INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Deteção de patologia cardíaca usando machine learning

JESSICA FELIZ DOS SANTOS
Outubro de 2022

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Detection of cardiac pathology using machine learning

Jessica Feliz dos Santos

Dissertation for obtaining the Master's Degree in Computer Engineering, Area of Specialization in Information and Knowledge Systems

Adviser: Elsa Ferreira Gomes

Co-adviser: Jorge Oliveira

Porto, October 2022

Resumo

Segundo a Organização Mundial da Saúde, as doenças cardiovasculares (DCV) representam 32% do número de mortes no mundo. A redução deste valor pode ser atingida através da deteção precoce que pode levar a um tratamento mais preciso, melhorando a expectativa de vida do paciente. A ausculta cardíaca é a principal técnica utilizada pelos profissionais de saúde para identificar muitas DCV. No entanto, a auscultação dos sons cardíacos é um procedimento difícil, já que muitos sons são fracos e difíceis de detetar, sendo necessário um processo de treino contínuo. Os estetoscópios modernos podem amplificar os sons cardíacos, reduzir o ruído de ambiente, melhorar a percepção do usuário e, mais importante, converter um sinal acústico em digital. Isto permitiu o desenvolvimento de sistemas de decisão assistidos por computador baseados na auscultação. Este documento apresenta uma metodologia que pode detectar automaticamente a existência de DCV através de sons cardíacos obtidos de diferentes partes do coração. Diversas tecnologias foram analisadas, assim como projetos que tentam resolver parte do problema em questão e a partir deles, três alternativas diferentes foram elaboradas e documentadas, assim como a divisão do dataset e métricas a serem usadas nos testes. Essas alternativas visam classificar anomalias na auscultação cardíaca dos pacientes. Vários modelos das duas primeiras alternativas foram implementados e seus resultados apresentados. Também é feita uma comparação entre as experiências desenvolvidas entre si, também com experiências básicas que não utilizam mecanismos inteligentes e com outros trabalhos que tenham o mesmo objetivo. O melhor resultado obtido foi pela primeira abordagem com uma exatidão de 94%, precisão de 81% e *recall* de 67%.

Palavras-chave: Doenças cardiovasculares, cuidados de saúde, auscultação cardíaca, redes neuronais, classificação, aprendizagem profunda

Abstract

According to World Health Organization, the cardiovascular diseases (CVD) represent 32% of the number of deaths worldwide. Early detection leads to a more accurate treatment plan and improves the patient's life expectancy. Cardiac auscultation is the main technique used by health professionals to identify many CVD. Nevertheless, heart sound auscultation is a difficult procedure, since it requires continuous training and many heart sounds are faint and hard to detect. However, modern stethoscopes can amplify heart sounds, reduce the environment noise, improve the user's perception and, more importantly, convert an acoustic signal to a digital one. This allowed, the development of computer assisted decision systems based on auscultation. This document presents a methodology that can automatically detect the existence of CVD through cardiac sounds obtained from different parts of the heart. Several technologies were analysed, as well as projects that try to solve part of the problem in question and from them, three different alternatives were elaborated and documented, as well as the division of test data and the metrics for their evaluation. These alternatives are intended to classify anomalies in patients' cardiac auscultation. Several models of the first two alternatives were implemented and their results presented. A comparison is also made between the experiences developed among themselves, also with basic experiments that do not use intelligent mechanisms and with other works that have the same objective. The best result obtained was by the first approach with an accuracy of 94%, precision of 81% and recall of 67%.

Keywords: Cardiovascular diseases, healthcare, heart auscultation, neural networks, classification, deep learning

Acknowledgements

I would like to express my gratitude to my advisers, Elsa Ferreira Gomes and Jorge Oliveira, who guided me throughout this project. The meetings and feedback provided were vital in the development of this dissertation.

Index

List of Figures

List of Tables

Acronyms

Acronyms List

AI	Artificial Intelligence
ANFIS	Fuzzy Inference System
ANN	Artificial Neural Network
AdaGrad	Adaptive Gradient
CNN	Convolutional Neural Network
CVD	Cardiovascular Disease
DL	Deep Learning
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
FAST	Function Analysis System Technique
FFT	Fast Fourier Transform
FT	Fourier Transform
GD	Gradient Descent
HMM	Hidden Markov Model
LSTM	Long-Short Term Memory
MFCC	Mel Frequency Cepstral Coefficients
ML	Machine Learning
NAG	Nesterov Accelerated Gradient
PCG	Phonocardiogram
RNN	Recurrent Neural Network
S1	First Heart Sound
S2	Second Heart Sound
STFT	Short-Time Fourier Transform

SVM	Support Vector Machine
KNN	K-Nearest-Neighbors

1 Introduction

In this section it is performed a general introduction to the work developed. Initially it is presented the context and the problem. Following with the main objectives, solution requirements and the summary of the value analysis of the project. In the end the approach is described as well as the document structure.

1.1 Context

Cardiovascular diseases (CVD) are a group of disorders of the heart and blood vessels that severely increases morbidity and causes lifelong disabilities. These diseases are the leading cause of mortality worldwide for many years now. In a world with 7.8 billion people, the World Health Organization estimates that 17.9 million (32% of all deaths) lives each year are taken by cardiovascular diseases. CVDs include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. More than four out of five CVD deaths are due to heart attacks and strokes, and one third of these deaths occur prematurely in people under 70 years of age (WHO, 2022).

To diagnose these disorders, stethoscopes are used worldwide in the primary health care to check the heart sounds of the patients. Health professionals usually auscultate the heart in four locations of the chest to maximize the detection of heart anomalies, these are called murmurs (Torres, 2021).

Nowadays there are several stethoscopes for different functionalities. Cardiologists use a cardiology stethoscope that gives the professional the ability to hear high and low frequency sounds from the diaphragm, it also has a thick earpiece that cancels unnecessary noise and restricts interference with the auscultations. With the evolution of medical technology, we can now record the sounds and murmurs made by the heart with the help of the phonocardiograph. This machine has a big impact in the prevention of CVD (Pulse Uniform, 2020).

1.2 Problem

CVD diagnosis has improved with the technology growth. We currently have more and better tools to do an accurate prevention of these pathologies, but it is not available to everyone. In under-developed and developing countries the access to healthcare is limited making the heart sound auscultation the chosen diagnostic tool (Oliveira, et al., 2022).

CVD cause health disabilities, which in turn increases the frequency of hospital admissions. This contributes to impoverishment specially in under-developed and developing countries, these expenses affect the economy in the healthcare system and population limiting the resources available. To minimize the impact on these countries and people's lives an early detection is essential to design an effective prevention plan (Oliveira, et al., 2022).

Another issue is the human hearing ability, even a professional with a vast experience with CVD cannot diagnose every single patient accurately. Failing is in the human nature which makes any help in critical subjects like CVD to be appreciated.

1.3 Objectives

The objective of this work is to produce a system capable of classifying heart sounds for screening first-level cardiac pathologies, in hospital and outpatient settings. This system is meant to be a contribute to the existing projects like the ones referred in the state of the art, giving more information about the heart sounds to the healthcare professionals. This additional information will support the professionals when preventing and diagnosing patients.

It is also intended to compare this approach with relevant works identified during the analysis of the state of the art performed.

1.4 Solution Requirements

To address the problems stated above, the following requirements are expected in the conceived system:

Functional Requirements

- Classify patients based in heart sounds of four different auscultation locations in several characteristics. Namely, presence of pathology, murmur quality, murmur shape and others.

Non-Functional Requirements

1. Modifiability - The system is easily modified.

- Reliability - High efficiency of the system after extensive use.
- Performance - Must deliver an output with high level of accuracy and speed.
- Scalability - The system must accept any size of data.
- Manageability - Easy way to change some characteristics of the model used.

1.5 Approach

In this project, the first phase consists in getting knowledge about cardiovascular diseases, the impact they have in people and countries, how they can be diagnosed and the concepts behind it.

After that, the problem is formulated and the next phase begins, which consists of getting knowledge about technologies that can help us solve it in the artificial intelligence area, namely deep learning. It was essential to study the different types of neural networks that are available, their characteristics and components (feature extraction, activation functions, optimizers, and evaluation methodologies). This includes tools to easily build models and test.

To learn what has been done by other authors a state of the art must take place, researching previous projects with similar objectives, analyse and make a critical comparison with the one we are aiming for.

An evaluation methodology is documented, as it will take place once the systems are developed.

A value analysis of the project was also done, it can be consulted in the attachments. This analysis focus in the importance of this study analysing the current status of the CVD prevention, specifically in third-world countries. An Analytic Hierarchy Process (AHP) was also applied to select the deep learning framework.

Next, some design alternatives are documented and analysed, to have options for the development of the system. Some other decisions must be made like the language and tools to use, which model types should be implemented, activation functions to be explored as well as optimizers.

Once the ideas are designed and analysed, implementation takes place being documented, it contains information regarding the models. After that, the evaluation is done and documented as well as the analysis of the results.

1.6 Document Structure

The structure of this document is divided into the following parts: Introduction, Theoretical Concepts, State of the Art, Methodology, Solution Design and Conclusion.

The Introduction is composed by context, the problem and objectives of the work, along with the main solution requirements, value analysis and approach.

The Theoretical Concepts is composed by the heart description, cardiovascular diseases, heart sounds, clinical applications, tools and methods.

The State of the Art is composed by technical concepts of artificial intelligence, neural networks, audio feature extraction, model evaluation methodologies, technology and scientific approaches related to the problem.

The Methodology is composed by the hypothesis, available data, its origin, analysis, the description of the evaluation of the system and the experimental environment.

The Solution Design is composed by framework selection and alternative designs to the problem.

The Implementation is composed by the details of pre-processing of the data, structure and content of the project files, and detailed description of the most significant experiment models.

The Evaluation is composed by the results of the developed models that were detailed in the implementation, comparison between experiments, comparison with some base algorithms and comparison with the literature projects.

The Conclusion and Future Work summarizes the dissertation and describes the possible future work.

2 Theoretical Concepts

In this section it is presented the main theoretical concepts necessary to understand this project. The addressed topics are the constitution of the heart, what are cardiovascular diseases, their burden in the society and the current process for prevention and diagnosis.

2.1 Heart

The human heart (illustrated in Figure 1) is made up of four chambers. Between those chambers there are valves that open when blood passes through them and then close to keep the blood from flowing in the wrong direction. In total there are four heart valves, tricuspid, pulmonary, mitral, and aortic (FPC, 2021).

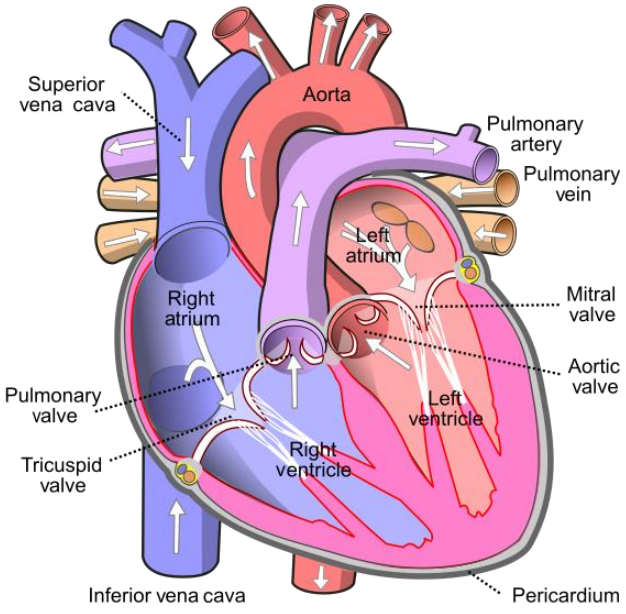

Figure 1 – Human heart (Wapcaplet, 2006)

2.2 Cardiovascular Diseases

The disorders that affect the structure or function of the heart and blood vessels are in a group called cardiovascular diseases (CVD). Each disease has a set of distinct characteristics, like the coronary heart disease, cerebrovascular disease and peripheral arterial, that occur when there is a malfunction on the blood vessels supplying the heart, the brain and the arms and legs respectively (WHO, 2021).

People with CVD are faced with several medical complications that force them to adjust their lifestyle to the disease, like arrhythmias, heart failure, and pulmonary hypertension. This illness is usually accompanied by psychological challenges related to lack of normality, social integration, body image, disclosure, uncertainty, dependence, and coping. Several studies have shown that people with CVD may experience psychological distress associated with feelings of persistent insecurity, depression, anxiety, and low self-esteem (Kim, Johnson, & Sawatzky, 2019).

2.3 Heart Sounds

During the cardiac cycle there are two factors that generate vibrations, the turbulence of the blood flow and the valves that open and close passively because of pressure differences on either side of the valve. These vibrations produce audible sounds, the “lub-dub” that is heard (Oliveira, et al., 2022).

The cardiac cycle has mainly two phases, systole and diastole, each can be associated with a sound. In the systole phase the mitral and tricuspid valves close after the atria have pumped blood into the ventricles generating the first heart sound (S1). In the diastole phase, after the ventricles have ejected the blood from the heart, the aortic and pulmonary valves close producing the second heart sound (S2). S1 and S2 are called the fundamental heart sounds, an illustration of both sounds can be seen in Figure 2 (Had, Sabri, & Aoutoul, 2020).

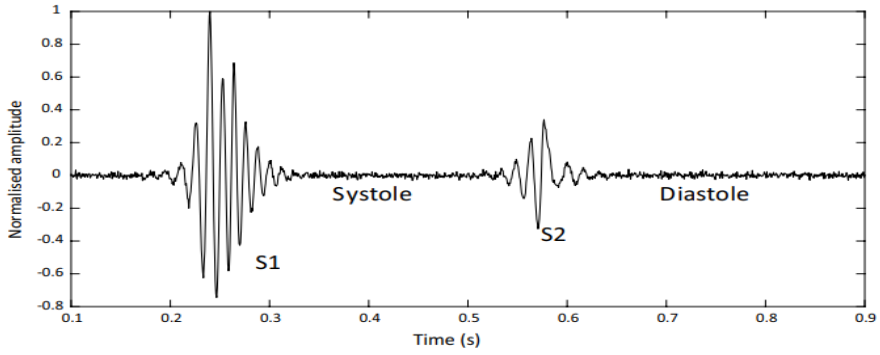

Figure 2 – Normal heart sounds for a single cardiac cycle. Image modified from (Had, Sabri, & Aoutoul, 2020)

2.4 Clinical Applications

When a patient suffers from heart problems, the sound of their heart cycle is different, it presents additional sounds called murmurs. The time between valves closure can also be an indicator of some pathologies.

After this feature of the heart was discovered, a new method of examination emerged, the cardiac auscultation. This is made with a single tool called stethoscope and relies in the human audition. There are mainly four locations of auscultation defined by the best positions to hear the heart valves generated sounds, during the auscultation the stethoscope should be placed at the following positions as illustrated in Figure 3 (Jevon, 2007):

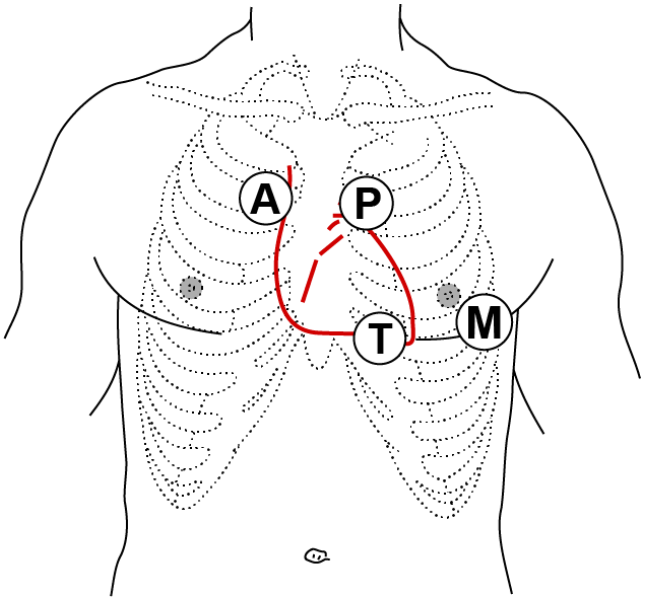

Figure 3 – Aortic(A), Pulmonary(P), Tricuspid(T) and Mitral(M) regions for heart auscultation (Al-Hadithi, 2020)

- Mitral area - left fifth intercostal space, mid-clavicular line.
- Tricuspid area - left fourth intercostal space, just lateral to the sternum.
- Pulmonary area - left second intercostal space, just lateral to the sternum.
- Aortic area - right second intercostal space, just lateral to the sternum.

2.5 Tools and Methods

Cardiac auscultation is the most common method for the detection and prevention of cardiovascular diseases. This procedure is done using a stethoscope and poses no risks or side effects. However, the human audition limitations can make the cardiac examination difficult.

Several approaches and tools have been proposed in this research field to mitigate this problem, like the digital stethoscope and the phonocardiogram (Nall, 2018).

The digital stethoscope allows the health professional to ear the cardiac sound without noise, speeding up the identification of anomalies in the heart. It also has a recording and reproduction function useful for further analysis of the sounds (Norreel, 2021).

The heart sound recorded using a digital stethoscope is called Phonocardiogram (PCG), it converts the acoustic sound waves to electrical signals. This tool is very important for researchers to create innovative methods that extract more information out of the PCG signal with the objective of supporting CVDs prevention and diagnosis (Had, Sabri, & Aoutoul, 2020).

3 State of the Art

In this section it is presented the artificial intelligence (AI) area, in particular the fields of machine learning (ML) and deep learning (DL). An overview of neural networks is also documented, from the available types to their components. The current technologies, tools and similar approaches to solve this type of problem are also studied.

3.1 Artificial Intelligence

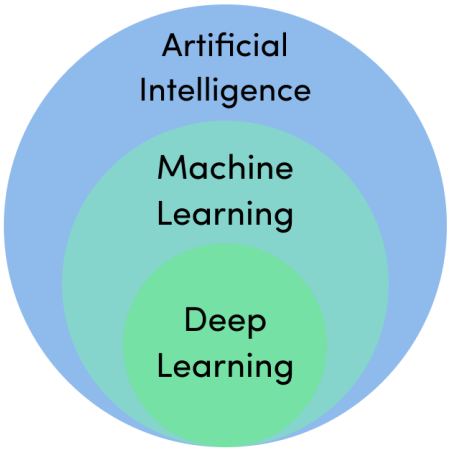

Figure 4 – AI vs ML vs DL (Wolfewicz, 2021)

AI is a field born in 1950s to add features to machines, like i) automation of repetitive learning and discovery through data; ii) intelligence and adaptation through progressive learning algorithms; iii) ability to analyse more and deeper data; iv) improved accuracy and v) extract the most information out of data (SAS Insights, 2022).

A machine with AI has the ability of thinking in a similar way as a human allowing it to do tasks and solve problems that are complex and need more than just coded rules. This includes

learning from experience and adjusting to new inputs. According to (Poole, 1998) an intelligent system is “a system that acts intelligently: What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation”.

In the AI area, input data is called feature and there are two important processes to prepare the data for the AI system, feature engineering and feature extraction. Feature engineering is the process of withdrawing important features from raw data. Feature extraction is the process of extracting features from the data (Mesquita, 2021).

3.1.1 Machine Learning

ML is a subset of AI that allows training systems to perform tasks or solve problems without having explicitly programmed rules. For this to work, the programmer needs to feed the system with pre-processed data, allowing it to learn patterns and be ready to fulfil its objective. This data needs to be previously filtered to contain only the important features to establish patterns (Mesquita, 2021).

As an example, we can look at a system that is meant to solve sudoku puzzles using machine learning, data would be gathered from solved sudoku games and fed to a statistical model. This model is constituted by the ML algorithm and the training dataset (previous inputs and outputs). The ML algorithm predicts a new result based on the training dataset assuming the new input follows the same probability distribution. If, for some reason, the distribution changes, the model needs to be trained with a dataset that follow the new distribution (Mesquita, 2021).

Machine learning can be applied in any business area in a lot of situations, the most important part when starting a new project in this area is to study and analyse the problem as well as the data to know how it can be used by the model. Some examples of machine learning algorithms are (Ray, 2017):

- Support Vector Machine (SVM) that classifies data using filters calculated from training data.
- K-Nearest-Neighbors (KNN) that estimates to which group a data point is likely to be in.
- Decision Tree that uses a tree like structure to classify data.
- Random Forest is a classification algorithm composed of many decisions trees.

3.1.2 Deep learning

DL is a subset of ML inspired in the way a human brain filters information. In this area the system figures out by itself which information is important to establish patterns instead of applying feature engineering techniques. This means that, the feature engineering process can be bypassed (Mesquita, 2021). In Figure 5 the main difference of machine and deep learning is presented.

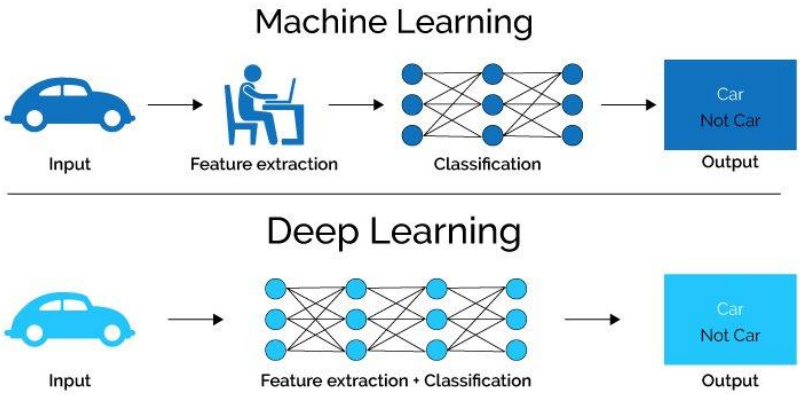

Figure 5 – Comparison between Machine Learning & Deep Learning (Pai, 2020)

Being able to skip the feature engineering gives us advantages in datasets that are complex like images and audios. In this type of data, we don't need to know which features are important, the system will find them. The way of deep learning mimicking human brains though process is by using deep neural networks (DNN). These networks have multiple hidden layers.

DL is usually used to classify objects, for example classify a picture or set of pictures into the name of the animal in them or their characteristic, the number of inputs and outputs can be multiple.

3.2 Neural Networks

When certain application scenarios are too heavy or out of scope for traditional machine learning algorithms to handle, neural networks are the chosen solution (Great Learning Team, 2021).

A neural network is a set of algorithms with the objective of recognizing underlying patterns in a set of data like humans do. The main characteristic is the ability to adapt to changing outputs, exercising a self-training (Chen, 2021).

