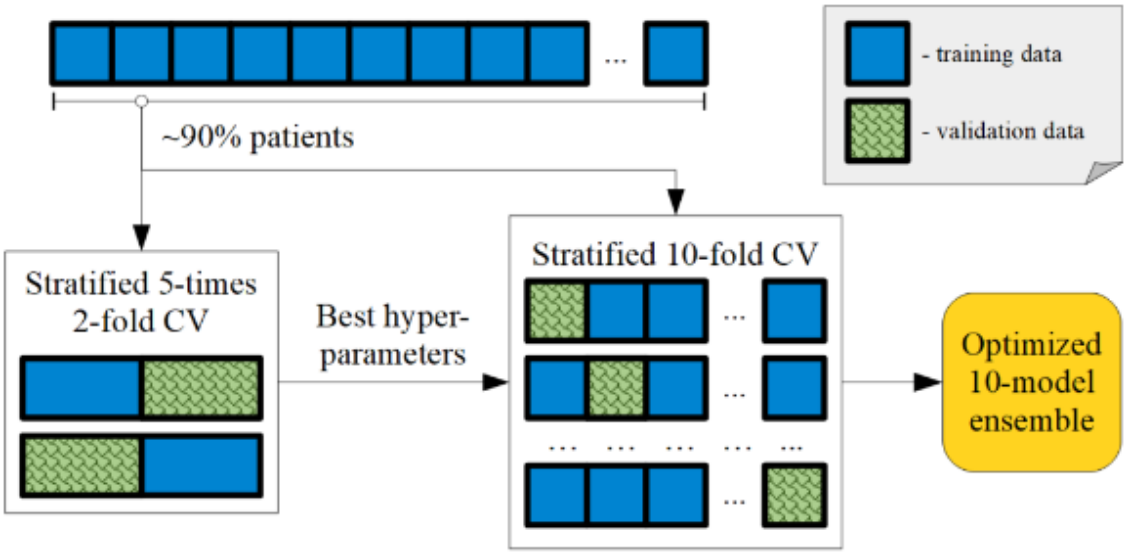

Figure 2.13: Model Selection procedure [19].

2.6 Averaging and Rearranging

Features for each patient are averaged based on the previous sound prediction carried out. The aim of this procedure is to get two rows per patient - one for each of the FHS with 101 features. Follow the final rearrangement of the data frame to get one row of feature per patient, which contains features from both cardiac sounds for a total of 200.

2.7 PDA/CHDs Classification

Following the Cycle Phase classification routine, two more XGBoost algorithms are used to finally predict the probability of having a patient with Patent Ductus Arteriosus (PDA) or other CHDs. Those algorithms are used to train and test two Logistic Regression models whose characteristics are described in [19].

The Area Under the Curve (AUC) metric was utilised to evaluate the accuracy of the implemented algorithm [34]. The performance of the final algorithm is shown in Table 2.5. From these results, it is evident that detecting PDA is more difficult than detecting CHD.

Table 2.5: Final Algorithm Performance [19].

Detection task	Validation (AUC) (mean±std)	Testing (AUC)
PDA	0.761±0.004	0.743
CHD	0.773±0.002	0.775

Chapter 3

Code Optimisations

The critical aspect is the algorithms' efficiency, measured through their execution times and power consumption. This project is evaluated in terms of inference. More precisely, execution time information has been extracted over different computers, with different amounts of data input and at different grades of code optimisation. In this chapter, optimisation processes that allowed a significant execution time reduction are explained.

The execution time of the algorithms would involve four main blocks: Pre-Processing, Automatic Segmentation, Feature Extraction, and Classification.

3.1 Computers

In this section, an overview of the computers used for this study and their main specifications will be given, with particular attention to the edge device object of the final implementation.

3.1.1 Dell Computer – Optiplex 3050

Main Specifications:

- Quad Core 3.4 GHz Intel i7-6700 64-bit CPU
- 8 GB RAM

3.1.2 Asus Laptop – N552VW

Main Specifications:

- Quad Core 2.6 GHz Intel i7-6700 64-bit CPU
- 16 GB RAM

Raspberry Pi 4 Model B

With features including USB and micro-HDMI ports, WiFi, Bluetooth, GPIO pins, ports and network boot, the RPi 4 Model B, represented in Figure 3.1, is a single-board computer which can carry out most of the operations that a standard PC can. For embedded applications like the current project, its compact design and reasonable price make it particularly appealing.

