



## **Deteção de patologia cardíaca utilizando aprendizagem profunda**

**HUGO FILIPE PADRÃO BRANDÃO ESTEVES**

Outubro de 2023

# **Deteção de patologia cardíaca utilizando aprendizagem profunda**

**Hugo Filipe Padrão Brandão Esteves**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Elsa Gomes**

**Co-orientador: Marcelo Nogueira**

**Júri:**

Presidente:

Vogais:

Porto, outubro 2023



# Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 14 de outubro de 2023

Hugo Esteves



# Resumo

As doenças cardiovasculares representaram 33% dos óbitos registados em todo o mundo no ano de 2019. Esta situação, juntamente com a ausência de sintomas evidentes, torna o diagnóstico precoce dessas condições consideravelmente mais complexo do que o habitual. Assim, anualmente são conduzidos vários estudos com o objetivo eficaz de detetar essas doenças antes que se tornem problemáticas.

Esta dissertação tem como propósito avaliar a capacidade de construir um modelo de aprendizagem profunda capaz de classificar os sons cardíacos de um paciente como normais ou anormais. Para alcançar este objetivo, utilizou-se o conjunto de dados do concurso George B. Moody PhysioNet Challenge para treinar modelos para a resolução do problema em questão.

Foram propostas duas abordagens distintas que se distinguem pela forma de construção dos modelos a serem utilizados. Na primeira abordagem, é criado um modelo de classificação para cada ponto de auscultação do coração, sendo a classificação final determinada com base em vários modelos. Já a segunda abordagem propõe a construção de um único modelo que recebe um segmento de áudio de cada ponto de auscultação e fornece uma classificação final do estado do paciente.

Em ambas as abordagens, foram aplicadas diferentes segmentações às amostras de áudio utilizadas, foram extraídas características de espectrogramas de mel e MFCC, e foram utilizados diversos algoritmos de aprendizagem foram empregues, nomeadamente algoritmos de aprendizagem profunda.

A melhor performance foi alcançada com a aplicação da primeira abordagem, obtendo um overall accuracy de 77.36% e um F1-Score de 62.22%. Este estudo acaba por fundamentar um ponto de progresso na resolução deste tipo de problema, demonstrando a viabilidade da utilização de aprendizagem profunda no seu percurso.

**Palavras-chave:** Aprendizagem Profunda, CNN, doenças cardiovasculares, modelo pré-treinado.



# Abstract

Cardiovascular diseases accounted for 33% of the deaths recorded worldwide in 2019. This, coupled with the absence of evident symptoms, makes early diagnosis of these conditions considerably more complex than usual. Consequently, numerous studies are conducted annually with the effective aim of detecting these diseases before they become problematic.

This dissertation aims to assess the ability to construct a deep learning model capable of classifying a patient's cardiac sounds as normal or abnormal. To achieve this goal, the dataset from the George B. Moody PhysioNet Challenge was utilized to train models for addressing the stated problem.

Two distinct approaches were proposed, differing in the manner of constructing the models to be employed. In the first approach, a classification model is created for each cardiac auscultation point, and the final classification is determined based on various models. Conversely, the second approach suggests building a single model that takes an audio segment from each auscultation point and provides a final classification of the patient's condition.

In both approaches, different segmentations of the audio samples were applied, features were extracted for both mel spectrograms and MFCCs, and various learning algorithms were employed, including deep learning algorithms.

The best performance was achieved by applying the first approach, yielding an overall accuracy of 77.36% and an F1-Score of 62.22%. This study ultimately substantiates a point of progress in addressing this type of problem, demonstrating the viability of employing deep learning in its course.

**Keywords:** Deep Learning, CNN, Cardiovascular diseases, pre-trained model.



# Agradecimentos

Para começar, agradeço aos meus orientadores Elsa Gomas e Marcelo Nogueira, por toda a disponibilidade e ajuda prestada durante o desenvolvimento do meu trabalho.

De seguida, agradeço aos meus pais e família, por todo o apoio incondicional que me deram ao longo dos anos e que me permitiram hoje chegar aqui e, por serem verdadeiros exemplos de como devo encarar a vida daqui em diante.

Seguidamente, agradeço a todos os meus amigos que se mantiveram ao meu lado durante estes anos e por me ajudarem a ser melhor e atingir os meus objetivos. Gostava de deixar um especial obrigado aos meus amigos Gonçalo Soares e Miguel Preto que me acompanharam durante a realização de todo o mestrado e tornaram os meus dias menos bons em dias fenomenais.

Por fim, mas não menos importante, agradeço à minha namorada Mestra de Medicina Tânia Fernandes, por todo o apoio e por acreditar em mim ao longo deste percurso. Certamente foi um pilar de esperança e motivação e serviu de exemplo para a minha disciplina, o que me ajudou a manter o foco para atingir os meus objetivos. #Hânia