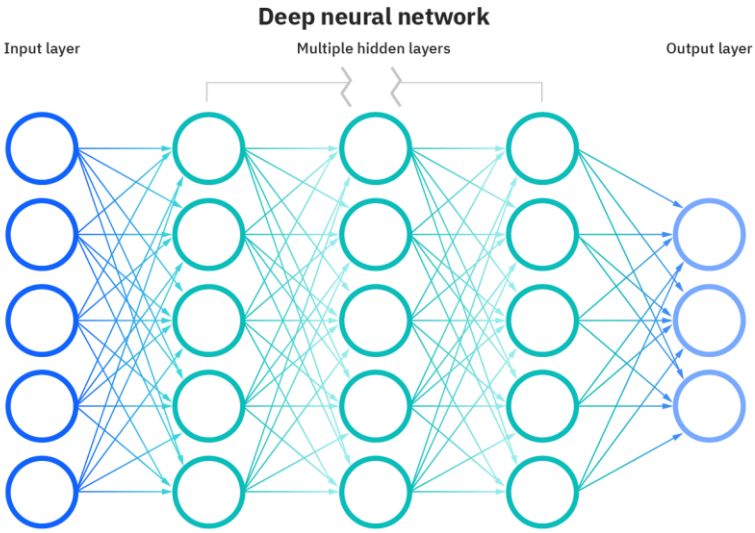

Figure 6 – An example of a neural network (IBM, 2020)

As we can see in the Figure 6, neural networks are constituted by layers, an input layer, one or more hidden layers and an output layer. Each layer has a set of neurons that have connections with another layer. Each connection has an associated weight and bias, this are required to, together with an activation function, calculate which of the neurons of the next layer will be activated. In deactivated neurons no data is passed to the next layer of the network. Neural networks learning process consists in getting input data with known outputs and run them through the network adjusting the weights and bias of the connections.

Neural networks are trained by a set of inputs with already known outputs. The weights in each layer begin with random values, and these are iteratively improved over time to make the network more accurate. A loss function is used to quantify how inaccurate the network is, and an algorithm called backpropagation performs a backward pass while adjusting the model's parameters like weights and biases (Great Learning Team, 2021).

Neural networks learn and improve their accuracy with experience, they rely on training data for that. These learning algorithms when well-trained are powerful tools in computer science and artificial intelligence, allowing us to classify data at a high velocity. The time to perform some tasks can go from several hours to just a few minutes. One of the most well-known neural networks is Google's search algorithm (IBM, 2020).

There are several types of neural networks, but the most used ones are Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) and Convolutional Neural Network (CNN).

3.2.1 Artificial Neural Network

The ANN, also called Feed-Forward Neural Network is a set of algorithms that mimic the human brain and find the relationship between the dataset (Agrawal, 2021). An example is illustrated in Figure 7.

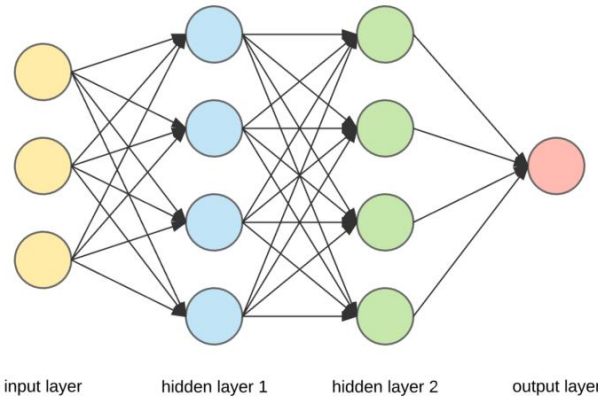

Figure 7 – Artificial Neural Network (Dertat A. , 2017)

Before making predictions, the model needs to be trained to learn the weights, the training process works as follows (Dertat A. , 2017):

- Randomly initialize the weights for all the nodes.
- For every training example:
    - Perform a forward propagation using the current weights and calculate the output of each node going from left to right (the final output is the value of the last node).
    - Compare the final output with the actual target in the training data.
    - Measure the error using a loss function.
    - Adjust the weights (using the learning rate increment or decrement) the backward gradient propagation.

Limitations (Dertat A. , 2017):

- To solve a problem where the data is multidimensional the data needs to be converted in a 1-dimensional vector.
- If working with images, the input vector size increases drastically with bigger images.

- The spatial features are lost, like the arrangement of the pixels in an image.
- Cannot capture sequential information.

To mitigate these limitations other neural networks were created, Recurrent Neural Networks and Convolution Neural Networks.

3.2.2 Recurrent Neural Network

A RNN is an upgraded version of an ANN. It consists of having a connection in each neuron to the same neuron, this extra connection makes the output of that neuron to have the influence, not only of the weight, but also of the previous input (IBM, 2020). In the Figure 8 it is presented the difference between ANNs and RNNs architecture.

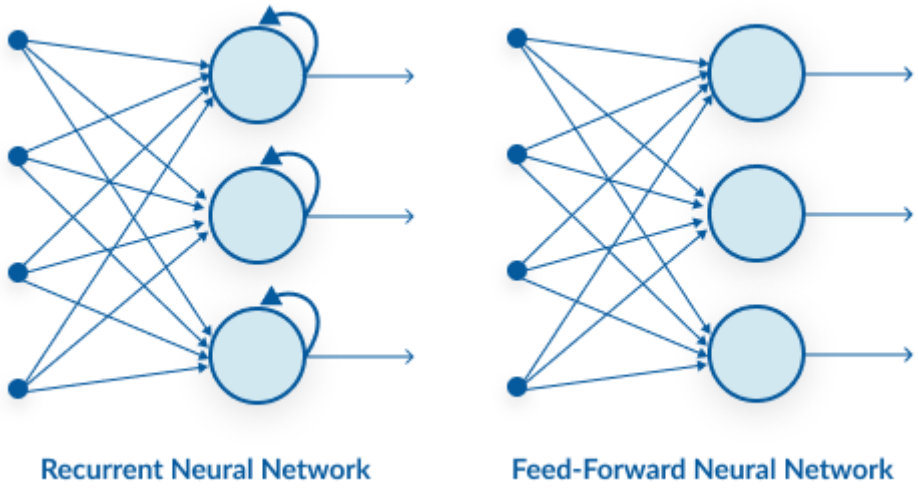

Figure 8 – RNN and ANN architecture (Pai, 2020)

This type of neural network works good for time-series data, text data or audio data, for example in stock market predictions or sales forecasting (IBM, 2020).

3.2.3 Long Short-Term Memory

The LSTM is a special type of RNN in which each neuron contains a memory block for storing the previous inputs. Each memory block consists of three ports: in, out and forget. The neuron decides what to store and when to allow readings, writings or deleting of information. This allows the neuron to be influenced by the previous inputs stored in the memory block (Latif, Usman, Rana, & Qadir, 2018). In Figure 9 a word prediction system is presented as an example of an LSTM.

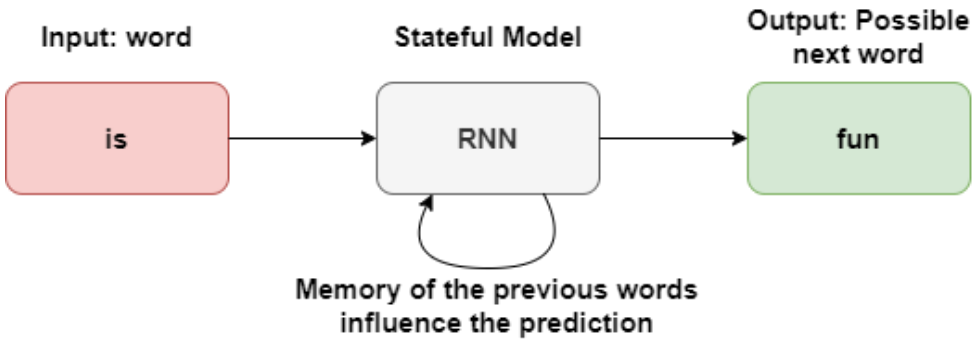

Figure 9 – Example of a word prediction system

3.2.4 Convolution Neural Network

The Convolutional Neural Network is a type of neural network that can take images as input. This neural network assigns importance to the arrangement of the image pixels and automatically detects the important features without any human supervision. It can be thought as a combination of feature extraction and classification (Dertat A. , 2017). In Figure 10 it is presented an example of a CNN.

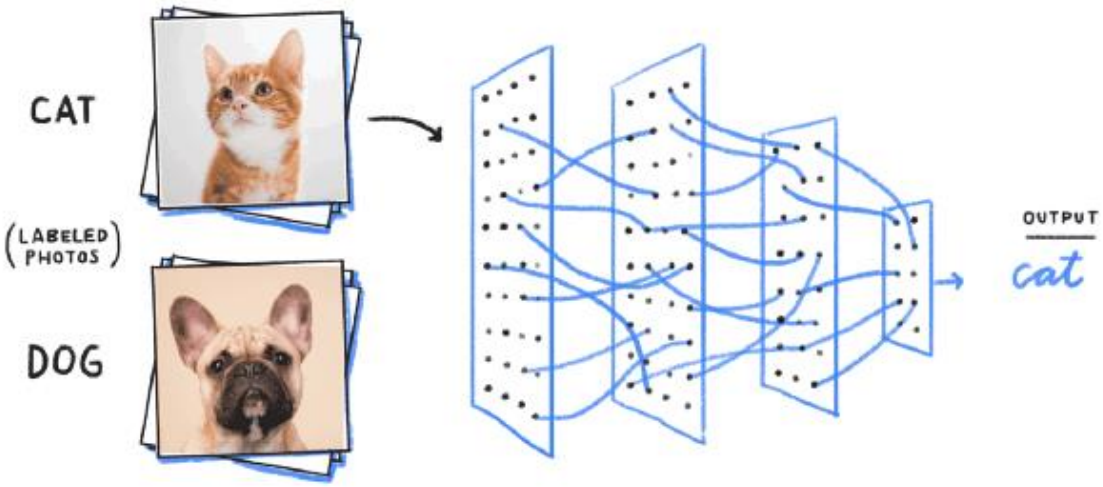

Figure 10 – Example of an image classification (Pai, 2020)

Advantages:

- Learns the filters automatically helping in the extraction of the important features from the input data.
- Captures the spatial features from an image.

3.2.5 Summary

In the Table 1 it is presented the summary of the neural networks studied.

Table 1 – Summary of the neural networks (Gupta, 2020)

	ANN	RNN	CNN
Type of Data	Tabular Data	Sequential Data	Image Data
Fixed Length input	Yes	No	Yes
Recurrent Connections	No	Yes	No
Spatial Relationship	No	No	Yes
Vanishing and Exploding Gradient	Yes	Yes	Yes
Performance	ANN is less powerful than CNN, RNN	RNN includes less feature compatibility when compared to CNN	CNN is more powerful than ANN, RNN
Applications	Facial recognition and Computer vision	Text-to-speech conversions	Facial recognition, text digitization and Natural language processing
Main advantages	Having fault tolerance and ability to work with incomplete knowledge	Remembers each information and time series prediction	High accuracy in image recognition problems and weight sharing
Disadvantages	Hardware dependence, and unexplained behaviour of the network	Gradient vanishing and exploding gradient	Large training data needed and don't encode the position and orientation of object

Analysing this summary, we can see that the different neural networks work better for different types of data, the ANN are used for tabular data, RNN for sequential data and CNN for image data. To choose the neural network to work is highly connected with the context and variables of the problem to solve.

3.3 Audio feature extraction

When working with audio data, the raw format cannot be understood by the models directly, so a conversion needs to be done. Audio feature extraction makes it possible to implement systems of classification, prediction and recommendation for audio data (Doshi S. , 2018). Next are presented the main methods to extract features from audio signals currently used in machine learning.

3.3.1 Fast Fourier Transform

The Fast Fourier transform (FFT) is a method for efficiently compute the Fourier transform (FT) of discrete data samples, such as signals. It significantly reduces the computation time by taking advantage of the fact that the calculation of the coefficients of the DFT can be carried out iteratively (Cochran, et al., 1967).

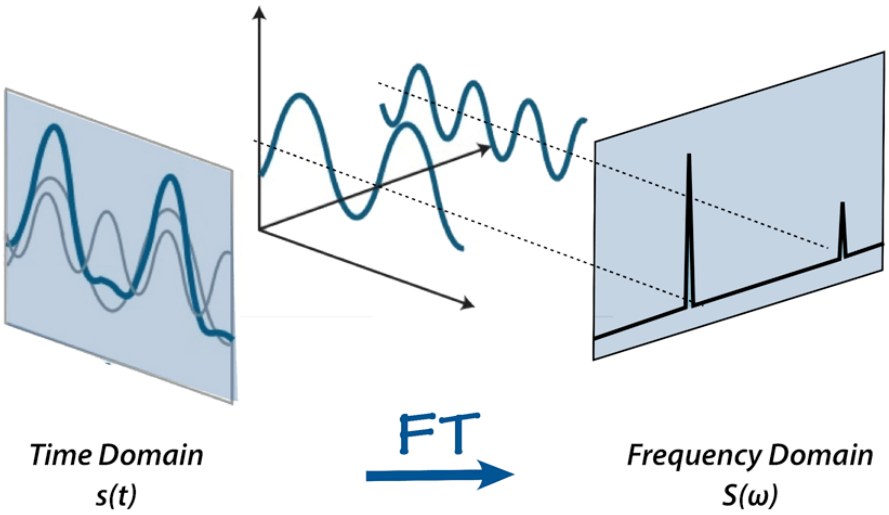

Figure 11 – Application of FT to a signal (Chaudhary, 2020)

The Fourier transform (FT) is a mathematical transformation (illustrated in Figure 11) that allows the conversion of a signal from the time domain into the frequency domain, it is done by decomposing the signal (Roberts, 2020).

The FFT extends the accessible information of the analysed data. One of the advantages is that it doesn't give any information regarding changes in the frequency of the signal over time, it only gives the overall frequency components (Doshi K. , Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation, 2021).

3.3.2 Short-time Fourier Transform

The Short-Time Fourier Transform (STFT) breaks up the audio signal into smaller sections, calculates the FFT for each section and then combines them. Unlike FFT that only provides the frequency information averaged over the time, SFTP captures the variations of the frequency (Doshi K. , Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation, 2021).

3.3.3 Mel Spectrogram

The application of the Fourier transform to a signal generates a spectrogram. It is a visual representation of the amplitude of each frequency present in the signal. Mel spectrograms are modified spectrograms, frequencies are converted to a range that human can ear called the Mel scale and instead of amplitudes, the decibel scale is used which gives colour to the spectrogram. A Mel spectrogram is illustrated in Figure 12.

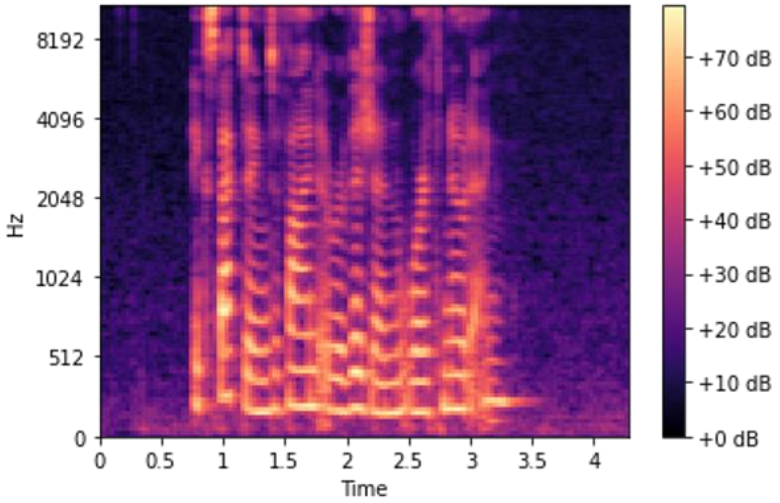

Figure 12 – Mel spectrogram (Doshi K. , Audio Deep Learning Made Simple (Part 2): Why Mel Spectrograms perform better, 2021)

3.3.4 Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients (MFCC) selects a compressed representation of the frequency bands from the Mel Spectrogram that correspond to the human speech frequencies (Singh, 2019).

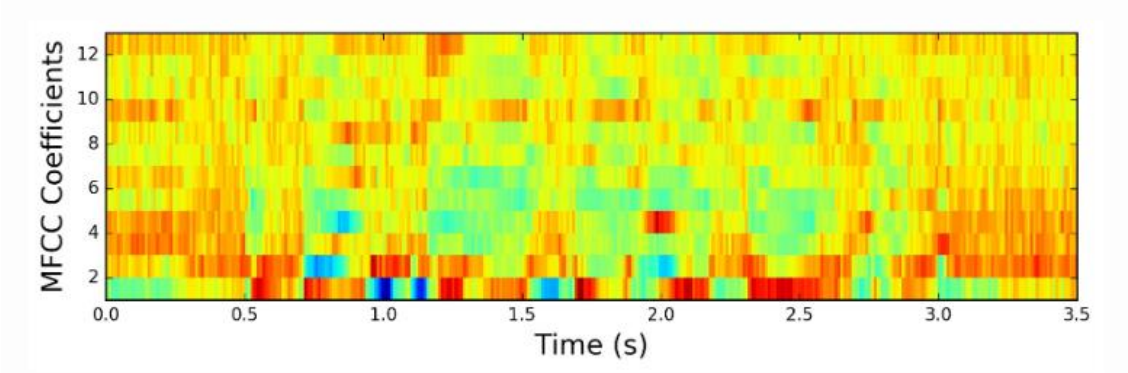

Figure 13 – MFCCs of a signal (Singh, 2019)

The MFCC applies several steps to extract information about the rate changes in the different spectrum bands. In the end we get a compressed representation as illustrated in Figure 13. This feature extraction technique is explained in detail in (Singh, 2019)

3.4 Activation Functions

Activation functions are used to compress the neurons output into a range, their purpose is to add non-linearity to the neural network. Some of the most popular functions (Sharma, 2017):

- Sigmoid is usually for the prediction of probabilities, it ranges between 0 and 1.
- Tanh is mostly used in binary classifications, it ranges between -1 and 1.
- Softmax is used for multi-class classifications.
- ReLU is the most popular for deep learning models, ranges from 0 to infinity.
- Leaky ReLU is a modified ReLu, ranges from -infinity to infinity.

3.5 Optimizers

Optimizers are algorithms used in neural networks to reduce the loss when training the model, they do that by changing the attributes of the networks like weights and learning rate. Some of the most popular optimizers (Kumar, 2020):

- Gradient Descent (GD) is the most basic, it helps the loss function to reach a minimum.
- Nesterov Accelerated Gradient (NAG) is faster than GD to converge.
- Adaptive Gradient (AdaGrad) has an adaptive learning rate.
- Adam is the most used and considered the best as it converges faster than other algorithms.

3.6 Model Evaluation Methodologies

Model Evaluation is an important part of the model development process. It helps to estimate how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate almost perfect (overoptimistic and overfitted) models. To avoid this the Hold-Out and Cross-Validation are two of the methods that can be used to slip the data in distinct subsets. To

evaluate the model performance several metrics can be calculated (Sayad, Model Evaluation, 2022).

The Hold-Out method is usually applied when the dataset is large, it randomly divides the dataset in three subsets (Sayad, Model Evaluation, 2022):

- **Training** set is a subset of the dataset used to train the predictive models.
- **Validation** set is a subset of the dataset used to assess the performance of the model training phase.
- **Test** set is a subset of the dataset to assess the future performance of a model.

For a small amount of data, the Cross-Validation (also known as k-fold cross-validation) method is suggested. In k-fold cross-validation, the data is divided into k subsets of equal size. The model is trained using k-1 of the divided subsets and use the remaining as the test set (Sayad, Model Evaluation, 2022).

3.6.1 Classification Evaluation Metrics

3.6.1.1 Confusion Matrix

A confusion matrix calculates some metrics using the number of correct and incorrect predictions made by the classification model compared to the actual outcomes in the data. The matrix is NxN, where N is the number of classes. The Table 2 is an example of a confusion matrix for a binary classification model (Sayad, Model Evaluation - Classification, 2022).

Table 2 – Confusion Matrix (Sayad, Model Evaluation - Classification, 2022)

Confusion Matrix		Classes			
		Positive	Negative		
Output	Positive	a	b	Positive Predictive Value	a/(a+b)
	Negative	c	d	Negative Predictive Value	d/(c+d)
		Sensitivity	Specificity	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Metrics (Sayad, Model Evaluation - Classification, 2022):

- **Accuracy**: proportion of the total number of predictions that were correct.
- **Positive Predictive Value or Precision**: proportion of positive cases that were correctly identified.
- **Negative Predictive Value**: proportion of negative cases that were correctly identified.

- **Sensitivity or Recall**: proportion of actual positive cases which are correctly identified.
- **Specificity**: proportion of actual negative cases which are correctly identified.

3.6.1.2 ROC Chart

The ROC chart provides a means of comparison between classification models. It used the sensitivity in the y-axis and 1-specificity in the x-axis (Sayad, Model Evaluation - Classification, 2022). This chart is illustrated in the Figure 14, the diagonal red line is for a random model.

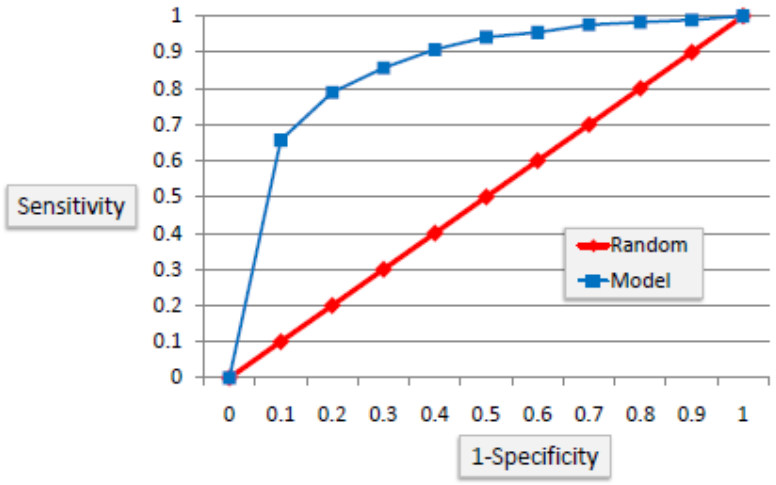

Figure 14 – ROC Chart (Sayad, Model Evaluation - Classification, 2022)

3.6.1.3 F1 Score

The F1 score is the weighted average of precision and recall (Korstanje, 2021):

$$F1\ Score = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

3.6.1.4 Overall Score

The Overall score is the average of sensitivity and specificity (Liu, et al., 2016):

$$Overall = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

3.7 Technology

3.7.1 Python

Python is a programming language easy to learn with efficient high-level data structures and approach to object-oriented programming. It is a scripting language and allows a rapid application development in many areas on most platforms (Python, 2022).

3.7.2 TensorFlow

TensorFlow is an open-source framework developed by Google researchers for developing and deploying Machine Learning applications. It is known for documentation and training support, scalable production and deployment options, multiple abstraction levels, and support for different platforms, such as Android. It supports developers with tools, libraries, and community resources. It provides different levels of abstraction (TensorFlow, 2022).

It is one of the most popular machine learning libraries being for all kinds of projects.

Advantages:

- Compatible with various coding languages such as C, C++, Java, etc.
- Access to both low-level and high-level APIs.
- Fast computational ability across several platforms.
- Fast execution of large datasets.

Disadvantages:

- Not very easy to use.