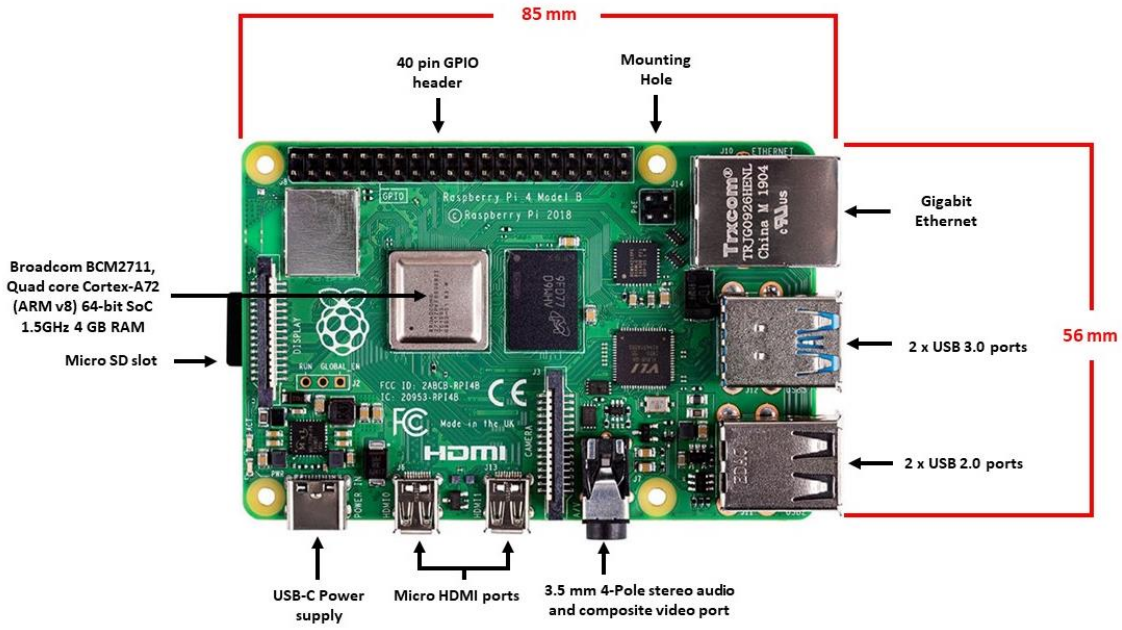

Figure 3.1 Raspberry Pi 4 Model B board.

The noteworthy specifications of the Raspberry Pi 4 Model B are:

- Quad-core 1.5 GHz BCM2711 64bit CPU
- 4 GB RAM
- BLE Gigabit Ethernet, 2.4-5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0
- Micro-SD card slot for data storage and loading of the operating system.

Having the RPi4 considerably less computing power than a normal regular PC/laptop, execution time information on this device has been evaluated only on the fully optimised code.

3.2 Resampling

Analysing and reviewing the code without optimisation, the attention went to the audio file loading function.

```
fs=2000
[x,fs]=librosa.load(audio_directory+'/'+audiofile,sr=fs,mono=Tru
e)
```

This function automatically resamples the audio signal from the initial sampling frequency of 44100 Hz to 2000 Hz (fs=2000), as expected, but using a high-quality mode. This is the default mode of the "librosa.load" resampling feature, which allows for improving audio quality but makes the loading process itself extremely slow.

```
fs=2000
[x,fs_in]=librosa.load(audio_directory+'/'+audiofile,sr=None,mono=T
rue)
x=fast_resample(x,fs,fs_in)
```

In order to reduce the execution time of this function, its automated resampling feature has been disabled (sr=None), and a new resampling function has been implemented:

```
def fast_resample(x,fs_out,fs_in):
    xf = x;
    if fs_in!=fs_out:
        if fs_in>fs_out:
            #perform antialias filter
            xf = AntiAliasFilter(xf,fs_out/2,fs_in)

        ti = np.arange(0,(len(xf))/fs_in,1/fs_in)
        to = np.arange(0,(len(xf))/fs_in,1/fs_out)
        y = np.interp(to, ti, xf)
    else:
        y = x
    return y
```

The above function is called fast resample because it allows for considerably speeding up the resampling process by simply performing an anti-aliasing filtering and returning the linear interpolant to the original time series evaluated at the new frequency.