# Índice

<b>1</b>	<b>Introdução</b> .....	<b>1</b>
1.1	Enquadramento .....	1
1.2	Problema.....	2
1.3	Objetivos.....	2
1.4	Abordagem .....	2
1.5	Estrutura do Documento .....	3
<b>2</b>	<b>Enquadramento Teórico</b> .....	<b>6</b>
2.1	Coração Humano.....	6
2.2	Sons Cardíacos .....	7
2.3	Doenças Cardiovasculares e sons anormais .....	8
2.4	Pontos de Auscultação.....	9
<b>3</b>	<b>Estado da Arte</b> .....	<b>11</b>
3.1	Inteligência Artificial .....	11
3.2	Aprendizagem Automática.....	12
3.3	Algoritmos de Aprendizagem Automática.....	13
3.3.1	Support Vector Machine (SVM) .....	13
3.3.2	Random Forest.....	14
3.3.3	Redes Neurais.....	14
3.4	Aprendizagem Profunda.....	16
3.4.1	Redes Neurais Convolucionais (CNN).....	17
3.4.2	Rede Neuronal Recorrente (RNN) .....	19
3.5	Extração de características de Áudio .....	20
3.5.1	Fast Fourier Transform (FFT) .....	20
3.5.2	Espectrograma de Mel.....	21
3.5.3	Mel Frequency Cepstral Coefficients (MFCC).....	21
3.6	Métodos de Avaliação.....	22
3.6.1	Holdout.....	23
3.6.2	K-Fold Cross Validation.....	23
3.7	Tecnologias.....	24
3.7.1	Python.....	24
3.7.2	TensorFlow/Keras .....	25
3.7.3	PyTorch .....	25
3.7.4	Caffe.....	26
3.8	Trabalhos Relacionados .....	26
<b>4</b>	<b>Análise de Valor</b> .....	<b>29</b>

4.1	Processo de Inovação .....	29
4.1.1	Identificação de Oportunidade.....	30
4.1.2	Análise da Oportunidade .....	31
4.2	Valor da solução .....	31
4.2.1	Proposta de valor.....	32
4.3	Técnica Sistemática de Análise Funcional .....	32
4.4	Método de Análise Hierárquico.....	33
<b>5</b>	<b>Design da Solução .....</b>	<b>38</b>
5.1	Linguagem e Framework de aprendizagem profunda .....	38
5.2	Dataset disponível.....	39
5.3	Abordagem.....	40
5.3.1	Abordagem 1 .....	40
5.3.2	Abordagem 2 .....	41
5.4	Pré-processamento dos dados.....	41
5.4.1	Segmentação dos áudios .....	42
5.5	Extração de atributos.....	42
5.6	Partição dos dados .....	43
5.7	Alternativas de Design.....	44
5.7.1	Alternativa 1 - Proposta Legado .....	44
5.7.2	Alternativa 1 - Nova Proposta.....	45
5.7.3	Alternativa 2 - Proposta Legado .....	46
<b>6</b>	<b>Implementação.....</b>	<b>48</b>
6.1	Pré-processamento.....	48
6.1.1	Remoção de Ruído.....	48
6.1.2	Segmentação dos áudios .....	49
6.2	Extração de Atributos.....	52
6.2.1	Espectrograma de Mel .....	52
6.2.2	MFCC.....	53
6.3	Alternativa de Implementação .....	54
6.3.1	Construção de Alternativas.....	54
6.3.2	Alternativa 1 - Nova Proposta.....	57
6.3.3	Alternativa 2 - Proposta Legado .....	63
<b>7</b>	<b>Avaliação .....</b>	<b>66</b>
7.1	Hipótese.....	66
7.2	Medidas de Avaliação .....	67
7.2.1	Matriz de Confusão.....	67
7.2.2	Medidas de desempenho .....	68
7.3	Avaliação do Sistema .....	69
7.4	Experiências Realizadas .....	70

7.4.1	Alternativa 1 - SVM.....	70
7.4.2	Alternativa 1 - Random Forest .....	71
7.4.3	Alternativa 1 - Rede Neuronal Convolutacional .....	72
7.4.4	Alternativa 1 - Modelo Pré-Treinado.....	73
7.4.5	Alternativa 2 .....	74
7.5	Comparação e discussão de Resultados .....	75
<b>8</b>	<b>Conclusão .....</b>	<b>77</b>
8.1	Trabalho Futuro .....	78



# Lista de Figuras

Figura 1 - Funcionamento do coração humano (Ribeira, 2022) .....	6
Figura 2 - Som normal produzido pelo coração (Noor, 2019) .....	8
Figura 3 – Sopros produzidos pelo coração (Noor, 2019) .....	9
Figura 4 - Pontos de auscultação (Karnath & Thornton, 2002) .....	9
Figura 5 - Conceitos associados a inteligência artificial (M, 2023) .....	11
Figura 6 - Algoritmo de classificação e regressão (Bindra, 2023) .....	13
Figura 7 - Rede neuronal artificial (Castrounis, 2022).....	15
Figura 8 - Aprendizagem automática vs aprendizagem profunda (DLT Labs, 2020) .....	16
Figura 9 - Diferentes camadas de rede neuronal convolucional (Vargas, et al., 2016) .....	17
Figura 10 - Camada de convolução (Pokhrel, 2019) .....	18
Figura 11 - Métodos de pooling mais comuns (Pokhrel, 2019) .....	18
Figura 12 - Fase de classificação de uma rede convolucional (Pokhrel, 2019).....	19
Figura 13 - Exemplo de rede neuronal recorrente (Roell, 2017) .....	19
Figura 14 - Aplicação da transformação de Fourier (NTI Audio, 2023) .....	20
Figura 15 – Exemplo de espectrograma de mel (Roberts, 2020) .....	21
Figura 16 - Exemplo de MFCC (Nair, 2018) .....	22
Figura 17 – Diagrama de passos de computação de MFCC (Bezoui, et al., 2017) .....	22
Figura 18 - Método de avaliação holdout (Devi, 2021).....	23
Figura 19 - K-Fold Cross Validation (Japkowicz, 2023).....	23
Figura 20 - Quantidade - Linguagens mais populares na plataforma StackOverflow (Gupta, 2021) .....	24
Figura 21 - Modelo de desenvolvimento de novo conceito (Koen, et al., 2002).....	29
Figura 22 – Causas de morte no ano de 2019 (Ritchie & Roser, 2019).....	30
Figura 23 – Proposição de valor.....	32
Figura 24 - Diagrama FAST .....	33
Figura 25 - Árvore hierárquica de decisão da melhor framework para construção do algoritmo .....	34
Figura 26 - Árvore de decisão de escolha de framework com prioridades relativas.....	37
Figura 27 - Abordagem de desenvolvimento 1 .....	41
Figura 28 - Abordagem de desenvolvimento 2 .....	41
Figura 29 - Arquitetura do sistema alternativa 1 – proposta legado.....	45
Figura 30 - Arquitetura do sistema alternativa 1 – nova proposta.....	46
Figura 31 - Arquitetura do sistema alternativa 2 - proposta legado .....	46
Figura 32 – Exemplo de onda de áudio normal e onda com aplicação de filtro .....	49
Figura 33 - Comparação das ondas .....	49
Figura 34 - Segmentação dos ficheiros de áudio .....	50
Figura 35 - Leitura de ficheiro ".tsv" .....	51
Figura 36 - Criação do espectrograma de mel.....	52
Figura 37 - Exemplo de espectrograma de mel gerado .....	53
Figura 38 - Criação do MFCC .....	53