3.7.3 Keras

Keras is a deep learning high-level API written in Python, that runs on top of the machine learning platform TensorFlow. It is an approachable, highly productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing machine learning solutions with high iteration velocity. It focuses on being modular, user-friendly, and extensible (Keras, 2022).

Key Principles (Keras, 2022):

- Simple: Reduces developer cognitive load to free you to focus on the parts of the problem that really matter.

- Flexible: Adopts the principle of progressive disclosure of complexity, simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what have been learned.
- Powerful: Provides industry-strength performance and scalability, it is used by organizations and companies including NASA, YouTube, or Waymo.

Advantages:

- For Python-based coding.
- Allows fast experimentation with neural networks.
- Very high-level API that could run on CNTK and Theano.
- Popular due to the syntactic simplicity and user-friendly nature.
- Simple architecture

Disadvantages:

- Slower speed than competitors.
- Usually used for small datasets as it is comparatively slower.

3.7.4 Optuna

Optuna is a software framework that allows automatic hyperparameter optimization of models. It has the following advantages:

- Lightweight and versatile handling a wide variety of task with a simple installation.
- Allows the definition of search spaces using familiar Python syntax.
- Efficient optimization algorithms.
- Easy parallelization and scalability.
- Quick visualization of optimization histories.

3.7.5 PyTorch

PyTorch is an open-source machine learning library for Python developed by Facebook's AI research group. It is based on Torch and is used for applications such as natural language processing (Sayantini, 2021).

Advantages:

- Famous for academic research purposes.
- Assists in deep learning applications.
- Fast execution of large datasets.
- Good for natural language processing.
- Used for high-performance models and large datasets that require fast execution.

Disadvantages:

- Only for the Python-based coding.
- Has only low-level APIs that would focus on the working of array expression.
- Complex architecture and the readability are weak.

3.7.6 Summary

Table 3 – Summary of the frameworks (Rogel-Salazar, 2022)

	TensorFow	Keras	Pytorch
API Level	High and Low level	High-level	High-level
Performance	High	Low	High
Architecture	Complex	Simple	Complex
Debugging	Difficult	Moderate	Easy
Dataset Capacity	Large	Small	Large
Popularity	Second most popular	Most popular	Third most popular

In conclusion each framework is good for distinct objectives. Keras is more user-friendly for beginners and allows to a quick start on prototyping ML systems, lacking in performance in large datasets. Pytorch on the other hand is complex and less readable, but it's better for systems that work with big datasets. Tensorflow is a bit in the middle of the previous ones, it allows the user to experiment with high and low API levels, giving it more control of the built systems, but it's complex and not easy to debug. All the frameworks have currently a good supply of online resources.

3.8 Scientific approaches related to the problem

In this section there are presented several articles of interest. First, we analyse two articles, one related to the classification of murmur specific location to aid in the diagnosis and the other that explores a multi-classification of the location of the heart sounds before of the

murmur presence classification. Next other articles are summarized including papers that resulted from a previous PhysioNet (PhysioNet, 2022) challenge. This challenge is relevant due to the topic addressed, its objective was to develop algorithms to classify heart murmurs as normal or abnormal.

3.8.1 Extraction and assessment of diagnosis-relevant features for heart murmur classification

In the (Levin, Ragazzi, Szot, & Ning, 2021) article it is documented an approach for heart murmur detection and multi-class classification. The main objective was to classify the heart sounds into the following seven categories:

- Early systolic murmurs.
- Mid systolic murmurs.
- Late systolic murmurs.
- Holosystolic murmurs.
- Diastolic murmurs.
- Systolic and diastolic murmurs.
- Normal heart sounds without murmurs.

For this the extracted features were from the time and frequency domains, combined with the next 16 additional calculated features:

1. Ratio of average systole amplitude to average S1 amplitude;
2. Ratio of average diastole amplitude to average S1 amplitude;
3. Theorized presence of systolic murmur, determined by whether Feature 1 crosses an empirically determined threshold;
4. Theorized presence of diastolic murmur, determined by whether Feature 2 crosses an empirically determined threshold;
5. Sum of the absolute values of the derivatives of every point in the systole silhouette;
6. Sum of the absolute values of the derivatives of every point in the diastole silhouette;
7. Number of peaks in the systole;
8. Number of peaks in the diastole;

9. Variance in the time between peaks within the systole;
10. Variance in the time between peaks within the diastole;
11. Number of peaks in systole below a threshold 1a, which is 10% of the amplitude of S1;
12. Number of peaks in systole above threshold 2a, which is 40% of the amplitude of S1;
13. Number of peaks in systole in between threshold 1a and threshold a2;
14. Number of peaks in diastole below threshold 1b, which is 10% of the amplitude of S2;
15. Number of peaks in diastole above threshold 2b, which is 40% of the amplitude of S2;
16. Number of peaks in diastole in between threshold 1b and threshold 2b;

These features were fed to supervised machine learning with KNN and SVM algorithms. Which had an accuracy, average precision, and average sensitivity of 91%, 86% and 86% for KNN and 94%, 91% and 92% for SVM.

3.8.2 Multi-label classification of heart sound signals

The (Zhiming & Sheng, 2021) article explores the problem of having mixed heart sound locations in a collection area. This study has two main objectives, to classify the location of the auscultation recordings and after classifying that auscultation in normal or abnormal heart sound. The extracted features used are the MFCC and Power Spectral Density, these were fed to an SVM, Random Forest and Back-Propagation Neural Network. The results for the auscultation location classification and murmur presence classification are shown in the Table 4 and Table 5 respectively.

Table 4 – Results of the auscultation location classification

Model	F1 Score	Accuracy	Recall
Random Forest	79.32%	86.78%	89.34%
SVM	81.65%	91.03%	91.56%
BP Neural Network	75.48%	82.08%	77.06%

Table 5 – Results for the SVM murmur classification

Location	F1 Score	Accuracy	Recall
A	94.04%	92.92%	91.56%
E	81.45%	80.65%	85.55%
M	92.14%	90.55%	85.54%
P	80.45%	86.55%	81.02%
T	89.99%	89.65%	88.45%

3.8.3 Other Papers

In this section it is presented other articles related to the problem. These studies summarized in Table 6 are all for the classification of a heart sound as normal or abnormal, taking as the input a sound recording file. The difference between them is in the models and features used as well as the metric results.

Table 6 – Other studies details

Paper	Methods	Features	Metrics
(Yaseen, Son, & Kwon, 2018)	SVM, DNN and Centroid Displacement based KNN	MFCC and Discrete Wavelet Transform (DWT)	Accuracy 92.1% F1 Score 98.3% Recall 94.5% Specificity 98.2%
(Wang, et al., 2020)	ANN	DWT and Shannon Energy	Accuracy 93% Recall 93.5% Specificity 91.7%
(Poornima & Savithaa, 2021)	Threshold	Shannon Energy, Spectral Width and Pitch Frequency	Normal heart sound efficiency 90.47% Murmur efficiency 87.53%
(Ahmad, Mir, Ullah, Shahid, & Syed, 2019)	SVM and KNN	MFCC	Accuracy 92.6%
(Demir, Şengür, Bajaj, & Polat, 2019)	AlexNet, VGG16, and VGG19 (pre-trained CNN), SVM	STFT	Normal heart sound precision 59% Murmur precision 77%
(Fahad, Khan, Saba, Rehman, & Iqbal, 2017)	Adaptive-Neuro Fuzzy Inference System (ANFIS) and Hidden Markov Model (HMM)	Shannon Entropy, Energy, Zero–crossing Rate, Spectral Entropy, Frequency and Spectral Centroid	Normal heart sound accuracy 98.7% Murmur accuracy 100% Recall 100% Specificity 99.3%
(Thompson, Reinisch, Unterberger, & Schriefl, 2018)	“non-linear artificial intelligence”	Unknown	Accuracy 88% Recall 93% Specificity 81%

3.8.4 PhysioNet 2016 Challenge Papers

Several papers from the PhysioNet 2016 Challenge (Liu, et al., 2016) where also analysed. These studies summarized in Table 7 are the result of the challenge and they have the same objective of the ones on the previous section, the classification of a heart sound as normal or abnormal, having a sound recording file as input.

Table 7 – Challenge studies details

Paper	Methods	Features	Metrics
(Grzegorczyk, et al., 2016)	Threshold and ANN	Time and Frequency domains	Specificity 76% Recall 81% Overall Score 79%
(Homsi, et al., 2016)	Cost-Sensitive Classifier, LogitBoost and Random Forest.	Time, Frequency, Wavelet and Statistical domains	Recall 81.2% Specificity 85.2% Overall Score 84.48%
(Langley & Murray, 2016)	Threshold	Wavelet entropy	Recall 98% Specificity 56% Overall Score 77%
(Goda & Hajas, 2016)	SVM	Average width of S1 and S2, DFT and Wavelet Transform	Recall 77.2% Specificity 85.2% Overall Score 81.2%
(Nilanon, Yao, Hao, Purushotham, & Liu, 2016)	Logistic regression, SVM, and Random Forest	Spectrogram and MFCC	Recall 77% Specificity 85% Overall Score 81%
(Ortiz, Phoo, & Wiens, 2016)	SVM	Time Interval, MFCC, Dynamic time warping distances	Overall Score 82.4%
(Potes, Parvaneh, Rahman, & Conroy, 2016)	AdaBoost and CNN	PCG intervals and amplitudes, DFT and MFCC	Recall 94.2% Specificity 77.8% Overall Score 86.0%
(Rubin, et al., 2016)	CNN	MFCC	Recall 76.5% Specificity 93.1% Overall Score 84.8%
(Vernekar, Nair, Vijaysenan, & Ranjan, 2016)	ANN, SVM, Random Forest	FFT, Autoregressive Moving-Average, MFCC, Wavelet entropy, Music features, Octave band features and Markov	Recall 71.6% Specificity 82.7% Overall Score 77.2%
(Zabihi, Rad, Kiranyaz, Gabbouj, & Katsaggelos, 2016)	ANN and DNN	Linear Predictive Coefficient, Natural and Tsallis entropy, MFCC, DWT and Power Spectral Density	Recall 86.91% Specificity 84.90% Overall Score 85.90%

(Singh-Miller & Singh-Miller, 2016)	Random Forest	Mean and Variance of spectral features	Recall 76% Specificity 87% Overall Score 81%
(Tschannen, Kramer, Marti, Heinzmann, & Wiatowski, 2016)	CNN and SVM	Amplitudes and Durations of periods, Power Spectral Density	Recall 84.8% Specificity 77.6% Overall Score 81.2%

3.8.5 Conclusion

Analysing the articles, we see that all the developed approaches reached good results compared to their objectives. The presented system designs had similar steps, pre-processing of the signals, segmentation, extraction of the features, creation of the models, classification of the sounds and at the end, the evaluation of the system.

Looking at the several articles we see a great variety in the models used, from Logistic regression, Random Forest classifiers, SVMs to different types of neural networks. In the feature extraction there is mainly time and frequency domain features. For the evaluation methodology of the models, the K-fold cross-validation with the accuracy, sensitivity and specificity metrics is the most common method, although the precision, overall score and F1 score metrics are also found among the articles analysed.

4 Methodology

In this section it is presented the hypothesis of this project, the origin of the test data, its analysis and how the system will be evaluated.

4.1 Hypothesis

The hypothesis of this project is that the system to be implemented will help professionals in the area of cardiovascular diseases by providing support and providing detailed information about possible murmurs in the patient's heart, giving information about the presence, timing, shape, grading, pitch, and quality of the murmur.

4.2 Available Dataset

4.2.1 Origin

The dataset that will be used in this dissertation comes from the George B. Moody PhysioNet Challenge 2022 (Reyna, et al., 2022). According to (Oliveira, et al., 2022) the data was collected as part of two mass screening campaigns, conducted in Paraíba state, Brazil in 2014 (CC2014) and in 2015 (CC2015).

The data available is only 60% of the gathered data, this is because the rest 40% will be used in the challenge for testing purposes. After the campaigns the data was segmented using three algorithms to identify the S1 and S2 sounds and their boundaries. The recordings were also analysed by experts for the presence of murmurs and their characteristics, being labelled accordingly. The available dataset is composed by 3163 recordings from 942 patients (Oliveira, et al., 2022).

4.2.2 Data Files

The dataset is composed by four different types of files (Reyna, et al., 2022). The Table 8 summarizes these files.

Table 8 – Available files details (Reyna, et al., 2022)

File	Format	Content	Number of Files	Name example
Audio	.wav	Auscultation recording data	1 file per auscultation location per subject	12345_MV.wav
Header	.hea	Description of the .wav file in the standard Waveform Database format	1 file per auscultation location per subject	12345_MV.hea
Segmentation	.tsv	Segmentation information regarding the start and end points of S1 and S2	1 file per auscultation location for all subjects	12345_MV.tsv
Subject	.txt	Demographic data such as age, sex, height, weight, and pregnancy status as well as a detailed description of any murmur events	1 file per subject	12345.txt

In the name of these files the subject is represented by its ID and the auscultation location by its code (PV for pulmonary valve, TV for tricuspid valve, AV for aortic valve, MV for mitral valve or Phc for any other auscultation location) (Reyna, et al., 2022).

4.2.3 Data Variables

Some of the variables available in the dataset subject file are annotations about the presence, location, most audible location, timing, shape, pitch, quality and grade of the murmurs. These characteristics are the categories for a classification system, Table 9 presents them. The timing, shape, pitch, quality and grade are annotation for both systolic and diastole periods.

Table 9 – Information details (Reyna, et al., 2022)

Tag Name	Description	Possible Values
Murmur	Presence of the murmur	**Present**: murmur were detected in at least one recording **Absent**: murmur were not detected in any recording **Unknown**: the presence or absence of murmurs is unclear
Murmur's locations	Auscultation location(s) where at least one	PV, TV, AV, MV, Phc

	murmur wave has been observed.	
Most audible location	Auscultation location where murmur waves were most audible	PV, TV, AV, MV, Phc
Murmur timing	Timing of the murmur wave	Early-systolic, Mid-systolic, Late-systolic, Early-diastole, Mid-diastole, Late-diastole, Holosystolic
Murmur shape	Shape of the murmur wave	Crescendo, Decrescendo, Diamond, Plateau
Murmur pitch	Pressure gradient felt in the heart chambers	High, Medium, Low
Murmur grading	Murmur's intensity grade according to the Levine scale (Kazemnejad, Gordany, & Sameni, 2021)	**Grade I/VI**: if barely audible and not heard in all auscultation locations **Grade II/VI**: if soft, but easily heard in all auscultation locations **Grade III/VI**: if moderately loud or loud. Grade III/VI denotes all grade III/VI and above (IV/VI, V/VI, and VI/VI)
Murmur quality	Murmur's quality feature from waves	Blowing, Harsh, Musical

4.2.4 Analysis

The analysis of the data was done and documented in (Oliveira, et al., 2022). The heart sound signals were collected using a Littmann 3200 stethoscope which had the DigiScope Collector technology embedded. The recordings were sampled at 4KHz with an average duration of 28.7 seconds and 19.0 seconds, in CC2014 and CC2015 respectively. The collected dataset includes a total number of 215780 heart sounds, 103853 heart sounds from CC2014 and 111927 from the CC2015. This sounds are summarized in Table 10.

Table 10 – Locations of the recordings

	CC2014	CC2015	Total
Aortic point	540	817	1357
Pulmonary point	497	793	1290
Mitral point	603	812	1415
Tricuspid point	461	754	1215
Unreported point	5	1	6

The collected auscultation recordings came from 1568 participants, 50.2% male and 49.8% female. There are samples from several age categories like children (63.0%), infants (19.80%), adolescents (8.1%), young adults (0.6%), neonates (0.7%) and patients without age data (0.8). There are also 8.1% of pregnant women. The same occurs regarding ethnicity varying from mixed race (82.8%), white (15.9%), and other ethnic backgrounds (1.4%) of other ethnic

backgrounds. Table 11 summarizes the population demography. This dataset appears to be a good representation of the demography in Northeast Brazil, Paraíba.

Table 11 – Gender, Age Group, Child's Race, Mother's Race Distribution from the CC2014 and CC2015 screening campaigns

		CC2014 (%)	CC2015 (%)	Total (%)
Gender	Male	325 (49.8)	462 (50.5)	787 (50.2%)
	Female	328 (50.2)	453 (49.5)	781 (49.8%)
Age Group	Child	405 (62.0)	583 (63.7)	988 (63.0)
	Infant	126 (19.3)	185 (20.2)	311 (19.8)
	Pregnant	57 (8.7)	53 (5.8)	110 (7.0)
	Adolescent	51 (7.8)	76 (8.3)	127 (8.1)
	Young adult	5 (0.8)	4 (0.4)	9 (0.6)
	Neonate	3 (0.5)	8 (0.9)	11 (0.7)
	No info	6 (0.9)	6 (0.7)	12 (0.8)
Race (child)	Mixed Race	492 (75.3)	806 (88.1)	1298 (82.8)
	White	151 (23.1)	98 (10.7)	249 (15.9)
	Black	9 (1.4)	11 (1.2)	20 (1.3)
	Asian	1 (0.2)	0 (0.0)	1 (0.1)
Race (mother)	Mixed Race	389 (59.6)	705 (77.0)	1094 (69.8)
	White	240 (36.8)	195 (21.3)	435 (27.7)
	Black	24 (3.7)	14 (1.5)	38 (2.4)
	Asian	0 (0.0)	1 (0.1)	1 (0.1)

Table 12 – Age Statistics of the Participants in Months

	CC2014	CC2015	Total
Mean	74.7	72.5	73.4
Median	78.4	70	72.1
Standard deviation	50.4	50.3	50.3
Minimum	0.1	0.1	0.1
Maximum	217.8	356.1	356.1

The mean age (± standard deviation) of the participants is 73.4 ± 0.1 months, ranging from 0.1 to 356.1 months, as presented in

Table 12.

4.3 Evaluation of the systems

The evaluation of the developed systems is very important to understand how well it achieves its goals. This can be done by using an evaluation methodology for neural networks and retrieving metrics that reflect the effectiveness and efficiency of the system.

The evaluation methodology chosen is the cross validation because of the small dataset available (with only 942 patients), this dataset is going to be separated in three subsets (train, validation and test). The metrics presented in Table 13 are going to be used to measure the alternatives developed and compare them with each other and with other studies made in the field. In this project the class considered as positive is the “Present” class.

Table 13 – Evaluation metrics

Metric	Comparison between alternatives	Comparison with other studies
Accuracy	X	X
Precision	X	X
Recall	X	X
F1 Score	X	X
Computation Time	X	

Regarding the comparison with other studies, we can only compare the classification of normal or abnormal heart sounds except for the (Levin, Ragazzi, Szot, & Ning, 2021) study that implemented a multi classification system. The metrics to be compared are the ones available in each study.

When comparing the different alternatives, we can look at other metrics like the computation time for training the models, as this phase will be computed by the same environment. If needed other metrics will be added for the evaluation of the systems.

4.4 Experimental Environment

CNN are going to be used in the current study due to their time invariance properties. Furthermore, according to "Do we really need a segmentation step in heart sound classification algorithms?", no segmentation step is needed when using CNNs.

The CNN is going to have as input the Mel spectrograms of the auscultation recordings, as this type of neural network is the most appropriate for working with images. Other reason is that we want the system to consider the spatial structure of the data.

Regarding to the activation functions and optimizer, it was decided to choose the most referred on literature as these components are not the focus of this study. As activation functions we are going to use ReLu in the hidden layers and softmax in the last layer. For the optimizer it was decided to use Adam, as it is computationally efficient and has little memory requirements (Brownlee, 2017).

Regarding the development environment, it was decided to use Google Colab considering the following advantages:

- Is free to use.
- Doesn’t require any installation.
- The sharing system makes easy to share the work done.
- Integration with Github makes versioning easier.
- Compatibility with Jupyter Notebook allowing a better organization of the code.
- Runs in the cloud, in other words, free computation resources available.
- Code executes in the Google servers providing high performance.

Due to computation limitations, it was decided to upgrade to Paperspace. This environment offers the same advantages with the difference of the code executing in better servers.

5 Solution Design

In this section it is presented the architecture of the solution, the selection of the language and deep learning framework to be used and the analysis of three alternative designs. The proposed architectures address a very challenging and innovative task in which recordings extracted from various auscultation points can lead to new architectures.

5.1 Language and Deep Learning Framework

The chosen language for this project was python (Python, 2022) considering the available libraries to work with deep learning, audio files and high-level data structures. In section 3.7.1 there are more information about this language and its advantages.

Regarding the deep learning framework to be used, based in the analysis made in the state of the art and the Analytic Hierarchy Process (AHP), located in the attachments, applied to the three most popular frameworks, we chose to use Keras (Keras, 2022) in combination with TensorFlow (TensorFlow, 2022). It provides the project with tools to do fast experimentation in the systems and change their parameters to best fit the project needs.

5.2 Architecture Alternatives

Design alternatives need to consider some main characteristics of the desired solution: it must receive up to four audio files per patient as input and can classify them in the attributes presented in Table 14.

Table 14 – Patient classifications

General Classifications	Systolic Classifications	Diastole Classifications
Murmurs Presence	Murmur Systolic Timing	Murmur Diastole Timing
Murmurs Location	Murmur Systolic Shape.	Murmur Diastole Shape.

Most Audible Location	Murmur Systolic Grading. Murmur Systolic Pitch. Murmur Systolic Quality.	Murmur Diastole Grading. Murmur Diastole Pitch. Murmur Diastole Quality.

To study how the number of input auscultation locations and the new available murmur characteristics can affect the decision system, several alternatives of the system architecture were designed and analysed.

5.2.1 Alternative 1

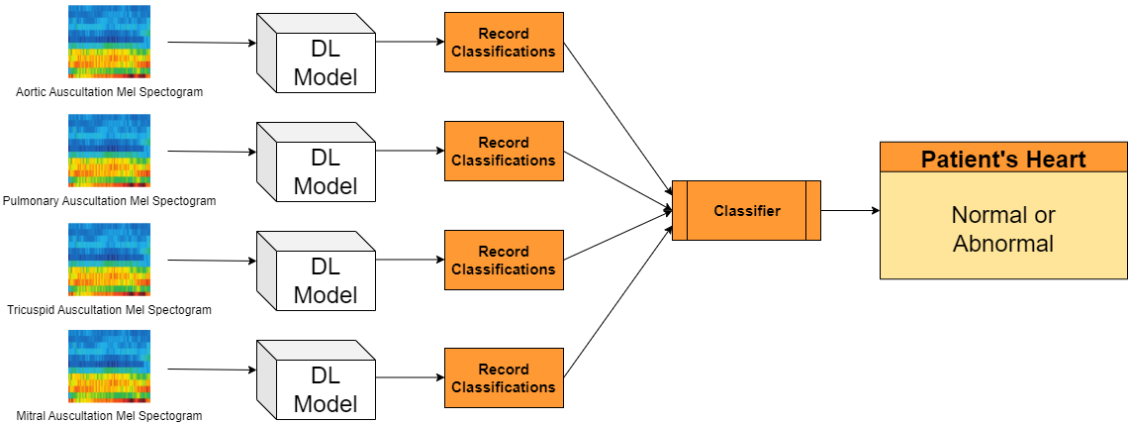

Figure 15 – System architecture alternative 1

In this design there is a deep learning model for each of the four auscultation locations. This design is based in the traditional approach. Each DL Model would try to predict the presence or absence of murmurs in their corresponding auscultation spot. At the end of the pipeline, another classifier is going to merge and combine the prediction of each single classifier, and output a patient's heart state, in another words, the last classifier is going to infer if the patient's heart is normal, abnormal or unknown.