3.3 Pre-Allocating Chunks

The first two Feature Extraction sub-section, "Extract_amplitude_features" (block A) and "Extract_band_features" (block B), both have a double nested loop that probably is one of the causes of the long execution time. Moreover, this double-nested loop operates the same signal partition in chunks for each of the blocks, basically repeating the same operation. Therefore, any optimisation needs to be done in this regard.

The main problem is that an individual x_chunk, which is a portion with a variable length of the full-length signal x, is obtained at every iteration of the inner loop:

```
for ix, elem in timestamps_audio.iterrows():
   for s in range(len(soundtype)):
       start_field = soundtype[s]+'_t_start'
       width_field = soundtype[s]+'_width'
       tstart = elem[start_field]
       tend = tstart + elem[width_field]
       x_chunk=x[np.round(tstart*fs).astype(int):np.round(tend*f
s).a
       stype(int)]
```

The identified solution was the pre-allocation of all those x_chunks into two lists of arrays – one for the heartbeat sounds and one for the silences (the two soundtypes). This pre-allocation is performed before the execution of the Amplitude and Band features extraction functions:

```
# Initialise lists of chunks
sc_c = []
mc_c = []
for ix, elem in timestamps_audio.iterrows():
    for s in range(len(soundtype)):
       start_field = soundtype[s]+'_t_start'
```

```
width_field = soundtype[s]+'_width'
tstart = elem[start_field]
tend = tstart + elem[width_field]
y_chunk=y[:,np.round(tstart*fs).astype(int):np.round(tend*fs
).astype(int)]
        if s == 0:
            sc_c.append(y_chunk)
        else:
            mc_c.append(y_chunk)
```

The result is a double nested loop which only performs the partition of the signal. Hence, the number of operations executed inside the nested loop has been substantially reduced. Consequently, the FE functions perform then just a single for loop, iterating over the pre-allocated lists and extracting the features from them.

Chapter 4

Results and Full Implemented Device

This chapter shows the results in terms of execution time reduction obtained through the whole optimisation process. Moreover, the final implementation of the Embedded System is presented.

4.1 Inference

Once a complete pipeline code was implemented, execution time (ET) information to process the full algorithm was obtained, both with a regular Lab computer (Dell computer) (Table 4.1) and with an Asus Laptop (Table 4.2).

Table 4.1: Full Pipeline Execution Time (seconds) – Dell Computer.

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	7.749	59.430	166.818
Sound Classification	0.047	0.062	0.156
Rearranging And Avg.	0.406	0.031	0.047
Final Classification	0.172	0.191	0.203

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Pre-Processing	3.858	30.320	83.157
Segmentation	0.219	1.640	4.783
Feature Extraction	3.593	25.776	76.396

Feature Extraction	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
A	1.953	13.700	41.321
B	1.375	9.841	29.179
C	0.266	1.781	5.378

Table 4.2: Full Pipeline Execution Time (seconds) – Asus Laptop.

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	15.458	110.444	329.773
Sound Classification	0.474	0.499	0.728
Rearranging And Avg.	0.041	0.045	0.074
Final Classification	0.257	0.268	0.270

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Pre-Processing	4.511	33.439	92.909
Segmentation	0.404	2.752	7.907
Feature Extraction	10.624	73.773	230.175

Feature Extraction	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
A	5.220	36.381	113.067
B	4.758	32.908	102.845
C	0.641	4.386	13.926

Columns of the previous tables give information about the number of audio files (patients) processed and the total length of the audio recording in terms of time (seconds) and cardiac cycles.

The results obtained with the Dell computer are comparable to the results obtained in [11].

Three different setups have been considered, simulating a cloud:

- One patient
- Ten patients
- Twenty-five patients

Analysing only Table 4.1, i.e., considering only the results obtained on the Lab computer, which is clearly the most performing device between the two, it is straightforward to identify which is, among all the macro sections of the "Full Pipeline" (upper part of the table), the one taking most of the time and requiring thus some optimisation. This section is related to segmentation and feature extraction ("Seg and FE").

Investigating it deeper made it clear that sections "Pre-Processing" and "Feature extraction" are the sub-sections that take more time to be executed. Optimisation work in this project was focused on these two segments of the code and evaluated only on the Dell Computer.

4.1.1 Pre-Processing Optimisation

The first optimisation was performed over the Pre-Processing (PP) sub-section. Table 4.3 shows the new performance in terms of execution time of the optimised section.

Table 4.3: New Execution time after PP Optimisation (seconds).

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	4.380	31.071	91.760

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Pre-Processing	0.515	4.311	11.825

Table 4.4: Execution time gain after PP Optimisation (per cent).