Figura 39 - Exemplo de MFCC.....	54
Figura 40 - Exemplo de partição da informação de cada foco .....	55
Figura 41 - Exemplo de construção de modelo CNN .....	55
Figura 42 - Exemplo de função de treino .....	56
Figura 43 - Matriz de confusão e métricas de avaliação dos modelos .....	57
Figura 44 - Criação de matriz de confusão .....	57
Figura 45 - Métricas de desempenho de modelo.....	58
Figura 46 - Classificação do paciente recorrendo a 4 modelos .....	58
Figura 47 - Função de treino de abordagem SVM .....	59
Figura 48 - Exemplo de utilização de GridSearchCV .....	59
Figura 49 - Exemplo de utilização de RandomSearchCV.....	60
Figura 50 - Modelo RCN para cada foco.....	61
Figura 51 - Construção de CNN.....	62
Figura 52 - Modelo Pré-Treinado para cada foco .....	63
Figura 53 - Modelo CNN com 4 inputs .....	64
Figura 54 - Construção de CNN com 4 focos .....	64
Figura 55 - Modelo CNN com 4 inputs com auxílio de modelo pré-treinado .....	65
Figura 56 - Escala fundamental – Níveis de importância de comparações (Saaty, 1990).....	87



# Lista de Tabelas

Tabela 1 - Vantagens e desvantagens da ferramenta TensorFlow (Geeks for Geeks, 2023).....	25
Tabela 2 - Vantagens e desvantagens da ferramenta PyTorch (Educba, 2023) .....	26
Tabela 3 - Vantagens e desvantagens da ferramenta Caffe (Ísbilen, 2022).....	26
Tabela 4 – Submissões de concurso George B. Moody PhysioNet Challenge 2022 .....	27
Tabela 5 - Submissões de soluções que utilizam o <i>dataset</i> do concurso George B. Moody PhysioNet Challenge 2022.....	28
Tabela 6 - Comparação dos diferentes critérios.....	34
Tabela 7 - Normalização dos valores e respectivas prioridades relativas .....	35
Tabela 8 - Propriedades relativas de facilidade de utilização .....	35
Tabela 9 - Propriedades relativas de comunidade e suporte.....	36
Tabela 10 - Propriedades relativas de desempenho .....	36
Tabela 11 - Propriedades relativas de dimensão de <i>datasets</i> .....	36
Tabela 12 - Informação disponível no <i>dataset</i> - 1 (Reyna, et al., 2022) (Oliveira, et al., 2022) .	39
Tabela 13 - Informação disponível no <i>dataset</i> – 2 (Reyna, et al., 2022) (Oliveira, et al., 2022)	40
Tabela 14 – Exemplo de ficheiro “.tsv” .....	50
Tabela 15 - Distribuição das amostras de classes por abordagem de segmentação.....	51
Tabela 16 - Parâmetros utilizados no algoritmo Random Forest .....	60
Tabela 17 - Matriz de confusão .....	67
Tabela 18 – Comparação de diferentes abordagens nos modelos SVM.....	71
Tabela 19 - Comparação de diferentes abordagens nos modelos Random Forest.....	72
Tabela 20 - Comparação de diferentes abordagens nos modelos CNN .....	73
Tabela 21 - Comparação de diferentes abordagens nos modelos Pré-treinados .....	74
Tabela 22 - Comparação de diferentes abordagens na alternativa 2.....	74
Tabela 23 - Melhor modelo de cada experiência realizada .....	75
Tabela 24 - Comparação da solução com a literatura .....	76