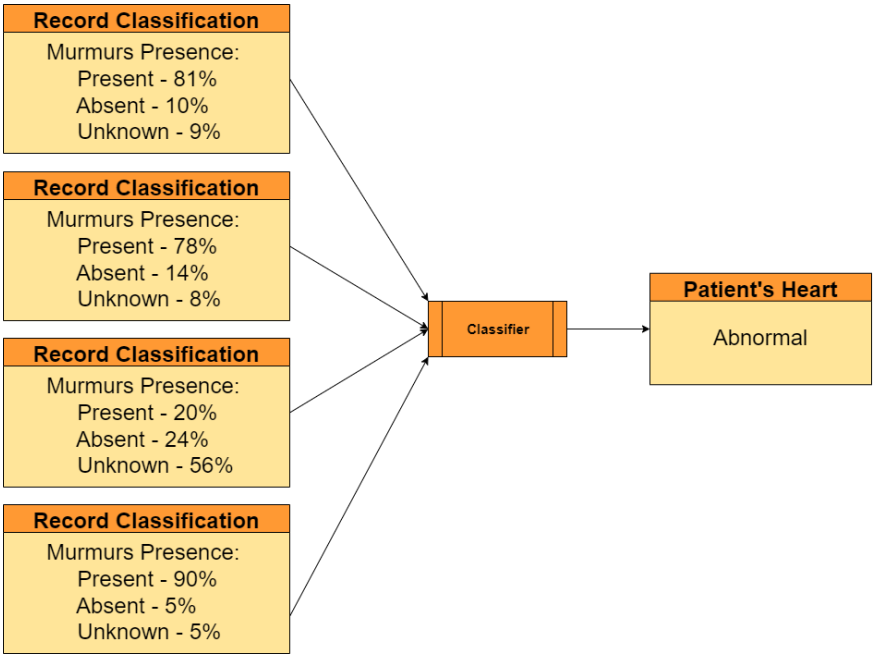

Figure 16 – Final prediction process example

When the audio files are processed the classifications would need to be joined, this process needs a strategy. The usual strategy used in the literature is that if any of the predictions points to the presence of heart murmurs, then the system infers that the patient is abnormal and complementary exams should be made. To simplify the system, in this design it is considered the prediction with the higher probability among the four predictions as illustrated in Figure 16.

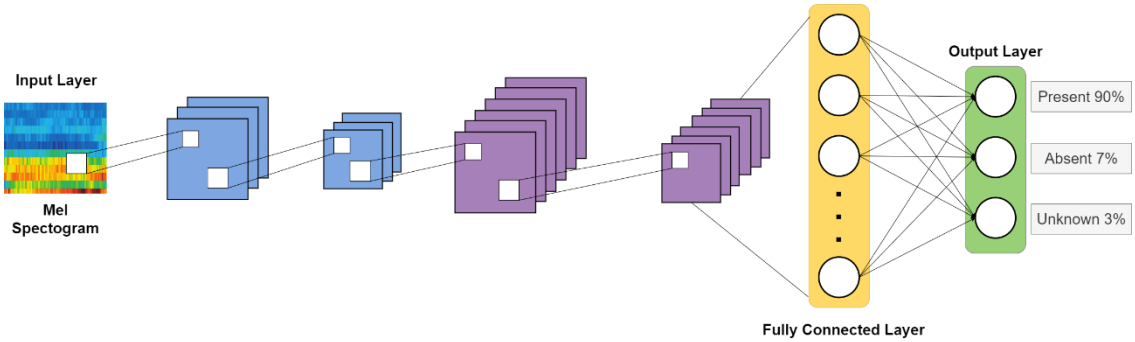

Figure 17 – Architecture of the alternative 1 DL Model

The DL model (presented in Figure 17) will receive a single Mel spectrogram, generated from one of the auscultation points recording. The neural network is composed by an input layer which receives the image, a fully connected layer and an output layer with 3 neurons. Each neuron represents a possible outcome of the variable murmur presence, and its output will be the probability for each class. For example, we could have as output 90% Present, 7% Absent and 3% Unknown.

Advantages:

- Each DL model would focus on a single auscultation location making it simpler to implement.

Disadvantages/Limitations:

- Time complexity due to having four DL models running at the same time.
- The DL models don't consider the other audio files which may lead to wrong predictions.

5.2.2 Alternative 2

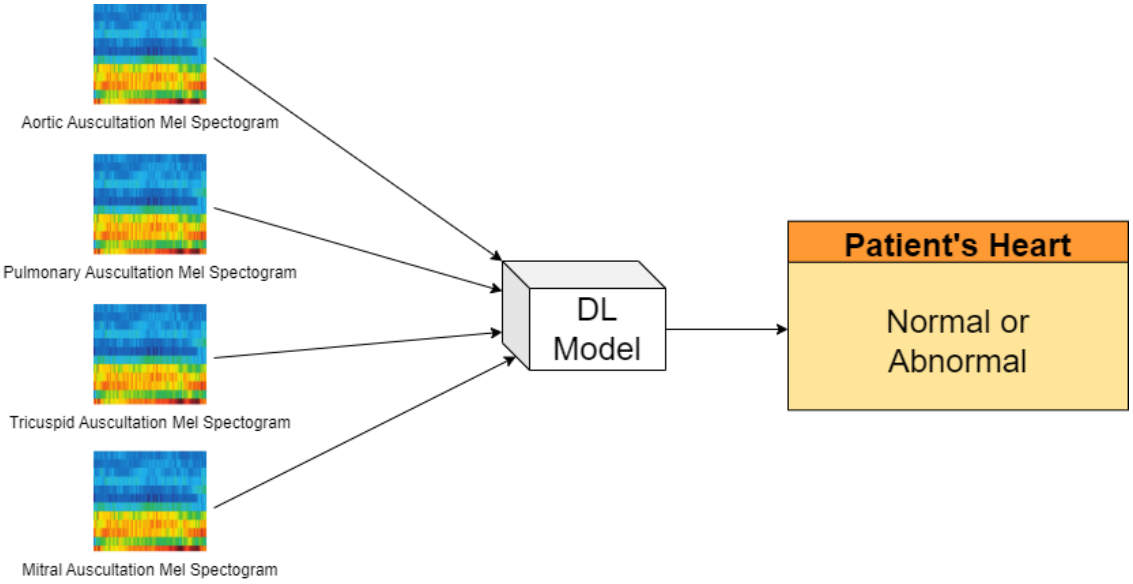

Figure 18 – System architecture alternative 2

Another design alternative is to have a single DL model receiving all the auscultation Mel spectrograms and classifying the patient's heart. The major difference to the previous alternative is that the DL system is a multi-input CNN and considers all the auscultation spots for the classification as it is presented in Figure 18.

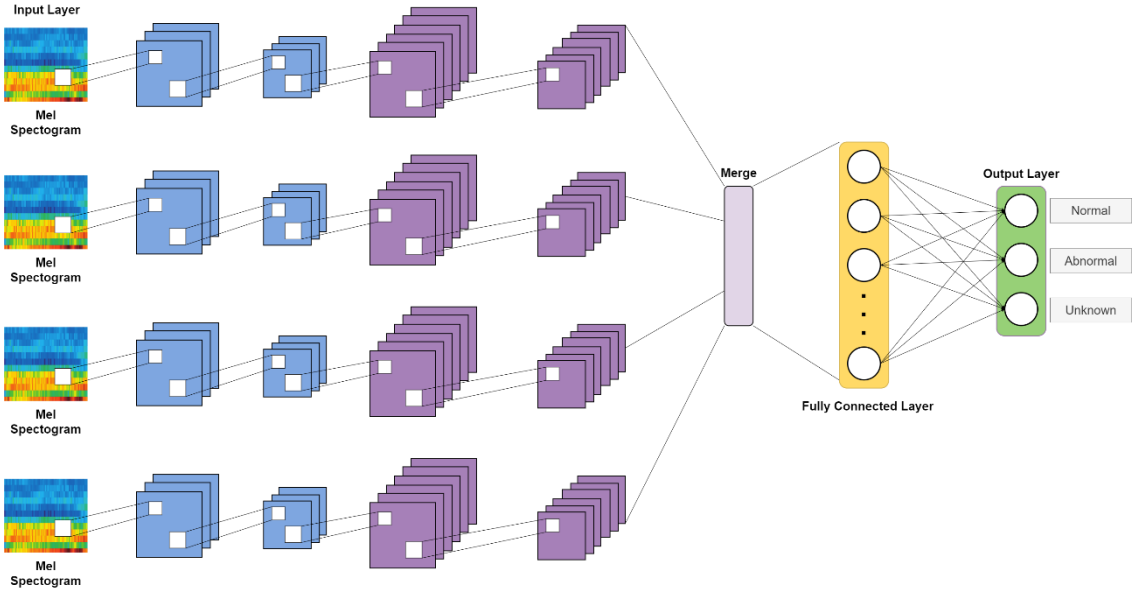

Figure 19 – Architecture of the alternative 2 DL Model

Advantages:

- The DL model consider all the files for the classification.
- All data is processed one time only.

Disadvantages/Limitations:

- Single DL model with multiple inputs and outputs, increasing the implementation complexity.

5.2.3 Alternative 3

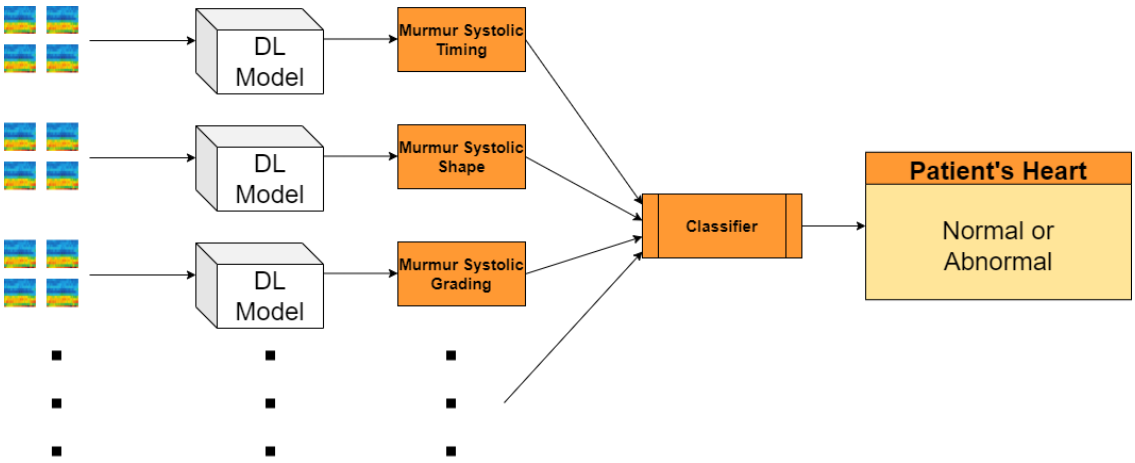

Figure 20 – System architecture alternative 3

A third alternative design is to have the four auscultation locations as input of several DL models and one model for each murmur characteristic (systolic and diastole timing, shape, grading, pitch and quality). In this alternative there is a pre-classification of the murmur characteristics and then a classification of the patient's heart. Each pathology has a specific characteristic pattern, this design will take advantage of the correlation between the characteristics for the normality/abnormality detection.

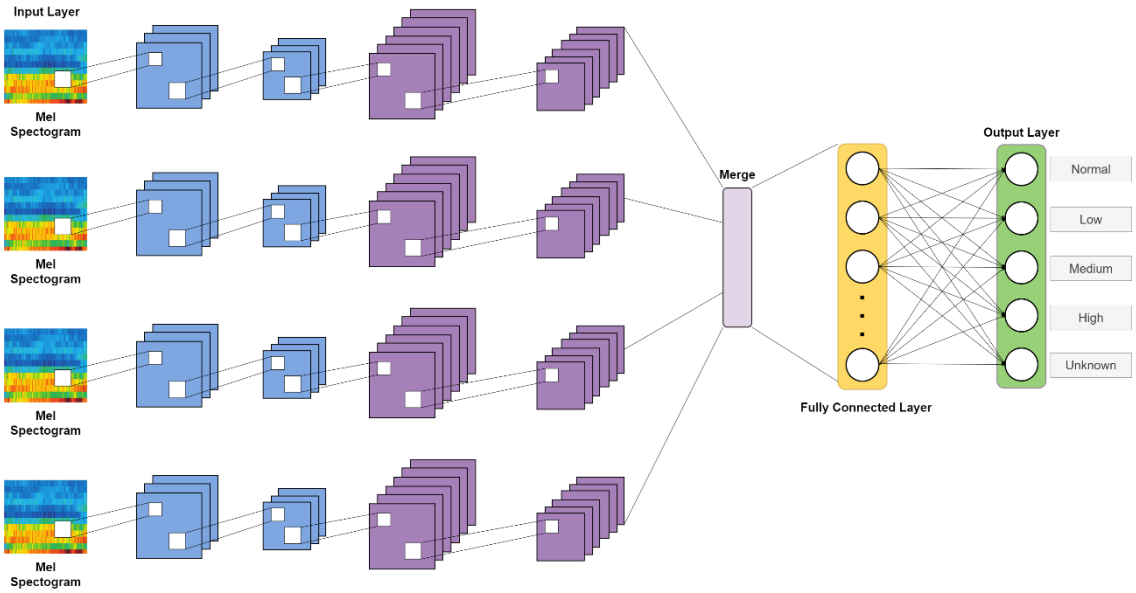

Figure 21 – Architecture of the alternative 3 DL Model for murmur pitch

Advantages:

- Each DL model would focus on a single murmur characteristic.
- The DL model consider all the sites for the classification.

Disadvantages/Limitations:

- Time complexity due to having several DL models running at the same time.

5.3 Conclusion

Three design alternatives were proposed and analysed. They differ in the DL model design as well as in their distinct objectives:

- Alternative 1, focus on a traditional approach having the auscultation recordings being analysed by different DL models, and after another model, the output of the system is if the patient should make complementary exams.

- Alternative 2, uses a single DL model that receives all the auscultation recordings with the same output of alternative 1.

- Alternative 3, uses multiple DL models to pre-classify some murmur characteristics for another DL model make the classification of the patient's heart with the same output as the other alternatives.

The final decision is that the three approaches are worthy of development.

In a final phase the developed systems will be evaluated and compared with the evaluation methodology documented in the previous section 4.

6 Implementation

In this section it is presented the implementation of the designed solutions in the previous section. The implementation is divided in 3 parts, the data pre-processing, the structure and content of the files of the project, and detailed description of the most significant experiment's models for alternative 1 and 2.

6.1 Data Pre-processing

The data for the alternatives presented in 5.2, must be processed. This is done mainly in the "Data_Preparation_Main.ipynb" script responsible to load "training.csv", the file that contains the metadata of the patients regarding the locations of the murmurs and their characteristics, to a dataframe. The data is then processed by removing unnecessary columns, renaming columns to code-oriented names (Eg. "Patient ID" to "Patient_ID") and changing some column values. These transformations are described in Table 15.

Table 15 – Transformations of the metadata

Column Name	Transformation	Result
Patient ID	Change of column name	Patient_ID
Murmur	Change of column name	Murmurs
Murmurs	Convert string values to numbers	present = 1 absent = 2 unknown = 3
Murmur locations	Change of column name	Murmurs_location
Systolic murmur timing	Change of column name	S_Timing
Systolic murmur shape	Change of column name	S_Shape
Systolic murmur grading	Change of column name	S_Grading
Systolic murmur pitch	Change of column name	S_Pitch
Systolic murmur quality	Change of column name	S_Quality
Diastolic murmur timing	Change of column name	D_Timing
Diastolic murmur shape	Change of column name	D_Shape
Diastolic murmur grading	Change of column name	D_Grading
Diastolic murmur pitch	Change of column name	D_Pitch
Diastolic murmur quality	Change of column name	D_Quality

The processed dataframe is then divided into 3 other metadata dataframes: i) metadata_alt1 composed of Patient_ID, Location and Murmurs which has the classification for each location of the patients, used in alternative 1; ii) metadata_alt3 composed of Patient_ID, S_Timing, S_Shape, S_Grading, S_Pitch, S_Quality, D_Timing, D_Shape, D_Grading, D_Pitch and D_Quality used in alternative 3 and iii) metadata_pat_mur composed of Patient_ID and Murmurs which has the overall classification of the patients, used in all 3 alternatives.

Another dataframe is created with the features extracted from the audio files being composed by Patient ID, Location and Features columns. To maximize the number of samples available, each audio file is split in segments of 3 seconds. This means that for the same patient there are multiple entries for every auscultation location. All files and segments that had a duration less than 3 seconds were discarded.

A feature extraction method was applied to each segment to calculate its Mel spectrogram. This approach is based in the works of (Doshi K. , Audio Deep Learning Made Simple - Why Mel Spectrograms perform better, 2021) and (Torres, 2021). A snippet of the features extraction code is shown in Figure 22.

```
# Extract Mel spectrogram from audio
def get_mel_spectrogram(x, sr):
    hop_length = int(numpy.round(0.015 * sr))
    sgram = librosa.stft(x, n_fft=256, hop_length=hop_length)
    sgram_mag, _ = librosa.magphase(sgram)
    mel_scale_sgram = librosa.feature.melspectrogram(S=sgram_mag, sr=sr)
    mel_spectrogram = librosa.amplitude_to_db(mel_scale_sgram, ref=numpy.min)

    return mel_spectrogram
```

Figure 22 – Features extraction method

The Features column is composed of an array with 3 dimensions, with the shape (201, 128, 1) meaning width, height, and channel, respectively. At the end of the feature extraction of the auscultations of all patients, a normalization is applied to the entire features column.

Since the data needed for each alternative is different, it was implemented a function to get and prepare the data for the experiments. To simplify this step, the division of the patients for training, validation and testing of each alternative is done separately and before the experiments, this division is recorded in txt files. In alternative 1, for the auscultation location focused models the training, validation and testing data is separated by location. For its classifier, the data is predicted with specified models and then combined in the 4 locations with the multiple segments of each location for the patients. In alternative 2, combinations of the segments of each location are made for further processing by the generator. In alternative 3, it is used a mixture of alternative 1 and 2 data preparation strategies, combinations of the location segments are made for the murmur characteristics models. For the classifier, similarly to alternative 1 classifier, predictions are made models focused on classifying the murmurs characteristics. These predictions are then combined into sets of 4, each element representing each auscultation location, to constitute the training, validation and testing samples. The use of combinations greatly increases the number of samples available.

6.2 Alternatives Implementation

For each alternative, several experiments were implemented. Regarding the structure of the models, these experiments are represented by scripts. In alternative 1 and 3 the classifier is implemented in a separated script to allow the usage of data predicted by different models. The name of the scripts reflects what they are experimenting, for example, the file "alt1_m1_genetic_03" is a 3rd attempt of a genetic model for the location models of alternative 1.

All scripts are structured in the same way, having the following sections:

- Imports
- Configuration
- Get train, validation and test data
- Build model
- Search (only for genetic models)
- Training
- Testing

The configuration section is composed by the alternative name, current experiment, experiment name to predict the data in case of the classifiers, number of epochs, batch size, learning rate and max trials for genetic models. These configurations allow the user to easily run the models of a script with different parameters and save them and its results without

overriding other experiments, simplifying further usage of the models. The configuration of two experiments can be seen in Figure 23 and Figure 24.

```
# Config
alternative_path = 'alt1'
m1_experiment_path = 'm1_static_02_2'

epochs = 30
batch_size = 50
learning_rate = 0.01
```

Figure 23 – Alternative 1 experiment configuration

```
# Config
data_path = '00_Data_alt1'

alternative_path = 'alt1'
m1_experiment_path = 'm1_static_02_3'
m2_experiment_path = 'm2_dense_01_1'

epochs = 30
batch_size = 50
learning_rate = 0.01
```

Figure 24 – Classifier configuration

Next, the data to be used in the training, validation and testing is generated using an auxiliar script “alt<n>_prepare_data”, being *n* the alternative number. This file uses the dataframes built in the pre-processing phase to generate the X and y variables needed for training and evaluating the developed models. Regarding the classifiers, the data is predicted using a specified model. Furthermore, in the data for the classifier in alternative 1, combinations of the predictions of each location are calculated, increasing the number of samples to be used. In alternative 2 and 3 the same approach is used for the models input. To better simulate the real cases, the combinations include samples without values/with -1 to represent patients with missing auscultation locations files.

The next step is the creation of the model, in which the experiments differ between themselves, more details of this differences can be consulted in sections 6.3 and 6.4. This is implemented in a function, as seen in Figure 25, to easily create multiple models. All models were compiled using the categorical cross entropy loss function and the Adam optimizer.

```python
def build_model():
    input = tf.keras.layers.Input(shape=(201, 128, 1))
    x = tf.keras.layers.BatchNormalization()(input)

    # Layer 1
    x = tf.keras.layers.Conv2D(filters=16, kernel_size=3, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)

    # Layer 2
    x = tf.keras.layers.Conv2D(filters=32, kernel_size=3, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)

    # Layer 3
    x = tf.keras.layers.Conv2D(filters=64, kernel_size=3, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)

    # Layer 4
    x = tf.keras.layers.Conv2D(filters=32, kernel_size=3, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)

    # Layer 5
    x = tf.keras.layers.Conv2D(filters=16, kernel_size=3, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)

    # Fully connected layer 2
    x = tf.keras.layers.Conv2D(filters=3, kernel_size=1, strides=1)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.GlobalMaxPooling2D()(x)
    predictions = tf.keras.layers.Activation('softmax')(x)

    model = tf.keras.Model(inputs=input, outputs=predictions)
    metrics = ['accuracy', tf.keras.metrics.Precision(class_id=0), tf.keras.metrics.Recall(class_id=0)]
    model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate), metrics=metrics)

    return model
```

Figure 25 – Function to build a model

Tuning of the hyperparameters was done using genetic algorithm, in this project they were given the name of genetic models for simplification. In these models, initially the framework keras-tuner was being used, but because of its limitations regarding the tuning objective and learning rate, a decision was made to use Optuna (Optuna, 2022) instead. More information about this decision is addressed in section 6.3.2.

The layers filters range of values are calculated based on the layers of fixed hyperparameters experiments where the first and last layers has 16 filters. The range of values for the filters of a layer follows the formula:

$$[2^{(l+3)} - \frac{2^{(l+4)} - 2^{(l+3)}}{2} - 2\,;\; 2^{(l+3)} + \frac{2^{(l+3)} - 2^{(l+2)}}{2}]$$

Or if the layer is after the middle layer, the formula:

$$[2^{(2m-l+3)} - \frac{2^{(2m-l+4)} - 2^{(2m-l+3)}}{2} - 2\,;\; 2^{(2m-l+3)} + \frac{2^{(2m-l+3)} - 2^{(2m-l+2)}}{2}]$$

Where *l* is the layer number and *m* is the number of the layer in the middle. In Table 16 the range values for a model with 7 layers is presented.