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	43.48	47.72	44.99

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Pre-Processing	86.65	85.78	85.78

The improvements are more appreciable considering the percentage of time gained (Table 4.4). The results achieved are substantial: a gain of around 86% for the Pre-Processing sub-section, which results in approximately 45% of the "Seg and FE" macro section.

4.1.2 Feature Extraction Optimisation

For further optimisations, the focus was on the Feature Extraction (FE) section (blocks A and B).

Table 4.5: New Execution time after FE Optimisation (seconds).

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	3.488	25.097	74.419

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Feature Extraction	2.743	19.348	58.011

Feature Extraction	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
A	1.474	10.193	30.861
B	1.014	6.936	21.214

Table 4.6: Execution time gain after FE Optimisation (per cent).

Full Pipeline	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	20.37	19.23	18.9

SEG and FE	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
Feature Extraction	24.04	22.88	22.78

Feature Extraction	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
A	24.53	25.60	25.31
B	26.25	29.52	27.30

Considering the time gain in percentage (Table 4.6), the outcomes obtained are worth mentioning: more than 25% of time saved for each of the FE blocks, and its direct consequence of "SEG and FE" macro section nearly 20% faster, compared with the result achieved with only the first optimisation.

4.1.3 Full Optimisation Outcome

Table 4.7: Execution time gain after full Optimisation (per cent).

Sections Optimized	1 (71.4 s – 143 cycles)	10 (503.1s – 999 cycles)	25 (1540.4 s – 3085 cycles)
SEG and FE	55	57.8	55.4
Pre-Processing	86.4	86.4	86.1
Feature Extraction	24	32.8	24
FE - A	25.2	25.5	25
FE - B	27.4	29.6	27

The full optimisation process led to a significant gain in terms of execution time. Table 4.7 summarise this result in percentage. A total time gain of 55% has been experienced for the "SEG and FE" macro section, saving thus more than half of the time that was necessary without optimisation. The most appreciable gain has been achieved through Pre-Processing optimisation, around 45% of the total gain.

To conclude, the optimised code is capable of processing a single audio file 71 seconds long, related to a single patient, and returning a prediction in less than 4 seconds, while for a total of 25 recordings (more than 25 minutes), it is necessary 1 minute and 15 seconds.

In a real case application, 5 seconds per point, a total of 25 seconds, is a sufficient sound recording length to produce a prediction, reducing, even more, the execution time.

4.2 Edge Device

Execution time on the RPi4 has been evaluated only by processing a single soundtrack, simulating, therefore, a real-time application on a patient. As expected, the execution time reported on this device is considerably stretched. As shown in Table 4.8, approximately 20 seconds are necessary to obtain a prediction of a signal with the RPi4 containing approximately 140 cardiac cycles.

Table 4.8: Full Pipeline Execution Time (seconds) – Raspberry Pi 4.

Full Pipeline	1 (71.4 s – 143 cycles)
SEG and FE	18.543
Sound Classification	0.205
Rearranging And Avg.	0.0514
Final Classification	0.753

SEG and FE	1 (71.4 s – 143 cycles)
Pre-Processing	4.131
Segmentation	1.064
Feature Extraction	13.345

Feature Extraction	1 (71.4 s – 143 cycles)
A	7.168
B	4.897
C	1.271

4.3 Device Principle of Functioning

Figure 4.1 shows the final device implemented during this research, and its principle of functioning designed during this study is explained in this chapter.