# Acrónimos e Símbolos

## Lista de Acrónimos

<b>AA</b>	<i>Aprendizagem Automática</i>
<b>AHP</b>	<i>Analytic Hierarhy Process</i>
<b>AI</b>	Inteligência Artificial
<b>ANN</b>	<i>Rede Neuronal Artificial</i>
<b>AP</b>	<i>Aprendizagem Profunda</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>CNN</b>	<i>Rede Neuronal Convolucional</i>
<b>CPU</b>	<i>Central Process Unit</i>
<b>FAST</b>	Function Analysis System Technique
<b>FFE</b>	<i>Fuzzy Front End</i>
<b>FN</b>	Falso Negativo
<b>FP</b>	<i>Falso Positivo</i>
<b>GPU</b>	<i>Graphics Processing Unit</i>
<b>MFCC</b>	Mel Frequency Cepstral Coefficients
<b>NPD</b>	<i>New Product Development</i>
<b>ReLU</b>	<i>Rectified Linear Unit</i>
<b>RNN</b>	<i>Rede Neuronal Recorrente</i>
<b>S1</b>	<i>First Heart Sound</i>
<b>S2</b>	<i>Second Heart Sound</i>
<b>TanH</b>	<i>Tangente hiperbólica</i>
<b>VN</b>	<i>Verdadeiro Negativo</i>
<b>VP</b>	<i>Verdadeiro Positivo</i>





# 1 Introdução

Nesta secção é introduzido de um modo geral o funcionamento do trabalho que foi desenvolvido durante a elaboração do documento. Começa pelo enquadramento e descrição do problema, bem como os objetivos que se pretende atingir com o projeto a desenvolver. Por fim, é descrita a abordagem utilizada para atingir os objetivos e é realizada uma breve descrição da estrutura do documento.

## 1.1 Enquadramento

A saúde é um dos bens mais importantes no quotidiano das pessoas. Através do investimento em áreas de pesquisa e de prevenção, é possível melhorar a qualidade de vida das pessoas. Para tal, todos os anos existe um grande investimento por parte de várias organizações a nível mundial de modo a conseguir evoluir esta área e conseguir trazer uma maior qualidade de vida para todos (Bloom, et al., 2011).

Após uma análise dos dados de 2019 fornecidos pela plataforma “Our World in Data” é possível verificar que durante o ano de 2019 houve 56 milhões de óbitos, dos quais 18.56 milhões (cerca de 33%) são devido a problemas cardiovasculares (Ritchie & Roser, 2019). Este tipo de doenças afeta não só o coração, mas como toda a rede de vasos sanguíneos existentes no corpo humano, sendo exemplos destas doenças a doença cerebrovascular, doença cardíaca reumática, doença congénita cardíaca, entre outras. Um dos principais motivos que dificulta o despiste deste tipo de doenças provem da falta de sintomas que as identifiquem antes de serem verdadeiramente problemáticas.

## **1.2 Problema**

Um dos principais objetivos do estudo de doenças cardiovasculares é conseguir descobrir métodos (Minneapolis Heart Institute Foundation, 2023) que permitam o mais rapidamente possível tratar e detetar alguma anomalia modo a conseguir tratar eventuais problemas antes que estes apresentem dimensões catastróficas.

Atualmente, o método mais relevante de despiste inicial de doenças cardiovasculares que afetam os músculos e válvulas do coração é a utilização de estetoscópios por parte de um profissional da área da saúde (Samuel, 2022). Apesar de ser um método económico para a deteção de um problema cardíaco, este incorre na possibilidade de erros humanos, como por exemplo o estado emocional do profissional de saúde ou mesmo da existência de ruído externo que impeça a compreensão dos sons auscultados.

Deste modo, existe a necessidade de promover a criação de uma solução que permita auxiliar o profissional de saúde a fazer um rastreio inicial de anomalia associados com o coração de modo a garantir uma melhor qualidade de vida para os pacientes.

## **1.3 Objetivos**

O objetivo da presente tese é estudar e desenvolver um modelo de aprendizagem profunda com base em redes neuronais que permita, através da análise de sons de batimentos cardíacos de diversos pacientes, obter uma resposta quanto à presença de anomalias nos sons produzidos. Este trabalho surge na continuação do trabalho apresentado numa dissertação anterior (Feliz, 2022), cujas propostas de solução desenvolvidas serviram de base ao trabalho aqui apresentado com o objetivo de serem aprimoradas de modo a obter resultados mais relevantes. O produto final deverá ser passível de ser utilizado por um profissional da área da saúde para conseguir detetar problemas cardíacos atempadamente a fim de melhorar a qualidade de vida do paciente.

Para além disto, pretende-se que o trabalho seja comparado com outras abordagens exploradas no estado da arte, de modo a validar o contributo da solução para o rastreio de doenças cardiovasculares.

## **1.4 Abordagem**

Numa fase inicial, é necessário obter conhecimento do domínio que envolve as doenças cardiovasculares. Para tal, é realizado um processo de estudo em que são aprofundados conceitos relacionados com o coração como a sua morfologia, sons normais e anormais e auscultação.

Numa segunda fase é necessário realizar um levantamento do estado da arte, onde serão explorados conceitos do âmbito da inteligência artificial, e as diversas tecnologias e

ferramentas existentes para a resolução do problema. Através do estudo de conceitos relacionados com inteligência artificial é então possível rever quais são as abordagens mais recorrentes para a resolução de problemas semelhantes, sendo então possível tomar decisões sobre algoritmos a utilizar e quais as ferramentas ideais para o desenvolvimento da solução.

Para conseguir compreender o valor do negócio foi efetuada uma análise de valor onde foram identificadas as principais oportunidades a ser exploradas derivadas da grande problemática dos problemas cardiovasculares. O método de tomada de decisão AHP foi utilizado para auxiliar na decisão da framework ideal para a resolução do problema.

Terminada a análise de valor, foi concretizada a análise do design ponderado e correspondente implementação para a resolução do problema. É importante retirar deste capítulo a estrutura do *dataset* a ser utilizado, as alternativas que são propostas de desenvolver e as arquiteturas ponderadas para cada uma delas. Com esta análise, é possível realizar o tratamento dos dados e construção de diversos modelos de classificação.