Table 16 – Filter values for a model with 7 layers

Layer	Min value	Max value
1	12	22
2	24	46
3	48	94
4	96	190
5	48	94
6	24	46
7	12	22

Next is the training of the model: it uses model checkpoints which allows saving the best epoch model, the metrics being monitored are the accuracy, precision and recall of the validation dataset. After the model is trained, plots of the loss, accuracy, precision and recall variation during training are saved as images as well as the trained models (Figure 26). In early versions of this function, only one of the metrics was being monitored.

```
def train_model_3ch(model, train_X, train_y, val_X, val_y, batch_size, epochs, class_weights, save_path):

    # Create checkout
    checkout_a = ModelCheckpoint(f'{save_path}/checkout_a.hdf5', save_best_only=True, save_weights_only=True, monitor='val_accuracy',
                                 verbose=1, mode='max')
    checkout_p = ModelCheckpoint(f'{save_path}/checkout_p.hdf5', save_best_only=True, save_weights_only=True, monitor='val_precision',
                                 verbose=1, mode='max')
    checkout_r = ModelCheckpoint(f'{save_path}/checkout_r.hdf5', save_best_only=True, save_weights_only=True, monitor='val_recall',
                                 verbose=1, mode='max')

    # Train model
    history = model.fit(x=train_X, y=train_y, batch_size=batch_size, epochs=epochs, validation_data=(val_X, val_y),
                        validation_batch_size=batch_size, class_weight=class_weights, callbacks=[checkout_a, checkout_p, checkout_r])

    # Plots
    save_training_plots(history, f'{save_path}')

    # Save model
    model.load_weights(f'{save_path}/checkout_a.hdf5')
    model.save(f'{save_path}/trained_a.h5')

    model.load_weights(f'{save_path}/checkout_p.hdf5')
    model.save(f'{save_path}/trained_p.h5')

    model.load_weights(f'{save_path}/checkout_r.hdf5')
    model.save(f'{save_path}/trained_r.h5')

    return model
```

Figure 26 – Alternatives training function

Finally, an evaluation of the model is done using the data from the test patients. This evaluation is composed of confusion matrixes and a classification reports. In Figure 27 it is presented an example of the classification report.

```
Best accuracy epoch model:
1899/1899 [==============================] - 3s 1ms/step
              precision    recall  f1-score   support

     Present       0.54      0.55      0.54      5389
      Absent       0.96      0.92      0.94     55296
     Unknown       0.02      0.49      0.03        82

    accuracy                           0.88     60767
   macro avg       0.51      0.65      0.50     60767
weighted avg       0.92      0.88      0.90     60767
```

Figure 27 – Classification report of a model

6.3 Alternative 1

6.3.1 Fixed Models

The first approach for implementing alternative 1, specifically the models focused on each auscultation location (PV, TV and MV), were simple CNNs with only Conv2D layers with ReLu activation functions disposed sequentially, with fixed hyperparameters values as illustrated in Figure 28. The input shape was three dimensional, with no restrictions to the width and height of the spectrogram, but with a mandatory 1 channel only. The filter values of the Conv2D layers assume values that are powers of two, starting at 16. In the end, a GlobalMaxPooling2D layer is used to down sample the data along its spatial dimensions and a Softmax activation function gets the predictions of the data.

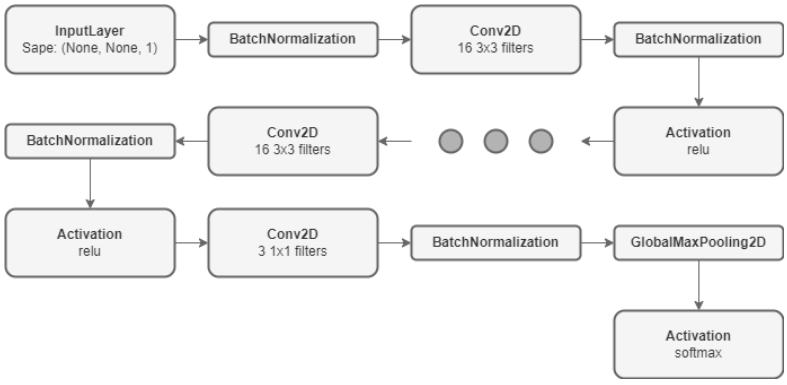

Figure 28 – Architecture of alternative 1 fixed hyperparameters models

The first experiments were models with this structure and received part of the data available for the training. Although the models developed had poor results, they served to get knowledge on how models are constructed and their needs, like the type and shape of the

input data. This new knowledge was the base to create the pre-processing steps previously documented in section 6.1.

Next, BatchNormalization layers were added to the model, these layers were applied to the input data and the output of each Conv2D layer.

The dataset being used is imbalanced, because there are more samples of the class "Absent" than the others. For these situations, the training function (Keras fit function) can receive a parameter called class_weight where the user can set the weight for each class in order to tell the model what adjustments it must make while training. If a weight of 5 is set to a class, the model will count each sample of that class as 5 samples. Initially the weights were calculated following a TensorFlow documentation tutorial at (TensorFlow, 2022) and the weights given to each class were rounded values. Later it was changed to being calculated by a sickit-learn function and keeping the most common class weight with the value of 1. The class weights calculated can be consulted in Table 17.

Table 17 – Class weights for all patients in each location

Location	Present	Absent	Unknown
AV	2.39	0.41	5.84
PV	1.99	0.42	8.73
TV	2.01	0.42	10.15
MV	2.07	0.43	5.05

In this dataset one of the characteristics of the patient's murmurs is its grade, murmurs with grade I/IV and II/IV, difficult the ability of models to find a pattern between absent and present murmurs. For this reason, it was made the decision to remove the patients with those types of murmurs, leaving only patients with absent or aggressive murmurs, to try to get models with better metrics. Removing samples from the dataset changes the weights of the classes, so they had to be calculated again. In Table 18 the updated values are documented.

Table 18 – Class weights for patients with absent and aggressive murmurs in each location

Location	Present	Absent	Unknown
AV	5.15	0.38	5.09
PV	4.58	0.38	7.58
TV	4.64	0.37	8.83
MV	5.43	0.39	4.36

To get the model with the best metrics from all the training epochs, a ModelCheckpoint was added to the implementation. Initially only the validation accuracy was being monitored, but as the experiments with the base tests in section 7.2 exemplify, using only one metric to evaluate the model can potentially discard good models. The main problem with ModelCheckpoint is that it only allows choosing one metric to be monitored, in order to

overcome this issue two more checkpoints were added to the training. These new checkpoints were monitoring the validation precision and recall. The three models selected by the checkpoints can then be compared to find the best model. A code snippet of the checkpoints being used in the training function is shown in Figure 29.

```
def train_model_gen_3ch(model, train_gen, val_gen, batch_size, epochs, class_weight, save_path):

    # Create checkout
    checkout_a = ModelCheckpoint(f'{save_path}/checkout_a.hdf5', save_best_only=True, save_weights_only=True,
                                 monitor='val_accuracy', verbose=1, mode='max')
    checkout_p = ModelCheckpoint(f'{save_path}/checkout_p.hdf5', save_best_only=True, save_weights_only=True,
                                 monitor='val_precision', verbose=1, mode='max')
    checkout_r = ModelCheckpoint(f'{save_path}/checkout_r.hdf5', save_best_only=True, save_weights_only=True,
                                 monitor='val_recall', verbose=1, mode='max')

    # Train model
    history = model.fit(x=train_gen, batch_size=batch_size, epochs=epochs, validation_data=(val_gen),
                        validation_batch_size=batch_size, class_weight=class_weight,
                        callbacks=[checkout_a, checkout_p, checkout_r])

    # Plots
    save_training_plots(history, f'{save_path}')

    # Save model
    model.load_weights(f'{save_path}/checkout_a.hdf5')
    model.save(f'{save_path}/trained_a.h5')

    model.load_weights(f'{save_path}/checkout_p.hdf5')
    model.save(f'{save_path}/trained_p.h5')

    model.load_weights(f'{save_path}/checkout_r.hdf5')
    model.save(f'{save_path}/trained_r.h5')

    return model
```

Figure 29 – Code snippet of the training process

6.3.2 Genetic Models

On a more elaborated approach, genetic models were developed with the same number of layers and characteristics of the previous approach to search for the best hyperparameters for each layer, specifically the filters parameter values. For this purpose, the Keras tuner framework was used since it allows an easy creation of genetic algorithms.

From the available tuners, the BayesianOptimization shown in Figure 30 was chosen for this task because it considers the first n trails result to calculate the next trials hyperparameters. This tuner objective was to maximize the recall of the validation data since it represents the percentage of patients that have murmurs present and were classified correctly.

```
dir_path = f'{alternative_path}/{m1_experiment_path}'

AV_tuner = BayesianOptimization(
    build_model,
    max_trials=max_trials,
    objective=keras_tuner.Objective("val_recall", direction="max"),
    directory=dir_path,
    project_name="tunner_av"
)
```

Figure 30 – BayesianOptimization Tuner for alternative 1 AV location model

One of the limitations of Keras tuners is that the learning rate for the model can’t be adjusted, so the default value of 1 is used for the search.

Analysing the obtained search results, the models that were being selected were the ones that predicted almost all patients as having murmurs present, which is problematic. This happens because the metric used to evaluate the model is not the most adequate. F1 could be a proper metric but it is not supported by Keras. Another solution would be using an additional metric to complement the already used recall, specifically the precision that represents the percentage of patients predicted as having murmurs present that were classified correctly, but unfortunately, multi-objective optimization is not supported by Keras.

For the previously mentioned reasons, another framework named Optuna that focus in hyperparameter optimization was used. It also supports tunning the learning rate of the model but a fixed learning rate of 0.0001 was used.

The same strategy used in the previous approach was implemented, where three model checkpoints are used. To define the best model, the F1 score of the 3 selected models was calculated.

6.3.3 Pre-trained Models

Another approach to the alternative presented in section 5.2.1 is using pre-trained models. The model chosen is the Xception, a convolutional neural network that is 81 layers deep and 22.9M parameters (Keras, 2022). At the end of this model, multiple layers were added to prevent drastic down sampling.

Several adjustments had to be made for the input data be compatible with the models. First, pre-trained models only allow an input with 3 channels and the data had only a single channel, the way to solve this issue was to transform the last dimension to an array of 3 equal values. Another issue was regarding the shape of the data, the size of the image had to be at least 299 x 299 for the used model and the added layers increased even more this size. This problem of the size of the spectrogram image was solved by increasing the duration of the segments.

Experiments with the Xception model were made but with no success. It was not possible to train the Xception model due to computational limitations.

6.3.4 Multi Model Strategy

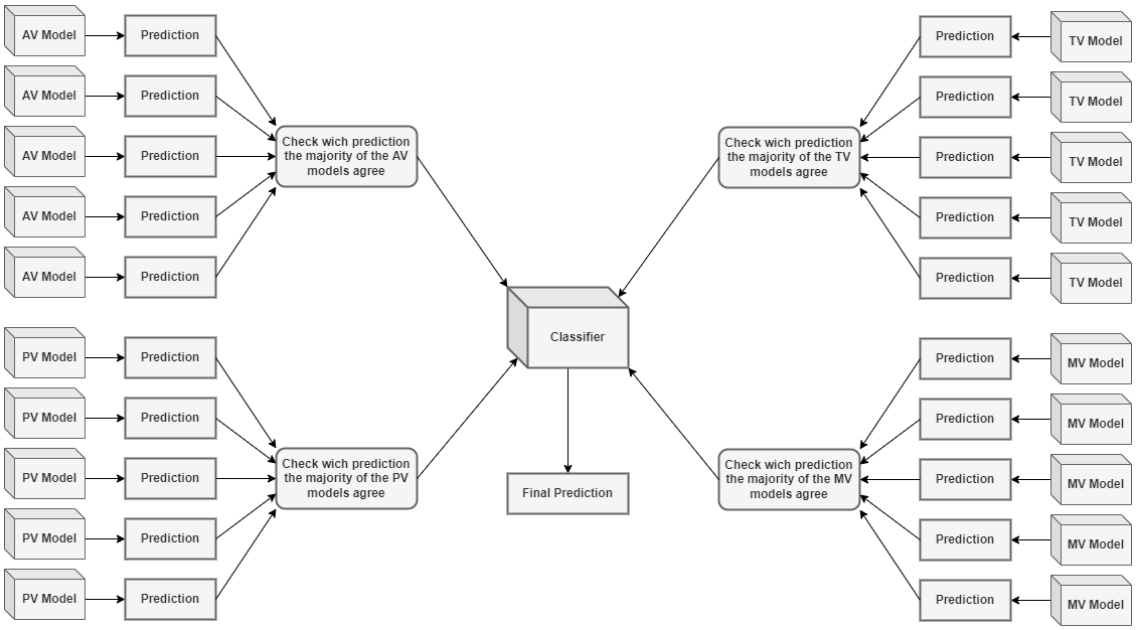

Figure 31 – Diagram of a multi model approach

To improve the results of the fixed hyperparameters and genetic models, another approach was developed (Figure 31). Instead of using just one model for the predictions, multiple models could be used, the predictions were then compared with each other. The class predicted by most of the models would be the input of the classifier.

6.3.5 Classifier

To decide regarding the patient's heart situation, a classifier was implemented. There were two versions of it, a simple model with a single Dense layer and a model with multiple Dense layers to expand and compress the data. The input of these models is a concatenation of the predictions of the four auscultation locations of the patients, to increase the samples available, these predictions were combined. The same strategy of using three checkpoints monitoring three different metrics was used.

6.3.6 Significant models details

6.3.6.1 Model of 5 layers with fixed hyperparameters

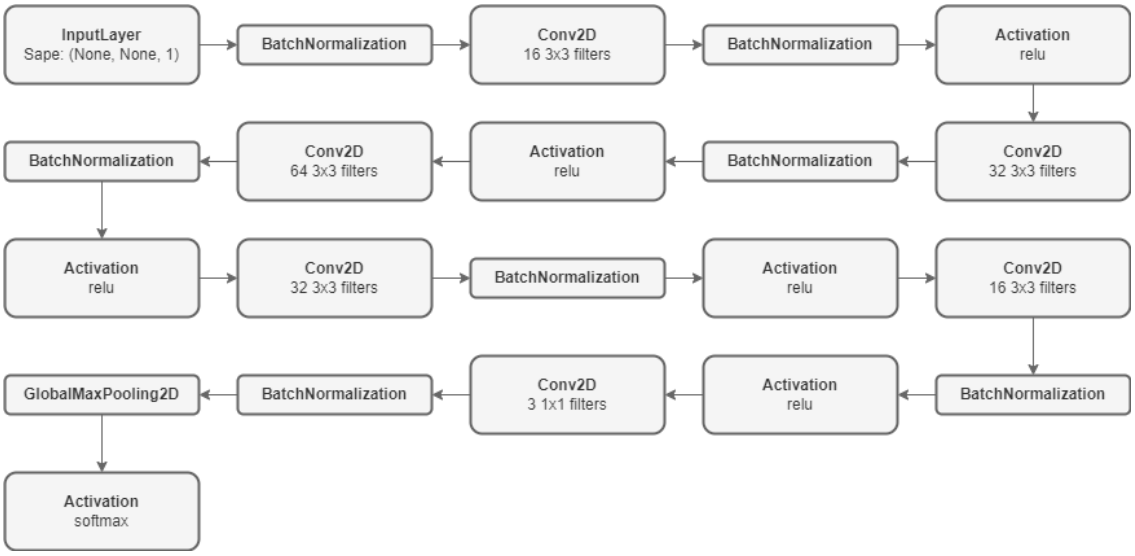

Figure 32 – Diagram of the fixed hyperparameters model of 5 layers

The model illustrated in Figure 32 is composed of multiple layers disposed sequentially, it has BatchNormalization, Conv2D, Activation and GlobalMaxPooling3D layers. The input is 3 dimensional representing the width, height, and number of channels of the Mel spectrogram. This input is first subjected to a batch normalization and then fed to a repetitive sequence of three layers. These layers are Conv2D, BatchNormalization and Activation, it repeats 5 times with different number of filters in the Conv2D layer and a fixed kernel size of 3. In the activation layer it is used the ReLu activation function. This model has a total of 46763 trainable params.

Multiple experiments were carried out since the development of the alternatives is an iterative process. Here are listed some of the experiments:

- Training with all the available patients without taking care of the imbalance.
- Experimentation with different filter values.
- Use of class weights presented in Table 19 to indicate the imbalance of the dataset to the model.
- Use of checkpoints focusing on a metric of the validation data.
- Training the model with only patients with absent murmurs and aggressive murmurs.
- Use of 3 checkpoints to monitor accuracy, precision and recall of the validation data.

- Different values for batch size, epochs and learning rate.

Table 19 – Class weights used in the fixed hyperparameters models training.

Location	Present	Absent	Unknown
AV	10.0	0.5	8.0
PV	10.0	0.5	11.0
TV	10.0	0.5	10.0
MV	10.0	0.5	6.0

6.3.6.2 Genetic model of 5 layers

The architecture of the genetic model is similar to the previous, the only difference being the filter values, in this case the filters change during the model tuning. In this approach, the filters determined by the tuner used.

Table 20 – Class weights used in the genetic model training

Location	Present	Absent	Unknown
AV	5.15	0.38	5.09
PV	4.58	0.38	7.58
TV	4.64	0.37	8.83
MV	5.43	0.39	4.36

The tuning of the model was done using only the absent and most aggressive murmurs with the weights presented in Table 20. Based in the training graphs of the fixed hyperparameters models, it was determined to use 30 epochs because after this number of epochs the metric values had minimal changes. It was also used a batch size of 100 and a learning rate of 0.0001. This tuning had a maximum of 100 trials and the results of the search are documented in Table 21.

Table 21 – Filters values after search

Location	Layer 1 filters	Layer 2 filters	Layer 3 filters	Layer 4 filters	Layer 5 filters	Trainable parameters
AV	18	46	62	34	18	58,361
PV	16	38	90	36	22	73,367
TV	16	26	50	36	12	36,143
MV	22	24	88	24	12	46,115

6.3.6.3 Classifier with single dense layer

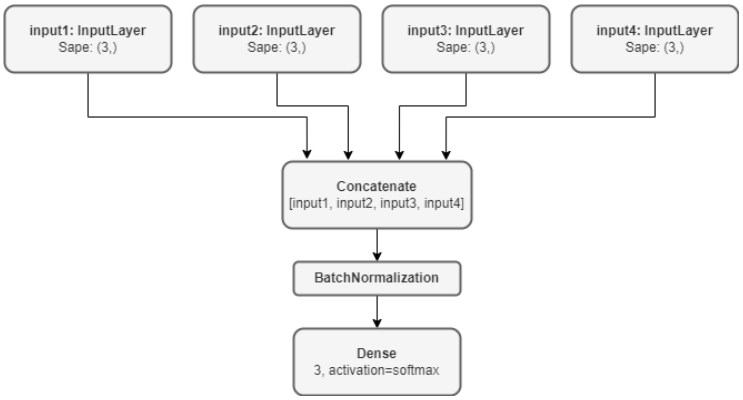

Figure 33 – Architecture of the classifier with a single dense layer

As shown in Figure 33 this model is very simple having only a Concatenate layer to join the inputs received, one for each auscultation location prediction of the patient, a BatchNormalization and a Dense layer with the 3 possible output values using the softmax activation function. The model has a total of 63 trainable parameters.

6.3.6.4 Classifier with multiple dense layers

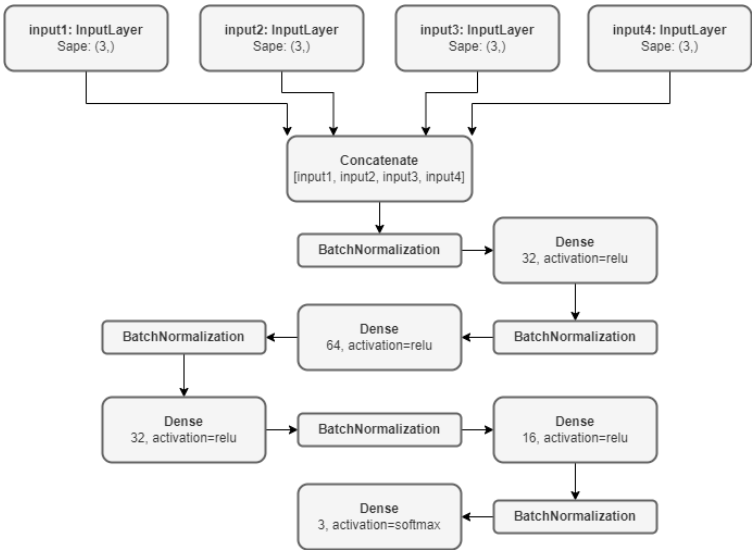

Figure 34 – Classifier model with multiple dense layers

In the more complex approach of the classifier a model with multiple layers was developed. It is composed of five dense layers that expand and compress the data as presented in Figure 34. This model has a total of 5499 trainable parameters.

6.4 Alternative 2

Taking advantage of the knowledge acquired in the implementation of alternative 1, the approaches of this alternative follow the same strategy. Starting with models with fixed hyperparameters, followed up by genetic models and pretrained models. The main difference is the architecture of the alternative, specifically the input of the models.

In alternative 2 the input is a set of auscultation locations features, which means that there are four 3 dimensional inputs. To merge these inputs together a Concatenate layer was used and like the name suggests, it concatenates a list of inputs. This allows doing combinations of the data as described in section 6.1, vastly increasing the number of samples. Unfortunately, it is not possible to use all the data because it would take a lot of time and computational resources. To solve this issue a generator was developed that received information of the combinations and during the training transform that information into real data.

The generator creates a subset of the data with the number of samples specified in a balanced way, in that subset 45% of the patients belong to the class “Present”, other 45% to the class “Absent” and the remaining 10% to the class “Unknown”.

Other than the input, there is also a huge difference in the amount of data available between this alternative and the previous one. As explained in section 6.1, the strategy used to combine all audio segments of patients gives the developer the opportunity to work with 2 million samples rather than only the 3000 of alternative 1. Due to computation constraints, the number of samples used for training, validation, and testing had to be limited.