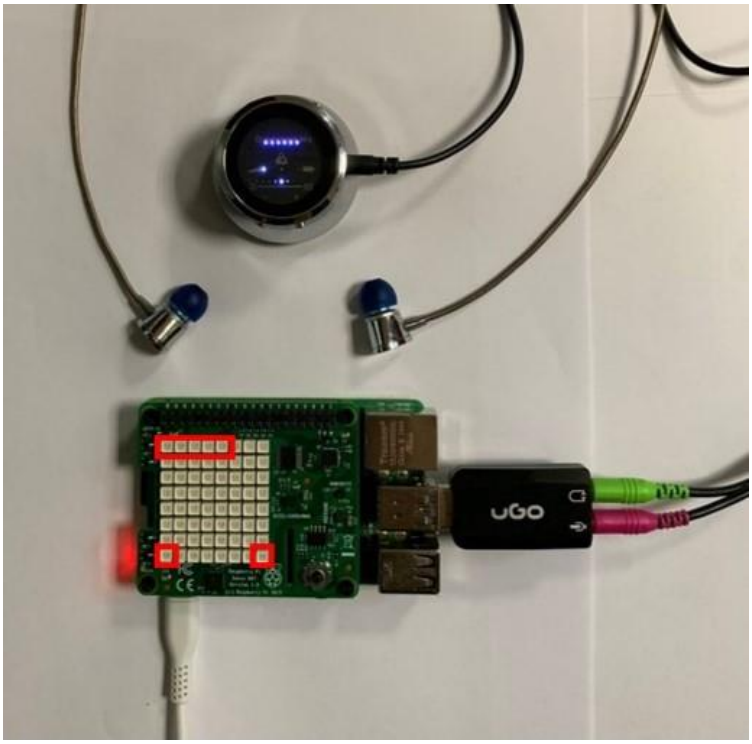

Figure 4.1: Digital stethoscope system.

The five left-upper LEDs of the RPi Sense HAT, highlighted in Figure 4.1, represent each one of the five auscultation points. The LEDs at the left and right bottom corners (Figure 4.1) give information about the prediction made by the device. The operator turns on the device, and the left-upper LED turns red (Figure 4.2.a), meaning that it is ready to record and save from the first auscultation point. At this point, the operator places the digital stethoscope on the first auscultation point and verifies through the headphones that the heartbeat sound is audible and clear. Once the stethoscope is correctly placed, the built-in central button on Sense HAT is pressed for at approximately 5 seconds (Figure 4.2.b): the LED related to the current auscultation point turns white, meaning that the RPi4 is recording the PCG signal received through the jack-port as a soundtrack. When the recording is done, the button is released, the LED turns green (Figure 4.2.c), and the audio file is saved as Point_1.wav.

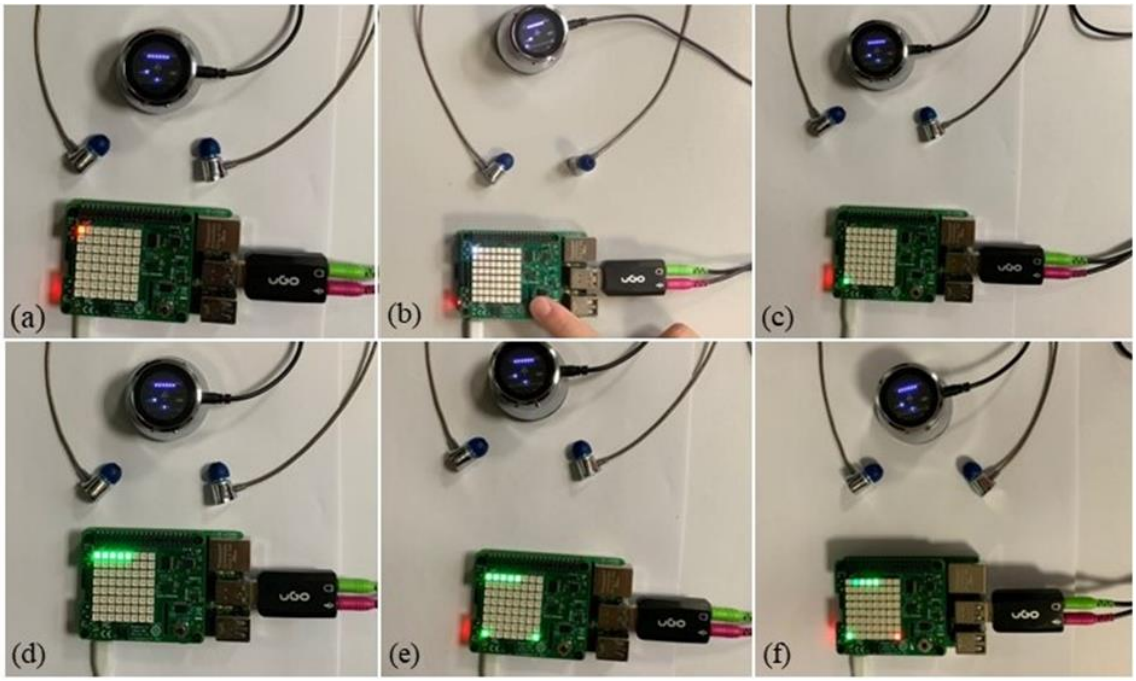

Figure 4.2: Device principle of functioning

As soon as the operator has repeated these three steps for each of the five auscultation points, all five LEDs are green (Figure 4.2.d), and the five audio files are now stored on the RPi4, ready to be processed.