Por fim, é avaliada a solução obtida. Para tal, são recolhidas diversas métricas de desempenho permitindo assim não só realizar comparações entre diferentes alternativas como também com os trabalhos existentes na literatura.

## **1.5 Estrutura do Documento**

O documento a desenvolver encontra-se estruturado em 8 partes, sendo estas a Introdução, o Enquadramento Teórico, o Estado da Arte, a Análise de Valor, o Design da Solução, a Implementação, a Avaliação e por fim a Conclusão.

O capítulo da Introdução tem como o seu principal objetivo oferecer ao leitor a compreensão global do que foi abordado ao longo do documento. Aqui é realizada uma descrição inicial do problema que pretendemos resolver, os objetivos que estão propostos de se atingir com a elaboração do documento e outras informações que consigam de forma sucinta suscitar interesse na leitura do documento.

No segundo capítulo, o Enquadramento Teórico, são apresentadas todas as noções teóricas que são necessárias que o leitor compreenda de modo a ter conhecimento suficiente sobre o coração humano a fim de conseguir perceber o valor da proposta a desenvolver.

O terceiro capítulo é o Estado da Arte e tem como objetivo compreender os conceitos gerais de aprendizagem automática, bem como aprofundar conceitos sobre aprendizagem profunda que é o foco do projeto a desenvolver. Para além disto, são abordadas as diversas tecnologias com as quais foi desenvolvida a solução final, oferecendo comparação com outras já existentes de modo a comprovar a adequabilidade da utilização das mesmas.

No quarto capítulo, a Análise de Valor, é abordado o valor do negócio que se pretende desenvolver. É neste capítulo que é elaborada uma proposta de valor, o diagrama FAST e onde são comparadas as diversas tecnologias a utilizar através do método AHP.

No quinto capítulo são exploradas as várias opções de design estipuladas para responder ao problema que se pretende resolver, bem como as tecnologias que são utilizadas para a elaboração da solução.

No sexto capítulo, são exploradas as implementações realizadas das alternativas de design que são exploradas no capítulo anterior. Aqui é possível observar alguns excertos de código referentes a segmentos da implementação considerados relevantes, como também são apresentadas as arquiteturas utilizadas para a construção dos modelos de aprendizagem profunda.

No sétimo capítulo é identificada a hipótese proposta para a investigação a decorrer, sendo também explorados os vários métodos e métricas necessários para conseguir estudar e avaliar a fiabilidade da solução produzida. Para além disto, é ainda realizada a avaliação das várias implementações desenvolvidas, sendo efetuada uma comparação final com as soluções já existentes.

Por fim, no oitavo capítulo é realizada a conclusão do trabalho realizado culminando na reflexão do tema explorado, dos resultados obtido e ainda são analisadas possíveis ações a realizar em trabalho futuro.



## 2 Enquadramento Teórico

Nesta secção são explorados diversos conceitos relacionados com a área de doenças cardiovasculares. É realizado um estudo sobre o funcionamento do coração, os sons que o mesmo produz e o modo de como se consegue efetuar a deteção de problemas cardíacos.

### 2.1 Coração Humano

Começando pela compreensão de um dos órgãos do corpo humano, o coração é responsável por garantir que o sangue flua e atinja todas as partes do corpo e garante que a pressão arterial se mantém estável ao longo do tempo (clevelandclinic, 2023). O coração é composto por duas metades separadas por uma pequena parede muscular chamada septo, sendo ambas compostas por um aurículo e um ventrículo, tal como podemos verificar na Figura 1.

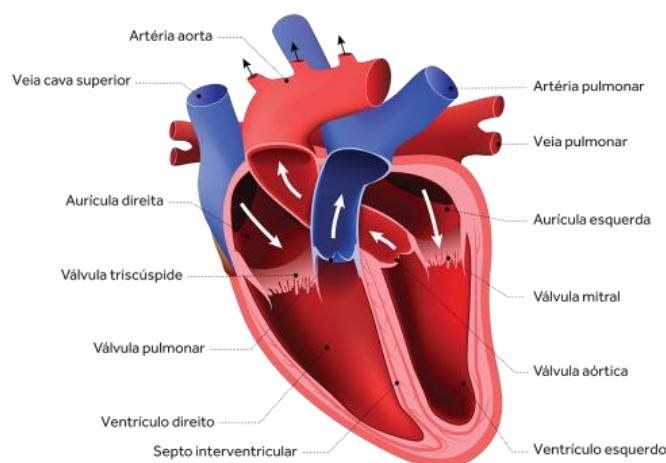


Figura 1 - Funcionamento do coração humano (Ribeira, 2022)

O processo de enviar sangue oxigenado para as diversas partes do corpo humano começa na veia cava superior onde o sangue venoso (sangue com baixo teor de oxigénio) entra na

aurícula direita através das veias cavas. De seguida, durante o relaxamento do coração (diástole) ocorre a passagem do sangue venoso para o ventrículo direito, que será posteriormente bombeado para os pulmões através da artéria pulmonar. Mais tarde, este sangue é transportado para a aurícula esquerda através das veias pulmonares. Este sangue, por sua vez, é denominado como sangue arterial visto conter um alto teor de oxigénio proveniente das trocas gasosas efetuadas nos pulmões. Por fim, volta a haver um relaxamento do coração havendo a passagem do sangue da aurícula esquerda para o ventrículo esquerdo, sendo este posteriormente bombeado para o resto do corpo através da artéria aorta. Este será então o procedimento normal do coração e o que será expectável de uma pessoa com um órgão saudável (Ribeira, 2022).