6.4.1 Significant models details

6.4.1.1 Model of 5 layers with fixed hyperparameters

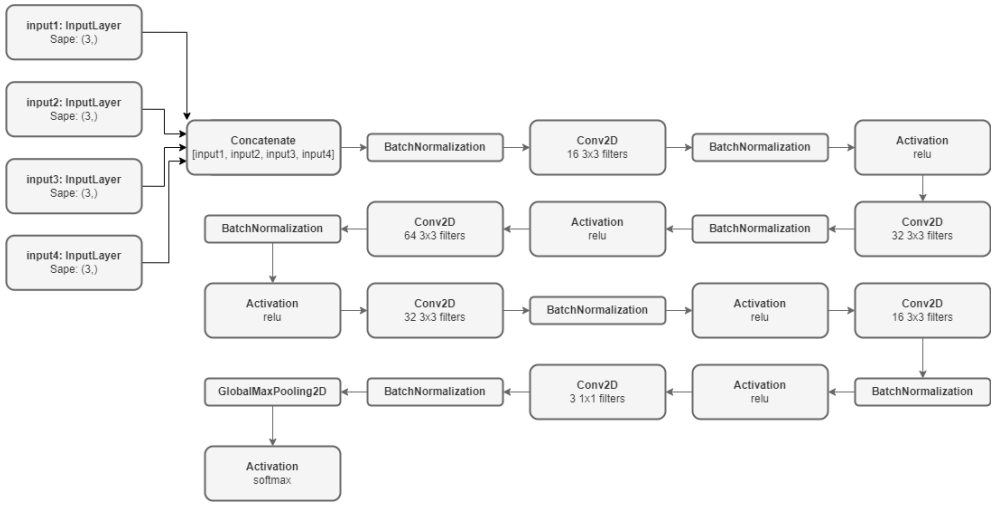

Figure 35 – Architecture of alternative 2 fixed hyperparameters models

As shown in Figure 35, the only difference to the model presented in 6.3.6.1 is the input structure. In this case, it is composed of four Mel spectrograms concatenated, in which each spectrogram represents an auscultation location of the patient. This model has a total of 47201 trainable parameters.

Since the generator developed for alternative 2 data balanced the data between present and absent murmurs, the only adjustment needed regarding class weights was to the unknown class. For this, class weights of 1.0, 1.0, and 4.5 for present, absent, and unknown were used.

6.4.1.2 Genetic model of 5 layers

Analysing the training done to the fixed hyperparameters models, it was determined to use 30 epochs in the model tuning. It was also used a batch size of 500 and a learning rate of 0.0001. This tuning had a maximum of 100 trials and was done using 10000 training samples and 5000 validation samples, the results of the search are documented in Table 22. Due to time constraints, only 20 trials were completed.

Table 22 – Filters values after search

Layer 1 filters	Layer 2 filters	Layer 3 filters	Layer 4 filters	Layer 5 filters	Trainable parameters
14	32	56	30	12	39,509

7 Evaluation

In this section it is presented the results obtained in some of the most significative experiments for alternative 1 and 2. A comparison between the alternatives and with the literature presented in 3.8 is also made.

7.1 Methodology

As explained in section 4.3, all the models were evaluated with the metrics: accuracy, precision, recall and F1 score. The evaluation was done with only the patients previously divided for testing, avoiding any overlapping between the datasets used for training, validation, and testing.

7.2 Base Tests

To establish base metrics for checking the performance of the models developed, some basic experimentations were done. These experimentations have no intelligent mechanisms and only do basic operations. As said before in section 4.3, the class considered as positive is the “Present” class.

In alternative 1 there is a separation between the models that classify each auscultation location spectrogram and the final classifier that predicts the patient heart murmurs presence. The first test pretends checking the metric results if all samples are classified with the class present. In the opposite side, the second test classifies all samples as being of the class absent. The metrics for these two tests are documented in Table 23 and Table 24.

Table 23 – Metrics if all samples are classified as class present

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.14**	**0.14**	**1.00**	**0.14**
PV	0.11	0.11	**1.00**	0.11
TV	0.13	0.13	**1.00**	0.13
MV	**0.14**	**0.14**	**1.00**	**0.14**

As Table 23 shows, the best metrics were achieved by the AV and MV auscultation locations with 0.14 for accuracy, precision and F1 score and 1.00 for recall.

Table 24 – Metrics if all samples are classified as class absent

Location	Accuracy	Precision	Recall	F1 Score
AV	0.79	0.00	0.00	0.00
PV	**0.82**	0.00	0.00	0.00
TV	0.80	0.00	0.00	0.00
MV	0.79	0.00	0.00	0.00

As for the nest test, classifying all samples as absent, the best metrics were achieved by the PV auscultation location with 0.82 accuracy and 0.00 for the rest of the metrics.

Using another test strategy, randomizing the prediction based in the dataset classes percentages, the metrics in Table 25 were obtained.

Table 25 – Metrics if the samples are classified in a randomized way

Location	Accuracy	Precision	Recall	F1 Score
AV	0.67	0.15	0.25	0.19
PV	**0.79**	**0.26**	**0.38**	**0.31**
TV	0.67	0.08	0.13	0.10
MV	0.68	0.09	0.13	0.11

When using a randomized test strategy, the best metrics were achieved by the PV auscultation location, but if this test is repeated, it can easily change.

Regarding the methods to classify the patient's murmurs, the three previous classification experimentations were applied, resulting in the metrics in Table 26.

Table 26 – Metrics of the tests classifying the patient's hearts using the alternative 1 test patients

Prediction strategy	Accuracy	Precision	Recall	F1 Score
All present	0.21	0.21	1.00	0.35
All absent	0.72	0.00	0.00	0.00
Randomized	0.63	0.25	0.33	0.28

Another prediction strategy was used for the patient's heart classification using the models for the auscultation locations developed for alternative 1. In this teste, if there was at least a model that classified the patient's heart as having the presence of murmurs, then that would be the final decision. Using this strategy, the most significant models were tested and documented in Table 27.

Table 27 – Metrics when using the strategy of at least one

Model	Accuracy	Precision	Recall	F1 Score
Fixed hp models tf	0.88	0.40	0.89	0.55
Fixed hp models sickit	0.43	0.12	**0.97**	0.21
Best of 5 fixed hp models	0.49	0.14	0.96	0.24
Genetic models	0.88	0.39	0.86	0.54
Best 5 of genetic models	**0.93**	**0.56**	0.84	**0.67**

Using the at least one strategy, the best results were obtained when using the best of 5 genetic models to feed the test, having 0.93 accuracy, 0.56 precision, 0.97 recall and 0.67 F1 score.

In alternative 2, because of the difference of the data available, the test metrics are slightly different but similar, except for the random classification algorithm as it can be seen in Table 28.

Table 28 – Metrics of the tests classifying the patient's heart using the alternative 2 test patients

Prediction strategy	Accuracy	Precision	Recall	F1 Score
All present	0.19	0.19	1.00	0.32
All absent	0.74	0.00	0.00	0.00
Randomized	0.63	0.16	0.19	0.17

7.3 Alternative 1

Regarding the auscultation location focused models, the initial experiments were done using all the patients available and without indicating the imbalance of the dataset to the models

which resulted in bad results. In Table 29 it is documented one of these first model's metrics, it refers to a fixed hyperparameters model of 5 layers.

Table 29 – Metrics of the fixed hyperparameters models using all patients and without class weights

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.80**	0.06	0.02	0.03
PV	0.79	0.11	0.10	0.10
TV	**0.80**	**0.33**	**0.24**	**0.28**
MV	0.75	0.14	0.10	0.12

Analysing the Table 29, we can see that this model is predominantly classifying the patient murmurs as absent, having better metrics in the TV location. This location presented better results than the base experiment that classifies every sample as absent, but all other locations were presented worse results.

After filtering the patients by absent and aggressive murmurs and adding weights to the models, slightly better results were obtained. There is a noticeable difference in the results between the use of TensorFlow tutorial-based class weights and class weights calculated by a sickit-learn (scikit-learn, 2022) function. The results of the best models can be seen in Table 30 and Table 31.

Table 30 – Metrics of the fixed hyperparameters models using TensorFlow tutorial-based class weights

Location	Accuracy	Precision	Recall	F1 Score
AV	0.87	0.13	0.07	0.09
PV	0.88	0.23	0.10	0.14
TV	**0.89**	**0.49**	**0.57**	**0.53**
MV	0.88	0.50	0.03	0.05

Table 31 – Metrics of the fixed hyperparameters models using class weights calculated by sickit-learn function

Location	Accuracy	Precision	Recall	F1 Score
AV	0.86	0.19	0.17	0.18
PV	0.77	0.16	**0.52**	0.25
TV	**0.87**	**0.41**	0.49	**0.45**
MV	0.52	0.11	0.30	0.16

In these experiments, the location TV model had significantly better metrics than the other locations, this behaviour was unexpected since there was no difference in how the data was processed and how the model was built. The same behaviour is seen in all experiments.

Using the multi model strategy documented in section 6.3.4, an experiment with the best 5 fixed hyperparameters models of the experiment with the sickit-learn class weights was conducted. This experiment slightly improved the model metrics. In Table 32 it are presented the F1 scores of the models used for this experiment, and in Table 32 its results.

Table 32 – F1 scores of the models used in the multi model strategy

Location	M1 F1 Score	M2 F1 Score	M3 F1 Score	M4 F1 Score	M5 F1 Score
AV	0.17	0.13	0.18	0.17	0.09
PV	0.18	**0.25**	0.16	0.15	0.14
TV	**0.45**	0.23	**0.32**	**0.30**	**0.37**
MV	0.12	0.16	0.12	0.16	0.14

Table 33 – Results of the multi model strategy of fixed hyperparameters models

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.90**	0.38	0.10	0.16
PV	0.61	0.11	**0.76**	0.18
TV	0.89	**0.44**	0.40	**0.42**
MV	0.60	0.11	0.72	0.19

Next are presented the metrics for the genetic models experiment. This experiment trained three models in each trial and completed 100 trials, meaning that 300 models were trained and tested. The model with the highest F1 score between all the models was considered the best model. As Table 34 shows, the metrics greatly improved when using this approach.

Table 34 – Results of the genetic models of 5 layers

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.89**	0.36	0.34	0.35
PV	**0.89**	0.50	0.48	0.49
TV	**0.89**	**0.57**	**0.49**	**0.52**
MV	0.84	0.36	0.45	0.40

Similar to the fixed hyperparameter experiment, the multi model strategy was applied to the 5 best genetic models, the F1 scores of these models are presented in Table 35.

Table 35 – Best 5 genetic models F1 scores used in the multi model strategy

Location	M1 F1 Score	M2 F1 Score	M3 F1 Score	M4 F1 Score	M5 F1 Score
AV	0.28	0.26	0.26	0.26	0.35
PV	0.40	0.45	0.43	0.46	0.49
TV	**0.50**	**0.50**	**0.52**	**0.49**	**0.51**
MV	0.33	0.40	0.32	0.31	0.32

Table 36 – Metrics of the multi model strategy with 5 best genetic models

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.91**	0.59	0.34	0.43
PV	0.90	0.52	0.45	0.48
TV	0.90	**0.61**	**0.54**	**0.58**
MV	0.85	0.35	0.35	0.35

Analysing Table 36, applying the multi model strategy metrics to the 5 best genetic models, resulted in the improvement of AV and TM location models, but a slight deterioration to PV and MV models. Applying this strategy with only the best 5 genetic models didn't result in significant better models, but an experiment with 10 models instead of 5 improved the results in all location models as shown in Table 37.

Table 37 – Metrics of multi model strategy with 10 best genetic models

Location	Accuracy	Precision	Recall	F1 Score
AV	**0.91**	**0.80**	0.28	0.41
PV	0.90	0.58	0.48	0.53
TV	0.90	0.64	**0.51**	**0.57**
MV	0.86	0.43	0.38	0.40

Regarding the classifier, only experiments that used the filtered patients are documented. In addition, the models were trained using combinations with incomplete sets of locations, meaning that samples without predictions to all locations were used. In Table 38 are presented the results for the simpler classifier, each model was trained 5 times with a batch size of 500 to ensure the best results. The experiments using the multi model strategy were also tested since it improved the models.

Table 38 – Results of classifiers with a single Dense layer

Model	Accuracy	Precision	Recall	F1 Score
Fixed hp models tf	**0.94**	**0.81**	0.64	0.72
Fixed hp models sickit	0.88	0.64	0.60	0.62
Best of 5 fixed hp models	0.93	0.78	0.51	0.62
Genetic models	0.93	0.71	0.61	0.66
Best 5 of genetic models	0.93	0.79	**0.67**	**0.73**

In the more complex approach described in section 6.3.6.4, the experiments had similar results as it can be seen in Table 39.

Table 39 – Results of classifiers with multiple dense layers

Model	Accuracy	Precision	Recall	F1 Score
Fixed hp models tf	0.89	**0.77**	0.67	**0.72**
Fixed hp models sickit	0.82	0.53	0.67	0.59
Best of 5 fixed hp models	0.89	0.70	0.49	0.58
Genetic models	0.88	0.61	0.70	0.66
Best 5 of genetic models	**0.91**	0.61	**0.76**	0.68

Some single layered classifiers were also tested using all patients to check how they would behave in a real environment. These tests are documented in Table 40.

Table 40 – Metrics of single layer classifiers using all the dataset

Model	Accuracy	Precision	Recall	F1 Score
Fixed hp models tf	**0.75**	**0.47**	0.11	0.17
Fixed hp models sickit	0.70	0.33	0.06	0.10
Genetic models	0.48	0.24	**0.41**	**0.30**

The genetic models showed a better performance among the tested models, with the cost of accuracy.

7.4 Alternative 2

In alternative 2, similarly to the first experiments in alternative 1, a model with fixed hyperparameters and with no indication of weights was tested using all patients. The trained model received all four auscultation locations instead of only one and was receiving incomplete combinations, meaning that there were combinations without a spectrogram, simulating the patient not having all locations auscultated. In the training, the model was fed by 10,000 training samples and 10,000 validation samples, being tested by another 10,000. It obtained 0.83 accuracy, 0.31 recall, 0.16 precision and 0.21 of F1 score. Although these results look better than the ones obtained in alternative 1 similar experiment, no conclusion can be taken because the models have different objectives, the first model is classifying a patient auscultation location and the second is classifying the patient when receiving all four auscultation locations.

When using only complete combos, the results of precision, recall and F1 score improved but with lowered the accuracy, passing from 0.83 to 0.38. In this experiment more samples were used to train the model, specifically 50,000 and the same 10,000 for validation and test purposes. It obtained a precision of 0.50, recall of 0.72 and F1 score of 0.59.

Adding the class weights to the model stabilized all metrics around 0.45 like the Table 41 shows. In this table it is presented the classification report metrics for the "Present" class for the models selected by each model checkpoint during the training phase.

Table 41 – Metrics of each model checkpoint for the fixed hyperparameters model of alternative 2

Monitor Metric	Accuracy	Precision	Recall	F1 Score
val_accuracy	0.48	0.44	0.46	0.45
val_precision	0.46	**0.45**	0.41	0.43
val_recall	**0.49**	**0.45**	**0.47**	**0.46**

The model that presents the best results is the one obtained by the model checkpoint that was monitoring the validation recall with 0.49 accuracy, 0.45 precision, 0.47 recall and an F1 score of 0.46.

Using a genetic algorithm to tune the model hyperparameters, the recall metrics improved by 0.34 for the best previous model. The model selected has accuracy of 0.44, precision of 0.44, recall of 0.81 and a F1 score of 0.57.

7.5 Analysis

This analysis is composed by a comparison between the experiments in each alternative including metrics and time spent in training, comparison of the experiments with the best results against the base algorithms described in section 7.2 and a comparison of the best location focused models of alternative against some of the models presented in the literature in section 3.8.

7.5.1 Comparison between experiments

To evaluate the alternative 1 experiments, regarding the models focused in the auscultation locations, their accuracy, precision, recall, and F1 score are presented in Figure 36, Figure 37, Figure 38, and Figure 39 respectively.

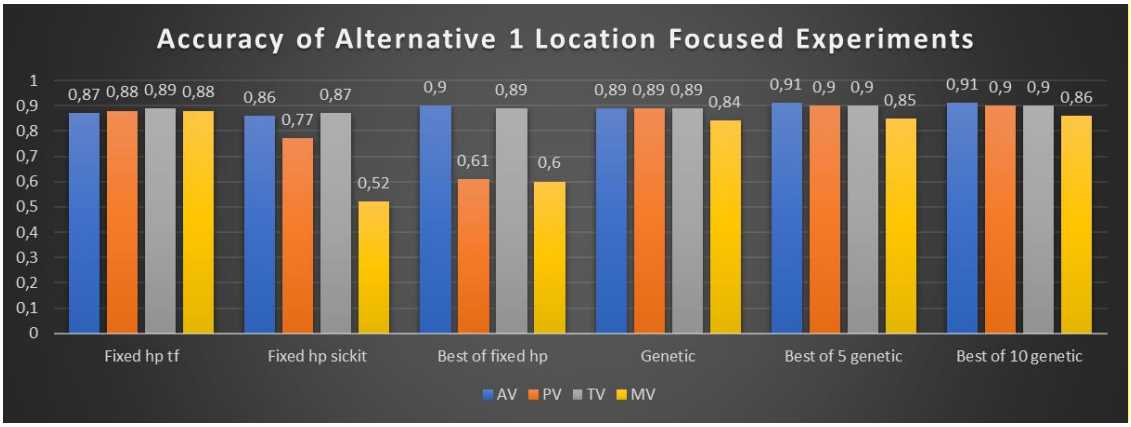

Figure 36 – Chart of alternative 1 location focused experiments accuracy

Across the experiments there is no significant variation of the model's accuracy, except for the experiments with the sickit-learn calculated weights. There is a slight improvement of this metrics when using genetic algorithms.

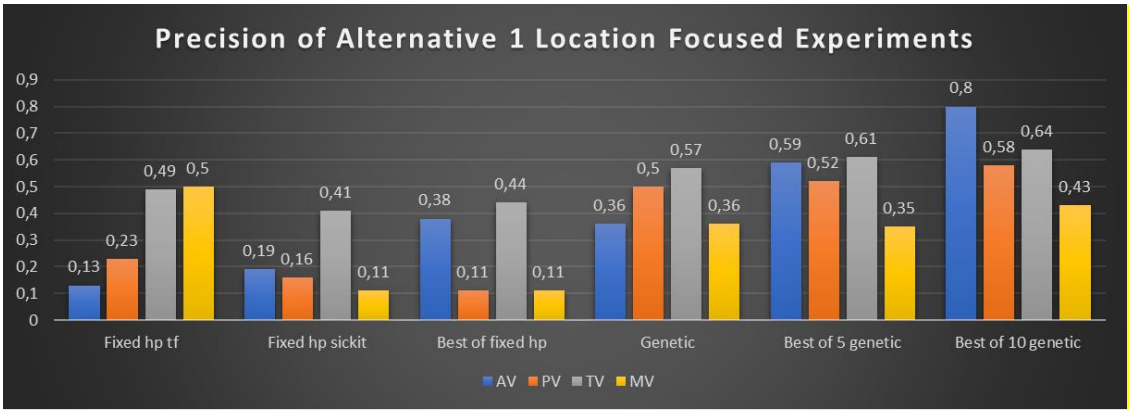

Figure 37 – Chart of alternative 1 location focused experiments precision

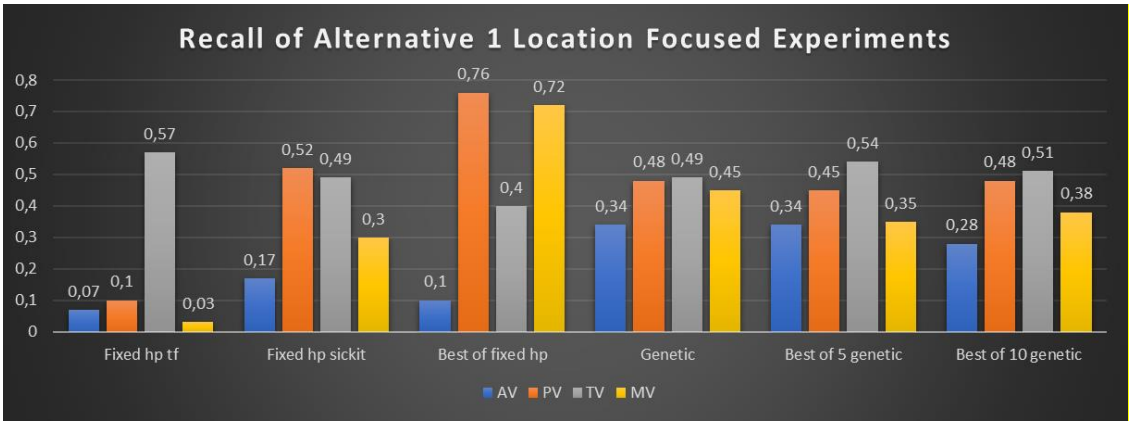

Figure 38 – Chart of alternative 1 location focused experiments recall

Looking at the information of the models regarding the precision and recall, when these metrics are analysed in a separated way no conclusions can be taken. For that reason, the F1 score metric, that correlates these two metrics, should be analysed.

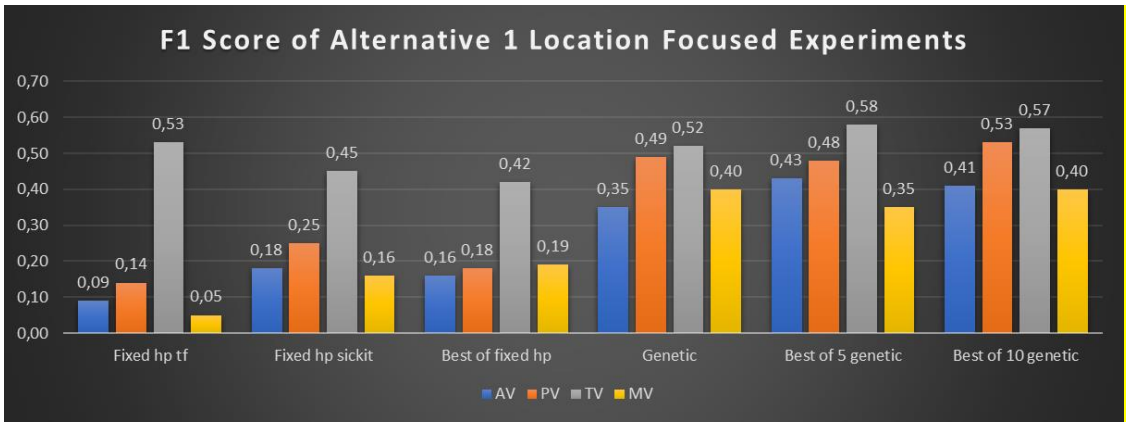

Figure 39 – Chart of alternative 1 location focused experiments F1 score

Looking at the chart in Figure 39, the genetic models achieved better performance overall among the experiments. Except for the TV location models, tunning the hyperparameters of the models increased the F1 score relatively to the previous approaches.

Between the 3 experiments with the genetic models, half the location didn't improve when using a multi model approach with the 5 best models selected by the genetic algorithm, but they all benefited when using 10 models instead of 5.

Regarding the classifier experiments, the metrics of the two developed models described in section 6.3.5 are illustrated in Figure 40, Figure 41, Figure 42 and Figure 43.