Finally, as shown in Figure 4.2.e-f, the prediction is computed from the algorithm and displayed on the two LEDs at the bottom, which both turn on green if the heartbeat is classified as normal. If the cardiac sound is classified as abnormal with a probability of CHDs' presence greater than 70%, only the right LED turns red, while if PDA is detected (probability of PDA's presence greater than 70%), they both turn red.

Chapter 5

Conclusions and future work

The heart represents one of the major organs for humans, and it is subject to a variety of diseases. Hearth auscultation remains a valid and intuitive non-invasive screening technic for heart defects detection, especially in a resource-constraint context.

Appropriate training and advanced medical knowledge are necessary to interpret neonatal PCG signals correctly. Previous research projects at University College Cork made by the Embedded Systems Team have shown important breakthroughs relying on Machine Learning and aimed at supporting CHDs detection.

The outcome of this research project is indeed an initial concept for a portable device which incorporates and optimises all these innovations conceived by the researchers and capable of delivering a reliable prediction on the possible presence of defections, given an auscultation recording. The final goal is not going without but assisting the clinician in making conscious decisions that could save lives with the help of cheap but accurate equipment.

In the case of examining a standard PCG signal (10s / 20 cycles), the full pipeline ET of the algorithm in the RPi4 is smaller than 3 seconds, making it appropriate for a quick computational response and, thus, for a real-time application at the edge.

This thesis work consists of a first Demo solution of an objective decision support tool. Consequently, the initial concept has been simplified, and further optimisations are required to improve the final device.

The algorithm implemented uses Machine Learning models developed during previous studies. Their performance has been evaluated by means of the Area Under the Curve technique and achieved a degree of accuracy in detecting PDA of the 77% and 78% for CHD detection. The first future development of the

current system could be retraining new models with a bigger availability of data, a result of the automated segmentation followed by the averaging procedure. This process yields a single raw of features per patient by averaging among all the cardiac cycles, providing an improvement in terms of execution time.

Testing the new models would allow stating if the implemented algorithm is leading to information of better accuracy than the original ones and if averaging can be a useful method. Further models based on a restricted set of features could be implemented and analysed to test and compare the obtained results and implement a final algorithm with the best accuracy possible.

Because of the high-power consumption of the Raspberry Pi 4 its use cannot be considered an effective solution for a portable application whose battery capacity is crucial. The optimal solution could be a low-power microcontroller in combination with the conversion of the codes, which are currently written in Python, to C/C++. This would guarantee the implementation of a more efficient device in terms of execution time and memory use beyond that power consumption.

Bibliography

[1] Moorman A, Webb S, Brown N A, Lamers W, & Anderson R H. Development of the heart: (1) formation of the cardiac chambers and arterial trunks. Heart, 2003, 89(7), 806.

[2] Knowles R, Griebsch I, Dezateux C, Brown J, Bull C & Wren C. Newborn screening for congenital heart defects: a systematic review and cost-effectiveness analysis. Health technology assessment, 2005, 9(44).

[3] Zhang Y F, Zeng X L, Zhao E F & Lu H W. Diagnostic Value of Fetal Echocardiography for Congenital Heart Disease: A Systematic Review and Meta-Analysis. Medicine, 2015, 94(42), e1759.

[4] Hanna I R & Silverman M E. A history of cardiac auscultation and some of its contributors. The American Journal of Cardiology, 2002, 90(3), 259–267.

[5] Bishop P J. Evolution of the stethoscope. Journal Of the Royal Society of Medicine, 1980, 73(6), 448–456.

[6] Sakula A. R T H Laënnec 1781--1826 his life and work: a bicentenary appreciation. Thorax, 1981, 36(2), 81–90.

[7] Laennec R & Forbes J. A Treatise on the Diseases of the Chest, and on Mediate Auscultation. 1838.

[8] Bökenkamp R, DeRuiter M C, van Munsteren C & Gittenberger-de Groot A C. Insights into the Pathogenesis and Genetic Background of Patency of the Ductus Arteriosus. Neonatology, 2010, 98(1), 6–17.

[9] Lim M K, Hanretty K, Houston A B, Lilley S & Murtagh E P. Intermittent ductal patency in healthy newborn infants: Demonstration by colour Doppler flow mapping. Archives of Disease in Childhood, 1992, 67, 1217–1218.