## 2.2 Sons Cardíacos

Os sons cardíacos produzidos durante os ciclos cardíacos referidos anteriormente são o resultado do fecho das válvulas presentes no coração ou das cordas tendinosas derivados do fluxo de sangue que entra e sai do coração. Devido à vibração produzida por estas estruturas, conseguimos então perceber o conhecido som de “lub-dub” utilizado para representar um batimento cardíaco (Dornbush & Turnquest, 2020).

O primeiro som que é produzido pelo coração, denominado de S1, resulta do fecho da válvula mitral que separa a aurícula esquerda do ventrículo esquerdo (também conhecido como M1) e da válvula tricúspide que separa a aurícula direita do ventrículo direito (também conhecido como T1).

O segundo som que é audível durante o ciclo cardíaco é denominado de S2. Tal som é produzido pelo fecho da válvula aórtica (conhecido com A2) e da válvula pulmonar (conhecido como P2) que ocorrem aquando do bombardeamento do sangue para os diversos órgãos do corpo e para os pulmões, respetivamente (Healio, 2017).

O conjunto dos sons S1 e S2, tal como apresentado na Figura 2, constituem o ciclo cardíaco completo, podendo ser utilizados para compreender a contração e o relaxamento do coração. O período compreendido entre o som S1 e o som S2 é designado de sístole, tendo este uma menor duração e corresponde ao período de contração do órgão. O período que decorre entre o som S2 e S1 é por sua vez o mais prolongado, sendo resultado do relaxamento do coração e da consequente entrada de sangue para as correspondentes aurículas.

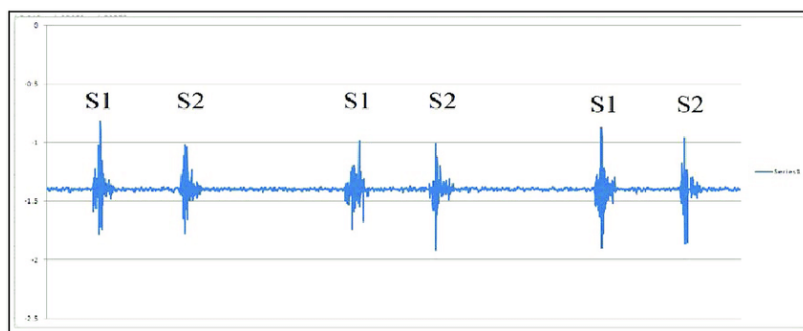


Figura 2 - Som normal produzido pelo coração (Noor, 2019)

## 2.3 Doenças Cardiovasculares e sons anormais

Tal como já referido anteriormente, as doenças cardiovasculares são as principais responsáveis pelo número de mortes que acontecem a nível mundial e caracterizam-se por problemas associados tanto ao coração ou aos vasos sanguíneos (World Health Organization, 2021).

Na secção anterior foram mencionados os sons S1 e S2 que ocorrem devido ao fecho das diferentes válvulas existentes no coração. Contudo, o coração poderá produzir alguns sons anormais durante o processo de ciclo cardíaco, como os denominados por sopro. Sopros podem ser algo não problemático quando ocorre em crianças devido ao fluxo normal do sangue na corrente sanguínea ou mesmo em jovens adultos devido a atividade física, febre ou mesmo gravidez (Healthline, 2019). Quando tal não se verifica, estes podem ser indicativos de uma doença cardiovascular, proveniente de malformações existentes nas cavidades do coração ou através do mal funcionamento das válvulas referidas na secção 2.1.

Para além dos sons S1 e S2 mencionados anteriormente, existem ainda mais dois possíveis sons produzidos pelo coração e que podem ou não estar associados a doenças cardiovasculares, tal como apresentado na Figura 3. Em primeiro lugar, os sons S3, também denominados de “galope ventricular”, que ocorrem após a diástole (S2) e são caracterizados pelo fluxo passivo de sangue das aurículas para os ventrículos (Healio, 2017). No caso de jovens atletas ou mulheres grávidas, estes sons podem não ser indicativos de que existe algum problema no coração. Passa possas idosas, este já se trata de um som anormal e é necessário recorrer a exames para compreender a gravidade da situação.

Em segundo lugar, existem os sons S4 também conhecidos como “galope arterial” acontece quando o ventrículo esquerdo não desenvolve corretamente, adotando assim uma fisiologia espessa e esponjosa ao invés do normal rijo e liso. O som ouvido durante o exame ocorre devido ao fluxo de sangue para os ventrículos referidos devido a contrações auriculares. Ao contrário do que acontece com os sons S3, este tipo de sons é sempre associado a problemas cardíacos sendo assim um fator de alarme quando presente (Healio, 2017).

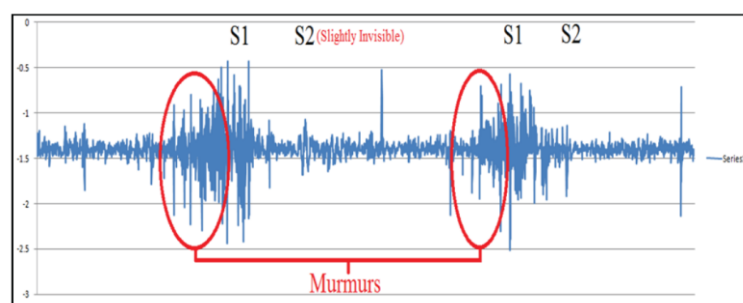


Figura 3 – Sopros produzidos pelo coração (Noor, 2019)

## 2.4 Pontos de Auscultação

Auscultação é um método pelo qual se torna possível ouvir os batimentos cardíacos através da utilização de um estetoscópio. É através dela que é possível verificar problemas associados ao coração através da análise de variados sons de frequências altas (derivadas da atividade sistólica) e frequências baixas (provenientes da atividade diastólica) (Karnath & Thornton, 2002).