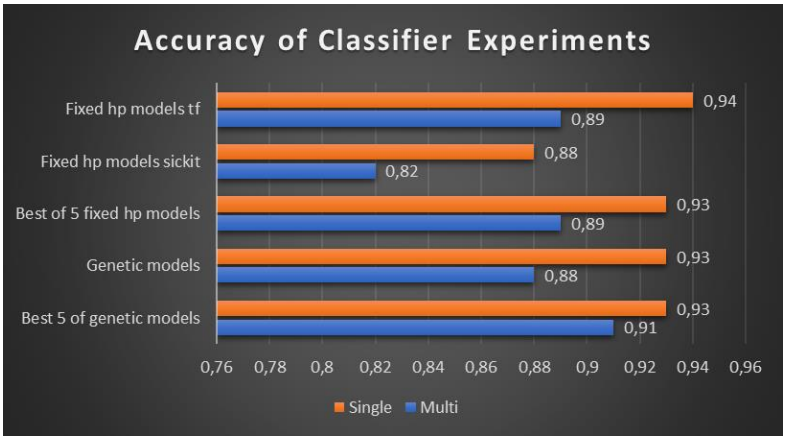

Figure 40 – Chart with the accuracy of the classifier experiments

The accuracy of the experiments with the model composed by a single dense layer surpasses the results of the multi layered model. Among all the experiments for the alternative 1

classifier the one that has better accuracy is the single layer model when using the locations models that were trained with the class weights calculated by the TensorFlow method.

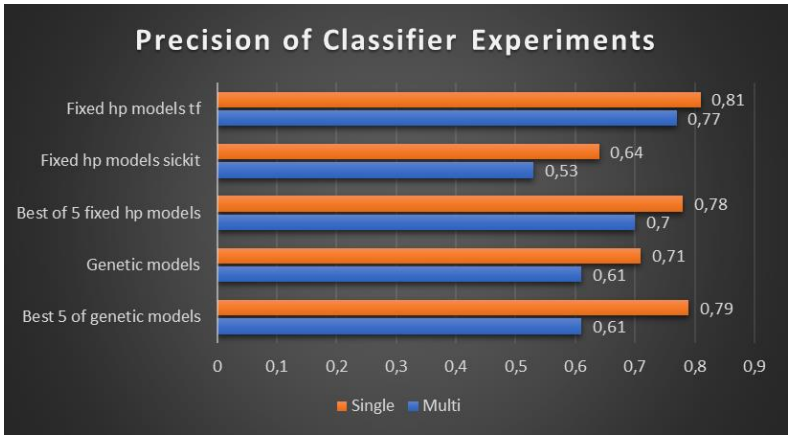

Figure 41 – Chart with the precision of the classifier experiments

In the precision, as it can be seen in Figure 41, the same behaviour of the previous metric is observed, when comparing the two models and relatively to the best model between all experiments. However, the behaviour is not the same for the recall measure.

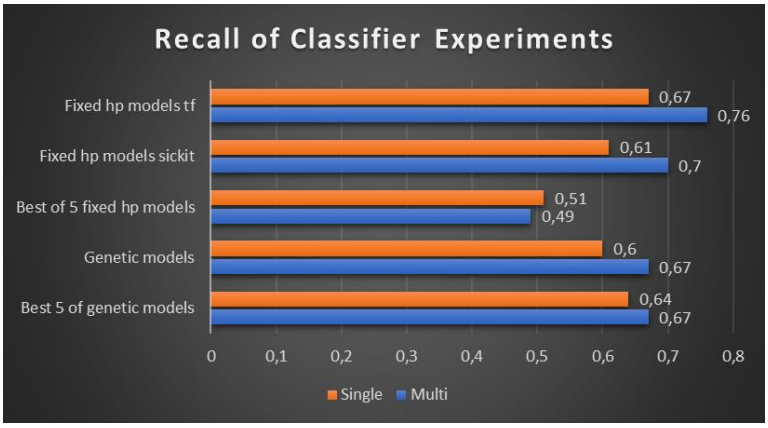

Figure 42 – Chart with the recall of the classifier experiments

As it can be seen in Figure 42, the multi layered model presents better recall values apart from when using the majority prediction among the 5 best fixed hyperparameters models. The best model regarding the recall metric is the multi layered model when using in combination with the 5 best genetic models.

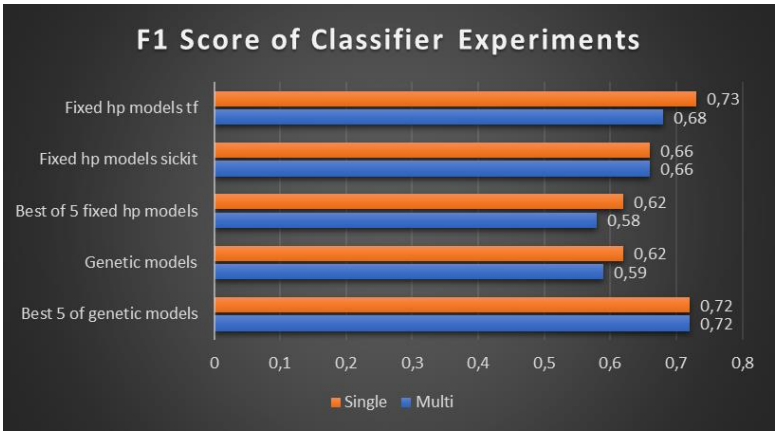

Figure 43 – Chart with the F1 score of the classifier experiments

Analysing the chart of Figure 43, the model that presents higher F1 score is the single layered model when using the fixed hyperparameters models with TensorFlow calculated weights to predict the auscultation location class.

A test using all the data instead of only the patients with absent or aggressive murmurs was also done to three of the models.

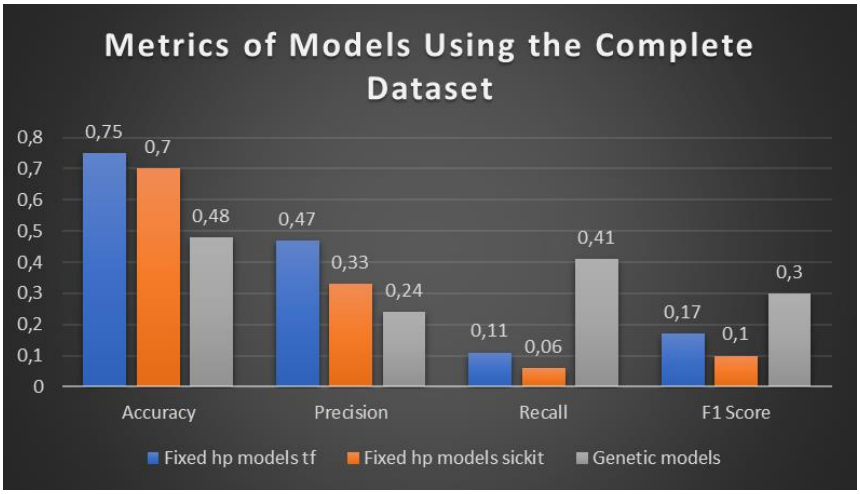

Figure 44 – Metrics of models using the complete dataset

When testing the experiments with data simulating a real environment, with all types of murmurs, the model's performance is low which is normal due to the training being done with only the filtered patients. This shows that murmurs that are not aggressive are harder to find.

Regarding alternative 2 experiments, the results are summarized in Figure 45.

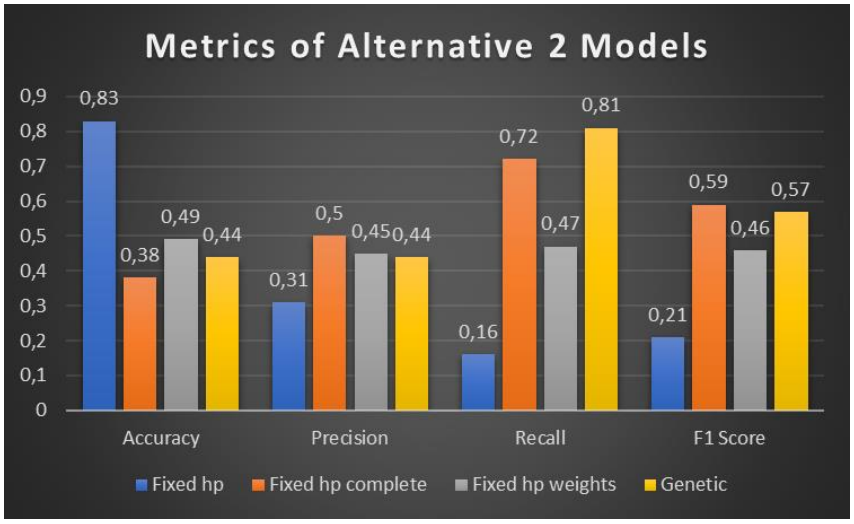

Figure 45 – Alternative 2 experiments metrics

As shown in the alternative 2 experiments metrics chart, the best model regarding accuracy was the model with the fixed hyperparameters, for precision and F1 score was the model with the fixed hyperparameters when using complete data and for recall, was the model selected by the genetic algorithm.

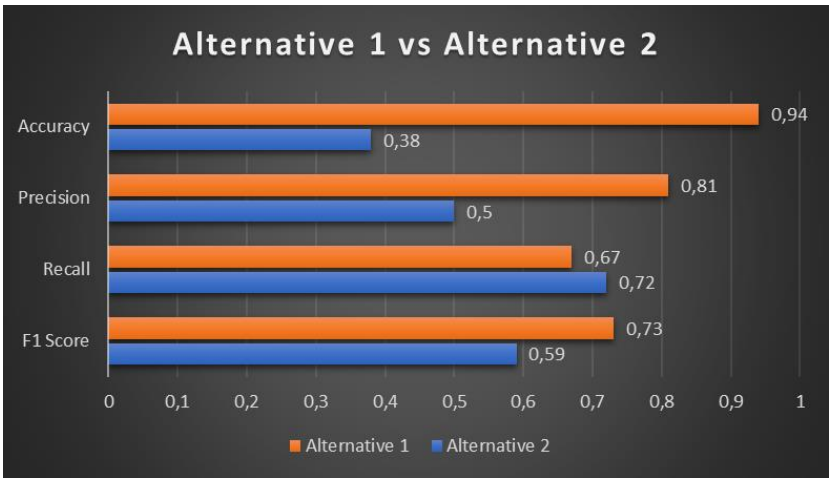

Figure 46 – Results of the models with higher F1 score of alternative 1 and 2

Between the two implemented alternatives the one that presents better results is alternative 1 because the F1 score is 0.14 higher and presents a much better accuracy and precision while the recall has only a decrease of 0.05 as shown in Figure 46 comparison chart.

In another perspective the time spent on training the models is another factor of distinction between the alternatives. These times are presented in Table 42. The times recorded include model checkpoints processing, genetic models' operations, and generator functions.

Table 42 – Times spent training the models

Alternative	Model	Number of samples*	Batch size	Time per epoch (seconds)
1	Fixed hp	3,249	100	6s
1	Genetic	3,249	100	5s
1	Single layer classifier	499,083	500	4s
1	Multi layer classifier	499,083	500	7s
2	Fixed hp	60,000	100	500s
2	Genetic	15,000	500	90s

*number of samples used for training and validation

The training of alternative 1 experiments is faster than alternative 2. This happens because of several factors: i) the model's structure is simpler than the ones from alternative 2; ii) number of samples is smaller in the location focused models, not needing as much memory as alternative 2; iii) the input of the classifier is an array of three float elements against an array with shape (201, 128), meaning each sample is significantly smaller and even with 8 times the number of samples of alternative 2, the size is smaller than the input of alternative 2.

7.5.2 Comparison of experiments with the base tests

When comparing the alternative 1 location focused models with the base experiments, the models developed have better results. The only base test that, after some tries, could surpass the models is the random output experiment, but it is not a reliable option because in a real situation we would never know which try is the right one.

In relation to the classifier of alternative 1, in the base experiments, the best metrics achieved are from the test that classifies the patient's heart murmurs as present if any of the prediction of the auscultation location focused models classifies the murmur as present. This algorithm F1 scores were added to the classifiers chart for comparison (Figure 47).

Figure 47 – F1 score of base algorithms and alternative 1 classifier experiments

Analysing the F1 scores for both base and the developed classifiers, all the implemented models have better score. Next in Figure 48 is presented a chart comparing the best base experiment with the best developed model.

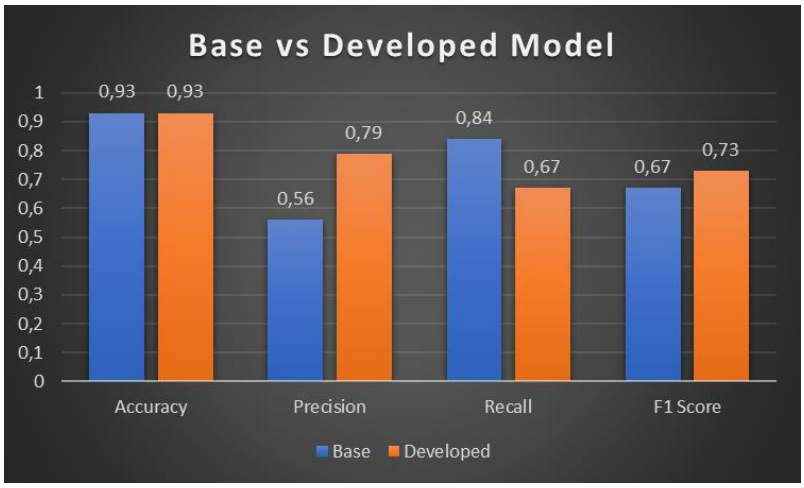

Figure 48 – Comparison of the best base and developed approaches

Although the base algorithm has better recall values, it has lower values of precision, making, not only the best, but all experiments of the alternative 1 classifier having better values of F1 score.

Regarding alternative 2, the models developed had better results than the base tests of classifying all the samples with the same class and when randomizing the prediction.

7.5.3 Comparison of developed models with literature

When comparing the models implemented for the alternatives with the approaches explored in the literature, only alternative 1 can be fairly compared, no studied literature work receives a recording of each auscultation location like in alternative 2.

Table 43 – Metrics of the literature study and the best developed models

Model	Location	Accuracy	Recall	F1 Score
Literature	AV	**0.93**	**0.92**	**0.94**
	PV	0.87	**0.81**	**0.80**
	TV	**0.90**	**0.88**	**0.90**
	MV	**0.91**	**0.86**	**0.92**
Developed Models	AV	0.91	0.34	0.43
	PV	**0.90**	0.48	0.53
	TV	**0.90**	0.54	0.58
	MV	0.86	0.38	0.40

Although it is not a fair comparison because the datasets used in both projects are not the same, the multi-label classification of heart sound signals paper described in section 3.8.2 presents better results than the location focused models of alternative 1 as it can be seen in Table 43.

8 Conclusion and Future Work

CVDs are in the top of the causes of death in the world, any help to prevent them is appreciated in the healthcare community. There are several studies with the objective of aiding the prevention and diagnosis of these diseases, they were studied and documented in the state of the art. We designed three alternatives to support on this subject. The first two alternative designs are base in the projects that use machine learning to classify if the patient heart is normal or abnormal being the main difference that we now have more than one auscultation location for each patient. The third alternative uses not only the four locations but also a pre-classification phase to classify some murmur characteristics.

During the implementation process a lot of knowledge was gained regarding audio processing, creation and tuning of deep learning models, and more specific components. Like generators, the types of layers available and what they do, how the dataset should be composed and more.

In alternative 1 a lot of experimentation was done because it was the first approach being implemented. A gradual increase of the components of the model and difficulty allowed gaining knowledge in a relaxed way. This made the entrance to a more complex alternative, like alternative 2, simpler and faster.

In the first alternative, several models achieved good results, with the accuracy above 0.81, precision above 0.52, recall above 0.48 and F1 score above 0.58 in the latest models tested. With the models developed surpassing the “at least one” test strategy, being this the conventional way of choosing the final prediction.

In alternative 2, from the four models developed, even not being from the same model, the highest accuracy achieved is 0.83, the highest precision was 0.5, the highest recall was 0.81 and the highest F1 score was 0.59.

Alternative 3 was not developed but from the early investigations to the dataset, a problem with the number of samples for each class was discovered. Some classes have less than 3

patients meaning that it is not possible to have a patient of those classes in the three datasets of training, validation, and testing. One way to solve this problem is to join some classes.

These models also require an increase of the duration of the segments retrieved from the recording or use of any other strategy to increase the spectrogram image since, currently, does not comply with the requirements of the pre-trained models studied.

Models for two of the alternatives were implemented and obtained better results than the ones established by the base experiments of section 7.2, the best model having 0.94 accuracy, 0.81 precision, 0.67 recall and F1 score of 0.73.

The model implemented obtained good results and can aid in the healthcare system. The investigation performed in this dissertation will also be of great help to the scientific community regarding the development of systems in the area of cardiovascular diseases.

As future work, we intend to work with pre-trained models, since it was not possible due to computational limitations. Implementing alternative 3 could also be a possibility. Other approaches like different model structures or more elaborated systems like the multi model strategy presented in section 6.3.4 could be investigated too.

References

Agrawal, S. K. (2021). *Understanding the Basics Of Artificial Neural Network*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/07/understanding-the-basics-of-artificial-neural-network-ann/

Ahmad, M. S., Mir, J., Ullah, M. O., Shahid, M. L., & Syed, M. A. (2019). An efcient heart murmur recognition and cardiovascular disorders.

Al-Hadithi, A. (2020). *Basics of Heart Auscultation*. Retrieved from Clinical Revision: www.clinicianrevision.com/courses/cardiology/lessons/cardiovascular-examination/topic/basics-of-heart-auscultation/

Brownlee, J. (2017). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Chaudhary, K. (2020). *Understanding Audio data, Fourier Transform, FFT and Spectrogram features for a Speech Recognition System*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520

Chen, J. (2021). *Neural Network*. Retrieved from Investopedia: https://www.investopedia.com/terms/n/neuralnetwork.asp

Cochran, W., Cooley, J., Favin, D., Helms, H., Kaenel, R., Lang, W., . . . Welch, P. (1967). What is the fast Fourier transform? *Proceedings of the IEEE*, 1664-1674.

Demir, F., Şengür, A., Bajaj, V., & Polat, K. (2019). Towards the classifcation of heart sounds.

Dertat, A. (2017). *Applied Deep Learning - Part 1: Artificial Neural Networks*. Retrieved from Towards Data Science: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

Dertat, A. (2017). *Applied Deep Learning - Part 4: Convolutional Neural Networks*. Retrieved from Towards Data Science: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Doshi, K. (2021). *Audio Deep Learning Made Simple - Why Mel Spectrograms perform better*. Retrieved from Ketan Doshi Blog: https://ketanhdoshi.github.io/Audio-Mel/

Doshi, K. (2021). *Audio Deep Learning Made Simple (Part 2): Why Mel Spectrograms perform better*. Retrieved from Towards Data Science: https://towardsdatascience.com/audio-deep-learning-made-simple-part-2-why-mel-spectrograms-perform-better-aad889a93505

Doshi, K. (2021). *Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation*. Retrieved from Towards Data Science: https://towardsdatascience.com/audio-deep-learning-made-simple-part-3-data-preparation-and-augmentation-24c6e1f6b52

Doshi, S. (2018). *Music Feature Extraction in Python*. Retrieved from Towards Data Science: https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d

EHN. (2022). *Fighting cardiovascular disease – a blueprint for EU action*. Retrieved from European Heart Network: https://ehnheart.org/eu-action-on-cvd.html

Fahad, H. M., Khan, M. U., Saba, T., Rehman, A., & Iqbal, S. (2017). Microscopic abnormality classification of cardiac murmurs using.

FPC. (2021). *Mapa do coração*. Retrieved from Fundação Portuguesa Cardiologia: http://www.fpcardiologia.pt/saude-do-coracao/mapa-do-coracao/

Goda, M. Á., & Hajas, P. (2016). Morphological Determination of Pathological PCG Signals by Time and.

Great Learning Team. (2021). *Types of Neural Networks and Definition of Neural Network*. Retrieved from Great Learning: https://www.mygreatlearning.com/blog/types-of-neural-networks/

Grzegorczyk, I., Soliński, M., Łepek, M., Perka, A., Rosiński, J., Rymko, J., . . . Gierałtowski, J. (2016). PCG Classification Using a Neural Network Approach.

Gupta, A. (2020). *Difference between ANN, CNN and RNN*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/

Had, A., Sabri, K., & Aoutoul, M. (2020). Detection of Heart Valves Closure Instants in Phonocardiogram Signals. *Wireless Personal Communications*.

Homsi, M. N., Medina, N., Hernandez, M., Quintero, N., Perpiñan, G., Quintana, A., & Warrick, P. (2016). Automatic Heart Sound Recording Classification using a Nested Set of Ensemble.

IBM. (2020). *Neural Networks*. Retrieved from IBM Cloud Education: https://www.ibm.com/cloud/learn/neural-networks

Jevon, P. (2007). Cardiovascular Examination – Part three – Auscultation of the heart. *Cardiovascular Examination*.

Kazemnejad, A., Gordany, P., & Sameni, R. (2021). *An Open–Access Simultaneous Electrocardiogram and Phonocardiogram Database*. Retrieved from bioRxiv: https://www.biorxiv.org/content/10.1101/2021.05.17.444563v2

Keras. (2022). *About Keras*. Retrieved from Keras: https://keras.io/about/

Keras. (2022). *Keras Applications*. Retrieved from Keras: https://keras.io/api/applications/

Kim, M.-Y., Johnson, J. L., & Sawatzky, R. (2019). Relationship Between Types of Social Support, Coping Strategies, and Psychological Distress in Individuals Living With Congenital Heart Disease. *The Journal of Cardiovascular Nursing vol. 34 iss. 1*, 76--84.

Korstanje, J. (2021). *The F1 score*. Retrieved from Towards Data Science: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Kumar, S. (2020). *Overview of various Optimizers in Neural Networks*. Retrieved from Towards Data Science: https://towardsdatascience.com/overview-of-various-optimizers-in-neural-networks-17c1be2df6d5

Langley, P., & Murray, A. (2016). Abnormal Heart Sounds Detected from Short Duration Unsegmented.

Latif, S., Usman, M., Rana, R., & Qadir, J. (2018). Phonocardiographic Sensing using Deep Learning for Abnormal Heartbeat Detection. *IEEE Sensors Journal 2018*, 9393-9400.

Levin, A. D., Ragazzi, A., Szot, S. L., & Ning, T. (2021). Extraction and assessment of diagnosis-relevant features for heart murmur classification.

Liu, C., Springer, D., Moody, B., Silva, I., Johnson, A., Samieinasab, M., . . . Clifford, G. D. (2016). *Classification of Heart Sound Recordings: The PhysioNet/Computing in Cardiology Challenge 2016*. Retrieved from PhysioNet: https://physionet.org/content/challenge-2016/1.0.0/

Martikainen, A. (2017). Front End of Innovation in Industrial Organization.

Medizinio. (2022). *Buying an ECG Machine - Tips and Prices*. Retrieved from Medizinio: https://medizinio.de/en/medical-equipment/ecg

Medizino. (2022). *Buying an Ultrasound Machine - Advice and Offers*. Retrieved from Medizino: https://medizinio.de/en/medical-equipment/ultrasound

Mesquita, D. (2021). *Python AI: How to Build a Neural Network & Make Predictions*. Retrieved from Real Python: https://realpython.com/python-ai-neural-network/

Nall, R. (2018). *Auscultation*. Retrieved from healthline: https://www.healthline.com/health/auscultation#why-its-important

Nilanon, T., Yao, J., Hao, J., Purushotham, S., & Liu, Y. (2016). Normal / Abnormal Heart Sound Recordings Classification.