[10] Nagasawa H, Hamada C, Wakabayashi M, Nakagawa Y, Nomura S & Kohno Y. Time to spontaneous ductus arteriosus closure in full-term neonates. Open heart, 2016, 3(1), e000413.

[11] Alagarsamy S, Chhabra M, Gudavalli M, Nadroo A M, Sutija V G & Yugrakh D. Comparison of clinical criteria with echocardiographic findings in diagnosing PDA in preterm infants. Journal of Perinatal Medicine, 2005, 33(2), 161–164.

[12] Davis P, Turner Gomes S, Cunningham K, Way C, Roberts R & Schmidt B. Precision and Accuracy of Clinical and Radiological Signs in Premature Infants at Risk of Patent Ductus Arteriosus. Archives of Pediatrics & Adolescent Medicine, 1995, 149(10), 1136–1141.

[13] O'Rourke D, EL-Khuffash A, Moody C, Walsh K & Molloy E. Patent ductus arteriosus evaluation by serial echocardiography in preterm infants. Acta Paediatrica, 2008, 97(5), 574–578.

[14] Pinto N M, Keenan H T, Minich L L, Puchalski M D, Heywood M & Botto L D. Barriers to prenatal detection of congenital heart disease: a population-based study. Ultrasound in Obstetrics & Gynecology, 2012, 40(4), 418–425.

[15] Bakker M K, Bergman J E H, Krikov S, Amar E, Cocchi G, Cragan J, de Walle H E K, Gatt M, Groisman B, Liu S, Nembhard W N, Pierini A, Rissmann A, Chidambarathanu S, Sipek A Jr, Szabova E, Tagliabue G, Tucker D, Mastroiacovo P & Botto L D. Prenatal diagnosis and prevalence of critical congenital heart defects: an international retrospective cohort study. BMJ Open, 2019, 9(7), e028139.

[16] Janicki M B, Fernandez C G, Wakefield D, Shepherd J P, Figueroa R. Improving fetal congenital heart disease screening using a checklist-based approach. Prenatal Diagnosis, 2020, 40(2).

[17] Letourneau K M, Horne D, Soni R N, McDonald K R, Karlicki F C, Fransoo R R. Advancing prenatal detection of congenital heart disease: a novel screening protocol improves early diagnosis of complex congenital heart disease. Journal of Ultrasound in Medicine, 2018, 37(5). 1073-1079.

[18] Watrous R L, Thompson W R & Ackerman S J. The Impact of Computer-assisted Auscultation on Physician Referrals of Asymptomatic Patients with Heart Murmurs. Clinical Cardiology, 2008, 31(2), 79–83.

[19] Gómez-Quintana S, Schwarz CE, Shelevytsky I, Shelevytska V, Semenova O, Factor A, Popovici E & Temko A. A Framework for AI-Assisted Detection of Patent Ductus Arteriosus from Neonatal Phonocardiogram. Healthcare (Basel), 2021, 9(2), 169.

[20] Hussein A F, Arun Kumar N, Burbano-Fernandez M, Ramirez-Gonzalez G, Abdulhay E & de Albuquerque V HC. An Automated Remote Cloud-Based Heart Rate Variability Monitoring System. IEEE Access, 2018, 6, 77055–77064.

[21] Hosseini M P, Soltanian-Zadeh H, Elisevich K, & Pompili D. Cloud-based deep learning of big EEG data for epileptic seizure prediction. IEEE Global Conference on Signal and Information Processing, GlobalSIP, 2016, 1151–1155.

[22] Poveda J, O'Sullivan M, Popovici E, Temko A. Portable neonatal EEG monitoring and sonification on an Android device. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2017

[23] Abbas A K & Bassam R. Phonocardiography Signal Processing. Synthesis Lectures on Biomedical Engineering, 2009, 31, 1–189.

[24] Al-Qazzaz N K, Abdulazez I F & Salma A R. Simulation Recording of an ECG, PCG, and PPG for Feature Extractions. Al-Khwarizmi Engineering Journal, 2014, 10(4), 81–91.

[25] Gómez-Quintana S, Shelevytsky I, Shelevytska V, Popovici E & Temko A. Automatic segmentation for neonatal phonocardiogram. Annu Int Conf IEEE Eng Med Biol Soc, 2021, 135–138.

[26] Fox S & Naughton J P. Physical activity and the prevention of coronary heart disease. Preventive Medicine, 1972, 1(1–2), 92–120.