Apesar de o estetoscópio ser bastante útil na compreensão e análise dos sons cardíacos, é necessário que este seja utilizado de forma correta de modo a conseguir tirar conclusões proveitosas da análise efetuada. Para tal, existem um conjunto de 4 pontos de auscultação que devem ser avaliados durante uma consulta de rotina. Em primeiro lugar o médico a efetuar o exame deverá validar o espaço intercostal direito (válvula aórtica) e o espaço intercostal esquerdo (válvula pulmonar). De seguida, deverá ser analisado a área da extremidade esquerda do esterno (válvula tricúspide) sendo por fim verificado o quinto espaço intercostal esquerdo (válvula mitral). Este processo, tal como visível na Figura 4, visa conseguir compreender o estado de funcionamento das diferentes válvulas existentes no coração e conseguir detetar assim patologias provenientes do mesmo.

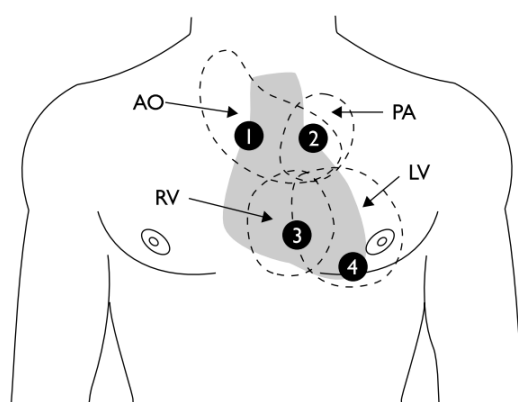


Figura 4 - Pontos de auscultação (Karnath & Thornton, 2002)



## 3 Estado da Arte

Nesta secção serão abordados conceitos teóricos sobre o estado da arte relacionado com a área de inteligência artificial. São explorados conceitos sobre diferentes tipos de algoritmos de aprendizagem profunda e quais as tecnologias existentes no mercado para os desenvolver, bem como as suas vantagens e desvantagens.

### 3.1 Inteligência Artificial

Inteligência artificial (AI) é um campo da área das ciências da computação fundado em 1950 que tem como objetivo dar ênfase à criação de máquinas inteligentes e que conseguem atingir comportamentos e ações à semelhança do ser humano. Devido à sua grande versatilidade e à capacidade de realizar tarefas específicas melhor do que qualquer ser humano, atualmente já se encontra disseminada nas mais diversas áreas do setor industrial, permitindo uma maior eficiência, qualidade e segurança do trabalho desenvolvido (Johns, 2021).

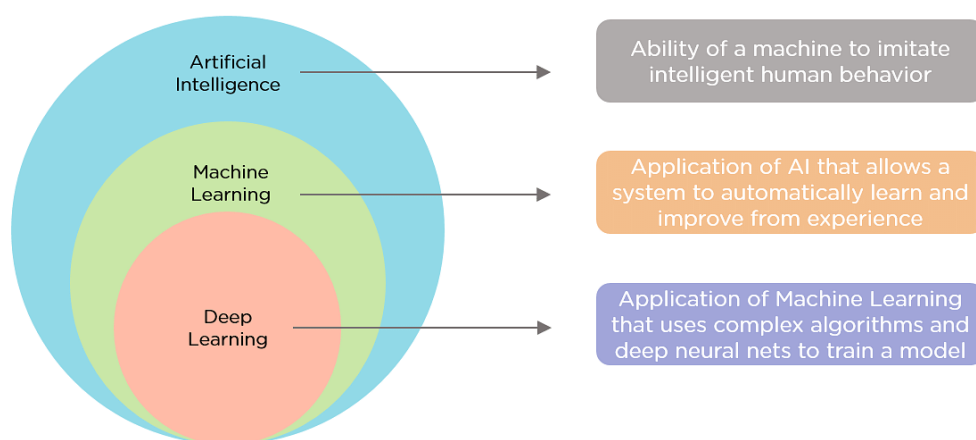


Figura 5 - Conceitos associados a inteligência artificial (M, 2023)

Tal como apresentado na Figura 5, aprendizagem automática é uma subárea de AI que auxilia na construção de aplicações capazes de realizar classificação e predições através de dados fornecidos. Por sua vez, aprendizagem profunda é ainda uma subárea de aprendizagem automática e que consegue a partir de grandes quantidades de informação e algoritmos complexos treinar um modelo para realizar uma determinada tarefa (M, 2023).

## 3.2 Aprendizagem Automática

Aprendizagem automática (AA) que tem como objetivo conseguir através de dados e algoritmos imitar o mecanismo de aprendizagem de um ser humano. Este método de aprendizagem é muito importante na área de ciência de dados sendo utilizado métodos estatísticos para a realização de classificações ou previsões (IBM, 2023). Os algoritmos de aprendizagem automática podem ser classificados segundo os seguintes 4 tipos (Bi, et al., 2019) (Ayodele, 2010):

- **Aprendizagem supervisionada:** A informação utilizada para a aprendizagem já se encontra rotulada podendo os resultados ser comparados com o previsto.
- **Aprendizagem não supervisionada:** A informação utilizada não se encontra rotulada, sendo necessário identificar as relações ou grupos existentes na informação sem se recorrer ao valor real.
- **Aprendizagem semi-supervisionada:** Um meio termo no que conta às abordagens anteriores, existindo apenas uma pequena porção da informação rotulada sendo a restante não rotulada.
- **Aprendizagem por reforço:** O algoritmo aprende a classificar de acordo com a observação dos resultados anteriores. Cada resultado resulta em feedback que guia o algoritmo a obter uma melhor classificação.