Norreel, J.-C. (2021). *What are the Pros and Cons of Digital Stethoscopes*. Retrieved from Digital Health Central: https://digitalhealthcentral.com/2021/03/18/pros-and-cons-of-digital-stethoscopes/

Oliveira, J., Renna, F., Costa, P. D., Nogueira, M., Oliveira, C., & Ferreira, C. (2022). The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification.

Optuna. (2022). *Optuna: A hyperparameter optimization framework*. Retrieved from Optuna: https://optuna.readthedocs.io/en/stable/

Ortiz, J. J., Phoo, C. P., & Wiens, J. (2016). Heart Sound Classification Based on Temporal Alignment Techniques.

Pai, A. (2020). *CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

PhysioNet. (2022). *Challenges*. Retrieved from PhysioNet: https://physionet.org/about/challenge/

Poole, M. &. (1998). What Is Computational Intelligence? In *Computational Intelligence* (p. 1). Retrieved from Computational Intelligence

Poornima, A., & Savithaa, N. (2021). Classification of Pathological and Non Pathological.

Potes, C., Parvaneh, S., Rahman, A., & Conroy, B. (2016). Ensemble of Feature-based and Deep learning-based Classifiers for Detection of.

Pulse Uniform. (2020). *The 2021 Ultimate Guide to Different Types of Stethoscopes*. Retrieved from Pulse Uniform - Medical Nursing scrubs: https://www.pulseuniform.com/coffee-time/types-of-stethoscopes/

Python. (2022). *The Python Tutorial*. Retrieved from Python: https://docs.python.org/3/tutorial/index.html

Ray, S. (2017). *Commonly used Machine Learning Algorithms (with Python and R Codes)*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

Reyna, M. A., Elola, A., Oliveira, J., Renna, F., Gu, A., Sadr, N., . . . Clifford, G. D. (2022). *Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022*. Retrieved from Moody PhysioNet Challenge: https://moody-challenge.physionet.org/2022/

Roberts, L. (2020). *Understanding the Mel Spectrogram*. Retrieved from Analytics Vidhya: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

Rogel-Salazar, J. (2022). *Tensorflow, PyTorch or Keras for Deep Learning*. Retrieved from Domino Data Logo: https://blog.dominodatalab.com/tensorflow-pytorch-or-keras-for-deep-learning

Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I., & Sricharan, K. (2016). Classifying Heart Sound Recordings using Deep Convolutional Neural.

Saaty, T. (2008). *Decision making with the analytic hierarchy process*. Retrieved from http://www.rafikulislam.com/uploads/resourses/197245512559a37aadea6d.pdf

SAS Insights. (2022). *Artificial Intelligence What it is and why it matters*. Retrieved from SAS Insights: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html#history

Sayad, S. (2022). *Model Evaluation*. Retrieved from An Introduction to Data Science: https://www.saedsayad.com/model_evaluation.htm

Sayad, S. (2022). *Model Evaluation - Classification*. Retrieved from An Introduction to Data Science: https://www.saedsayad.com/model_evaluation_c.htm

Sayantini. (2021). *Keras vs TensorFlow vs PyTorch : Comparison of the Deep Learning Frameworks*. Retrieved from edureka!: https://www.edureka.co/blog/keras-vs-tensorflow-vs-pytorch/#introduction

scikit-learn. (2022). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/

Sharma, S. (2017). *Activation Functions in Neural Networks*. Retrieved from Towards Data Science: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Singh, T. (2019). Retrieved from MFCC's Made Easy: https://medium.com/@tanveer9812/mfccs-made-easy-7ef383006040

Singh-Miller, N. E., & Singh-Miller, N. (2016). Using Spectral Acoustic Features to Identify Abnormal Heart Sounds.

TensorFlow. (2022). *About TensorFlow*. Retrieved from TensorFlow: https://www.tensorflow.org/about

TensorFlow. (2022). *Classification on imbalanced data*. Retrieved from TensorFlow: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data

Thompson, W. R., Reinisch, A. J., Unterberger, M. J., & Schriefl, A. J. (2018). Artificial Intelligence-Assisted Auscultation of Heart Murmurs: Validation by Virtual Clinical Trial.

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

Tschannen, M., Kramer, T., Marti, G., Heinzmann, M., & Wiatowski, T. (2016). Heart Sound Classification Using Deep Structured Features.

Vernekar, S., Nair, S., Vijaysenan, D., & Ranjan, R. (2016). A Novel Approach for Classification of Normal/Abnormal Phonocardiogram Recordings using Temporal Signal Analysis and Machine Learning.

Wang, J., You, T., Yi, K., Gong, Y., Xie, Q., Qu, F., . . . He, Z. (2020). Intelligent Diagnosis of Heart Murmurs in Children with. *Journal of Healthcare Engineering*.

Wapcaplet. (2006). Retrieved from Wikipedia: https://pt.m.wikipedia.org/wiki/Ficheiro:Diagram_of_the_human_heart_(cropped).svg

WHO. (2020). *The top 10 causes of death*. Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

WHO. (2021). *Cardiovascular diseases*. Retrieved from World Health Organization: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

WHO. (2022). *Cardiovascular diseases*. Retrieved from World Health Organization: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

Wolfewicz, A. (2021). *Deep learning vs. machine learning – What's the difference?* Retrieved from Levity: https://levity.ai/blog/difference-machine-learning-deep-learning

Yaseen, Son, G.-Y., & Kwon, S. (2018). Classification of Heart Sound Signal Using.

Zabihi, M., Rad, A. B., Kiranyaz, S., Gabbouj, M., & Katsaggelos, A. K. (2016). Heart Sound Anomaly and Quality Detection using Ensemble of Neural Networks without Segmentation.

Zhiming, L., & Sheng, M. (2021). Multi-label classification of heart sound signals.

Attachments

Value Analysis

In this section it is presented the value analysis of the system to be created, it consists in the application of the new concept development model, value proposition and function analysis system technique.

The value analysis in a project is important to pinpoint areas that need attention and improvement, it provides a means for evaluating alternatives and documents the rationale behind recommendations and decisions.

New Concept Development Model

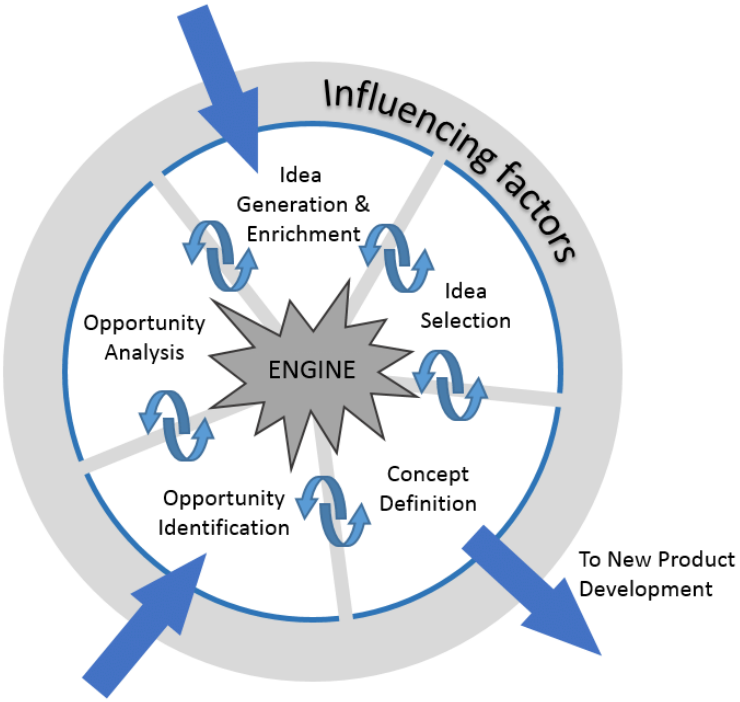

Figure 49 – The New Concept Development model (Martikainen, 2017)

Opportunity Identification

Cardiovascular diseases are currently the leading causes of death in the world as it can be seen in the graph of Figure 50, ischaemic heart disease and stroke alone represent 32% (17.9 million) out of 55.4 million disease related deaths worldwide. Since 2000, the largest increase in deaths has been for ischaemic heart disease, rising by more than 2 million to 8.9 million deaths in 2019 (WHO, 2020).

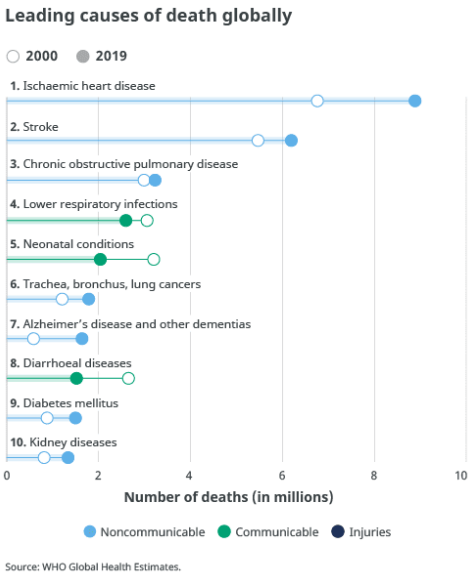

Figure 50 – Leading causes of death globally (WHO, 2020)

The World Bank classifies the world's economies into four income groups based on gross national income and in all the groups CVDs are in the top 3 causes of death (WHO, 2020), this makes the prevention and diagnosis of an extreme importance, to start an early planning and treatment of the patient (WHO, 2020).

People with CVD are faced with several medical complications that force them to adjust their lifestyle to the disease, like arrhythmias, heart failure, and pulmonary hypertension. This illness is usually accompanied by psychological challenges related to lack of normality, social integration, body image, disclosure, uncertainty, dependence, and coping. Several studies have shown that people with CVD may experience psychological distress associated with feelings of persistent insecurity, depression, anxiety, and low self-esteem (Kim, Johnson, & Sawatzky, 2019).

There is also an economic challenge to health care systems in the EU that is expected to grow in future years. The most recent data estimate that CVD costs the EU economy approximately €210 billion a year. Of that cost, around 53% (€111 billion) is for health care costs, 26% (€54 billion) is due to productivity losses and 21% (€45 billion) due to informal care of people with CVD (EHN, 2022).

Currently the main methods for cardiovascular diseases diagnosis are the electrocardiogram and the ultrasound, this one only used if necessary. The machine for the ECG costs from €500 to €15.000 (Medizinio, 2022), and for ultrasound machines the prices are higher, from €15.000 to €30.000 (Medizino, 2022).

Auscultation, despite being an economical method, is a complex method that depends, predominantly, on the experience and knowledge of the doctor and his hearing ability. A diagnosis made by an experienced cardiologist can have an accuracy rate of around 80%,

while a student or doctor at the beginning of their career has an accuracy rate between 20%-40% (Torres, 2021).

Opportunity Analysis

In the previous section we analysed the effects of cardiovascular diseases in the world regarding death rate, life disabilities, economy in the healthcare system and population. Based on the critical situation presented, in the poor health of the population and economic burden, we can conclude that the processes of prevention, diagnosis and treatment of CVDs need to be improved.

The suggested system focuses in accelerating the prevention and diagnosis phase, and at the same time reducing the costs for the healthcare system. This will benefit the patients, as they will have an early diagnosis increasing the speed of action for treatment, and the healthcare professionals by having a decision support system at their side.

Analytic Hierarchy Process

In this project we can apply the Analytic Hierarchy Process (AHP) to the selection of the DL framework.

First the decision tree must be created, it presents the structure of the problem, with the final objective, the criteria, and the alternatives in question. The decision tree for the selection of the DL framework follows:

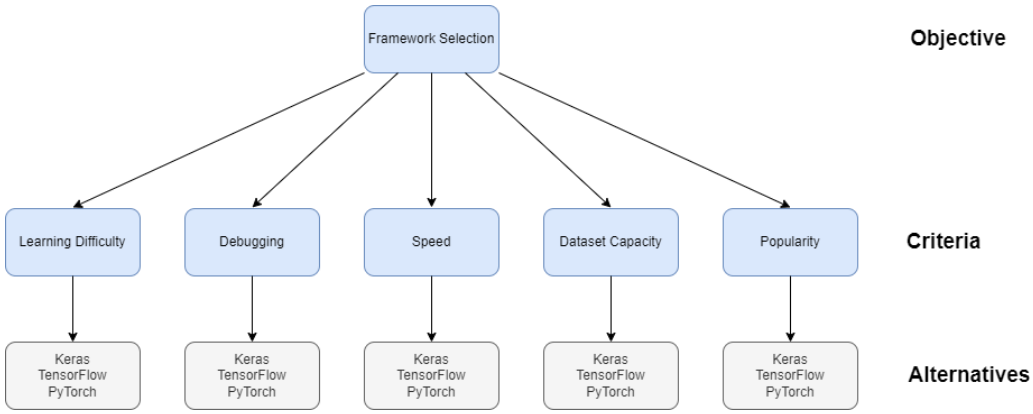

Figure 51 – Decision Tree

Next the levels of comparison between the different criteria defined in the previous step. For this, the fundamental scale of Saaty (Saaty, 2008) was used, which defines the following values:

Table 44 – Saaty's scale (Saaty, 2008)

Value	Definition	Explanation
1	Same importance	The two activities contribute equally to the objective
3	Moderate importance	Experience and judgment slightly favour one activity over another
5	Strong importance	Experience and judgement strongly favour one activity over another
7	Very strong importance	An activity is favoured very strongly over another
9	Extreme importance	The evidence favouring one activity over another is of the highest degree possible for affirmation
2, 4, 6, 8	Values in between	

By placing the criteria in a 5x5 table it is possible to give a value to each relationship between them:

Table 45 – Criteria comparison

	Learning Difficulty	Debugging	Speed	Dataset Capacity	Popularity
Learning Difficulty	1	4	2	5	1
Debugging	1/4	1	1/3	2	1/4
Speed	1/2	3	1	3	1/2
Dataset Capacity	1/5	1/2	1/3	1	1/5
Popularity	1	4	2	5	1

Next the criteria weights need to be calculated, the following table shows the results.

Table 46 – Calculation of the criteria weights

	Learning Difficulty	Debugging	Speed	Dataset Capacity	Popularity	Weight
Learning Difficulty	0,34	0,32	0,35	0,31	0,34	0,33
Debugging	0,08	0,08	0,06	0,13	0,08	0,09
Speed	0,17	0,24	0,18	0,19	0,17	0,19
Dataset Capacity	0,17	0,24	0,06	0,06	0,17	0,06
Popularity	0,34	0,32	0,35	0,31	0,34	0,33

To check if these weights are consistent, the Consistency Ratio (CR) was calculated resulting in a value of 0,013. Since we obtained a RC less than 0,1 we can conclude that the weights are consistent, which means we can continue with the decision process. The next step is the construction of the parity comparison matrix for each criterion.

Table 47 – Learning difficulty comparison matrix

Learning Difficulty	Keras	TensorFlow	PyTorch	Weight
Keras	1	5	7	0,70
TensorFlow	1/5	1	5	0,23
PyTorch	1/7	1/5	1	0,07

Table 48 – Debugging comparison matrix

Debugging	Keras	TensorFlow	PyTorch	Weight
Keras	1	3	1/3	0,26
TensorFlow	1/3	1	1/5	0,11
PyTorch	3	5	1	0,63

Table 49 – Speed comparison matrix

Speed	Keras	TensorFlow	PyTorch	Weight
Keras	1	1/5	1/5	0,09
TensorFlow	5	1	1	0,45
PyTorch	5	1	1	0,45

Table 50 – Dataset capacity comparison matrix

Dataset Capacty	Keras	TensorFlow	PyTorch	Weight
Keras	1	1/3	1/3	0,14
TensorFlow	3	1	1	0,43
PyTorch	3	1	1	0,43

Table 51 – Popularity comparison matrix.

Popularity	Keras	TensorFlow	PyTorch	Weight
Keras	1	3	5	0,63
TensorFlow	1/3	1	3	0,26
PyTorch	1/5	1/3	1	0,11

Finally, we calculate the composite priority for the alternatives:

Keras = (0,70 x 0,33) + (0,26 x 0,09) + (0,09 x 0,19) + (0,14 x 0,06) + (0,63 x 0,33) = 0.49

TensorFlow = (0,23 x 0,33) + (0,11 x 0,09) + (0,45 x 0,19) + (0,43 x 0,06) + (0,26 x 0,33) = 0.28

PyTorch = (0,07 x 0,33) + (0,63 x 0,09) + (0,45 x 0,19) + (0,43 x 0,06) + (0,11 x 0,33) = 0.23

According to the AHP the DL framework suggested is Keras with the highest score of 0,49.

Concept Definition

The system must be able of receiving the recorded sounds of the patient's heart in four auscultation locations. Extract features of the sound files and apply them in a model to make classifications regarding some characteristics of the heart sounds. These characteristics are, for example, presence of pathology, murmur location, murmur quality, murmur shape. Finally, it should retrieve to the user the information about the patient's heart.

The core of the system will be a convolutional neural network, its inputs will be four audio files each being the recorded sound in each auscultation location.

Value proposition

The Value Proposition Canvas is a tool to help understand the product or service and if it is positioned around what the customer values and needs are. In this project the customers are the health professionals.

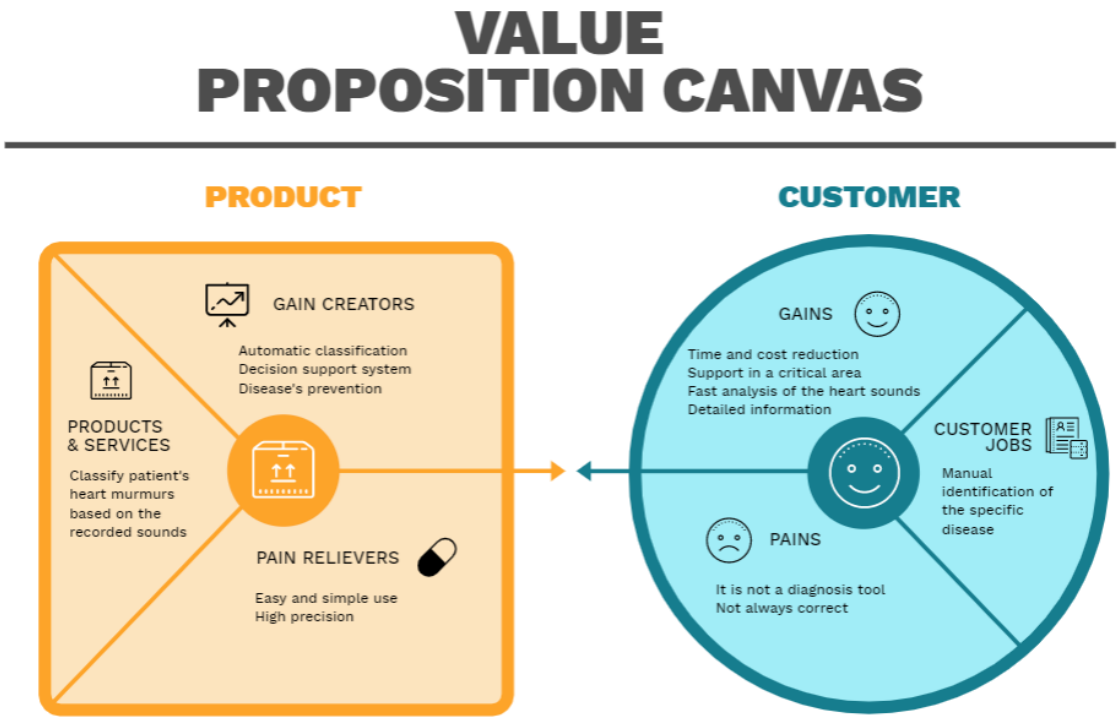

Figure 52 – Value Proposition Canvas

In the value proposition presented in Figure 52 we see that the product is a system that classifies the patient's heart murmurs based on the recorded sounds in the four auscultation locations (Aortic, Pulmonary, Tricuspid and Mitral regions). It can aid health professionals in 2 ways: i) when examining the patient's heart it supports them with detailed information, allowing them to make a more accurate diagnosis of cardiovascular diseases, and ii) in the prevention of diseases, as a first control option. The system makes an automatic classification of the murmurs and acts as a decision support system in the CVDs prevention and diagnosis. It is easy to use and has an elevated precision on the predictions, reducing time and cost to a critical medical area, it provides detailed information through a fast analysis. On itself it is not a diagnosis system, the expertise of a health professional is always needed to confirm the result and identify the specific condition.

Function Analysis System Technique

The Function Analysis System Technique (FAST) is a technique that aids in thinking about the problem objectively and in identifying the scope of the project by showing the logical relationships between steps.

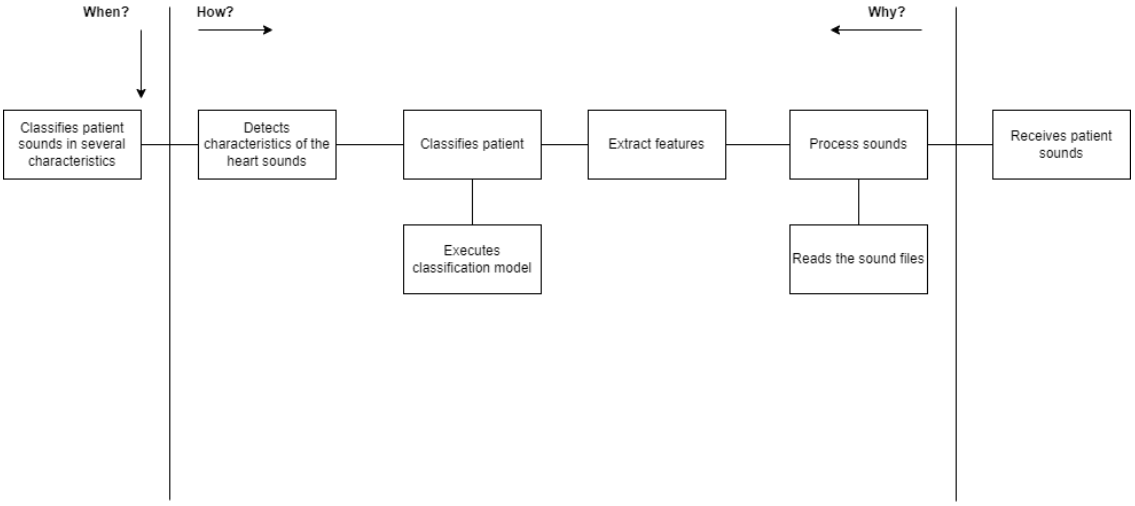

Figure 53 – FAST diagram

Looking at the FAST in Figure 53 we can see that the system is used to classify the patient sounds regarding several characteristics. It does that by receiving the patient sounds, doing an initial processing of them, extracts the features of each sound and classifies the patient by executing a classification model, which allows the system to detect the characteristics of the patient heart sounds.