[27] Gellish R L, Goslin B R, Olson R E, McDonald A, Russi G D & Moudgil V K. Longitudinal modeling of the relationship between age and maximal heart rate. Medicine and science in sports & exercise, 2007, 39(5), 822–829

[28] Fleming S, Thompson M, Stevens R, Heneghan C, Plüddemann A, Maconochie I, Tarassenko L & Mant D. Normal ranges of heart rate and respiratory rate in children from birth to 18 years of age: A systematic review of observational studies. The Lancet, 2011, 377(9770), 1011–1018.

[29] Montague T J, Taylor P G, Stockton R & Smith E R. The spectrum of cardiac rate and rhythm in normal newborns. Pediatric Cardiology, 1982, 2(1), 33–38.

[30] Gomez-Quintana S, Shelevytska V, Caracciolo G, Factor A, Popovici E & Temko A. Hearttone: An accurate, objective decision support tool for detecting congenital heart disease using phonocardiograms and artificial intelligence. European Academic Paediatric Society, 2022.

[31] Tang H, Dai Z, Jiang Y, Li T & Liu C. PCG Classification Using Multidomain Features and SVM Classifier. BioMed Research International, 2018.

[32] Shelevytsky I, Shelevytska V, Golovko V & Semenov B. Segmentation and Parametrisation of the Phonocardiogram for the Heart Conditions Classification in Newborns. Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, 2018, 430–433.

[33] Raschka S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. ArXiv, 2018.

[34] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 1997, 30(7), 1145–1159.

Acknowledgements

First and foremost, I would like to thank my supervisor at Politecnico di Torino, Professor Danilo Demarchi, for his patience and faith in me. His motivation and immense knowledge helped me during the whole journey which ends with the completion of my master's studies. I could have not imagined having a better advisor and mentor.

This project was realised in Embedded.Systems@UCC research group as part of my Erasmus programme. My gratidude goes to each component of this amazing team.

Words cannot express my absolute gratitude to Professor Emanuel Popovici, my Supervisor at University College Cork and head of the Embedded.Systems@UCC research group, but most importantly my point of reference during the beautiful experience that was my Irish Erasmus period. I could not have undertaken this journey without his wide knowledge and kindness, which led me throughout the obstacles and difficulties of the past year, inside and outside the lab walls. I will never forget his support. Thanks Emanuel, for everything.

The completion of this research project could not have been accomplished without the support and guidance of Sergi Gomez Quintana, former PhD student at the Embedded.Systems@UCC research group, to whom I am sincerely grateful.

My gratitude also goes to Professor Andriy Temko of the Embedded.Systems@UCC, first promoter and originator of the project.

Many thanks to Qualcomm for promoting research in UCC and providing continuous support, essential for the achievements accomplished.

I thank my fellow labmates at the UCC: Giulia, Matteo, Isa, Quim. They made my Irish experience unforgettable. It would have not been the same without their support during the research project and their friendship.

Un grazie a tutti i Professori incontrati negli anni a Palermo e Torino, che hanno contribuito alla mia formazione. Un ringraziamento particolare va al Professore Luca Faes, che per primo mi ha trasmesso la passione per la strumentazione biomedica e che è stato il promotore della mia prima esperienza Erasmus, quella che ha cambiato radicalmente la mia vita.

Ringrazio la mia famiglia, che mi è sempre stata accanto in questi anni di studio e che mi ha sempre supportato in tutte le mie scelte, nonostante spesso ci abbiano portato ad essere lontani. Un immenso grazie per i sacrifici fatti affinché io potessi realizzarmi a livello accademico e professionale. Dedico questa laurea a voi, e non smetterò mai di ringraziarvi. Vi voglio bene.

Un immenso grazie agli amici di sempre, che durante questi anni di studio hanno allietato e animato i miei ritorni a casa. Grazie Simone, Gabriele, Marco, Annalisa, Antonino, Giulia, Betta, Elena, Jasmine.

Ringrazio Sofia, che in questi due anni non ha mai smesso di credere in me. Il tuo supporto e la passione per quello che fai mi hanno da sempre motivato durante questo percorso. Il tuo incoraggiamento e la tua fiducia in me sono stati fondamentali nei momenti di grande difficoltà. Grazie per esserci.

Infine, ma non per importanza, voglio ringraziare mio fratello Michele, una costante in tutta la mia vita, accademica ma soprattutto personale. Da sempre consigliere, spalla su cui piangere, motivatore, sostenitore, malgrado la distanza che ci separa ormai da diversi anni. Grazie infinite, non ce l'avrei mai fatta senza di te.