Ao entrar no domínio dos algoritmos de aprendizagem supervisionada é ainda possível realizar a segmentação entre algoritmos de classificação e algoritmos de regressão. A decisão de qual dos dois deverá ser utilizado depende do âmbito e propósito do problema a resolver sendo que algoritmos de classificação apresentam resultados sobre a forma de valores discretos (classes pré-definidas), enquanto algoritmos de regressão oferecem a resposta sob forma de um resultado contínuo, tal como apresentado na Figura 6 (Bi, et al., 2019).

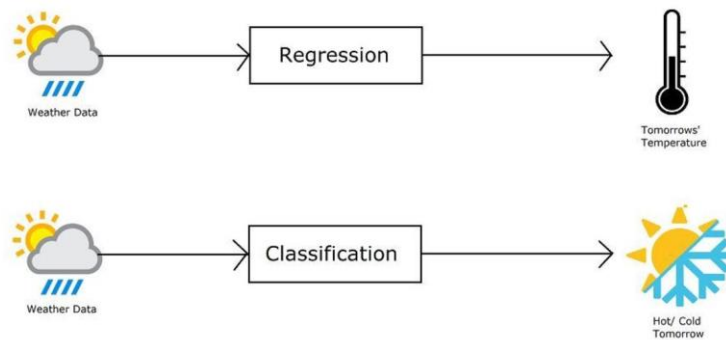


Figura 6 - Algoritmo de classificação e regressão (Bindra, 2023)

### 3.3 Algoritmos de Aprendizagem Automática

Algoritmos de aprendizagem automática são definidos pelo processo de aprendizagem realizado sobre um determinado conjunto de dados. Este processo culmina na criação de um modelo, sendo este a representação da informação que foi compreendida durante o processo de aprendizagem (Brownlee, 2016).

Nesta secção é realizada a análise de alguns algoritmos de classificação que foram relevantes para a resolução do problema e que permitiram a criação dos modelos que serão discutidos em capítulos posteriores.

#### 3.3.1 Support Vector Machine (SVM)

A support vector machine é um algoritmo de classificação linear dentro da área de aprendizagem supervisionada. Este tipo de algoritmo é ideal para a resolução de problemas que apresentam apenas duas classes de classificação sendo o seu objetivo descobrir o modo ideal de separar *hyperplanes* que classificam corretamente a informação recebida. Para tal, a construção destes planos é realizada de modo a separar as amostras de ambas as classes da melhor maneira possível, minimizando assim o risco de classificar incorretamente tanto amostras de treino como amostras de teste (Farquad & Bose, 2012).

Este é um processo de aprendizagem baseado em teoria de aprendizagem estatística, sendo uma das melhores técnicas de aprendizagem automática para a resolução de problemas relacionados com *data mining*. É também utilizada em diversas áreas, como a da medicina (detecção de cancro do colon e análise genética) ou na área financeira (previsão de séries temporais financeiras e detecção de fraudes) (Farquad & Bose, 2012).

### 3.3.2 Random Forest

O algoritmo de Random Forest é um método de aprendizagem supervisionada baseado na utilização de árvores de decisão. Este método angaria um conjunto de árvores de decisão que posteriormente serão utilizadas para a realização de uma classificação. Esta classificação será decidida pela maioria absoluta dos votos realizados por cada árvore. Para a criação de cada árvore são geralmente gerados vetores aleatórios que estão responsáveis por governar o crescimento de cada uma das árvores da floresta (Breiman, 2001).

A utilização deste algoritmo tem vindo a crescendo ao longo dos anos devido à sua capacidade natural de limitar o *overfitting* dos modelos criados, devido à sua flexibilidade e à sua natureza não paramétrica (Cheng & Wang, 2019).

### 3.3.3 Redes Neurais

Redes neuronais, também conhecidas como redes neuronais artificiais (ANN), são uma subárea de aprendizagem automática que pretende simular o funcionamento do cérebro humano, mais concretamente a comunicação dos neurónios, para conseguir relacionar e interpretar informação (IBM, 2023).

Uma rede neuronal é composta por 3 ou mais camadas de nós, sendo que uma delas é a camada de nós de entrada, outra a camada dos nós de saída e entre estas duas camadas, pelo menos mais uma camada escondida. Tal como é verificável na Figura 7, cada um dos nós da camada anterior conecta-se a todos os nós existentes na camada seguinte (Nielsen, 2021). Cada uma destas ligações tem associado um peso e cada um dos nós da camada seguinte ligado a esta conexão apresenta um limite. O valor de saída do nó, também conhecido como valor de ativação, corresponde à seguinte fórmula:

$$\sum_{j=0}^n w_j x_j > limite$$

Onde  $\sum$  representa o somatório de todos os nós de input de 0 até n,  $w_j$  representa o peso da ligação ao nó destino,  $x_j$  representa o valor de ativação do nó de origem e o limite representa o valor que deve ser atingido para se ativar o nó da camada seguinte.

Tal como podemos verificar, caso o somatório de todos os produtos do valor do nó inicial pelo peso da ligação for superior a um determinado limite calculado, então ocorre ativação deste mesmo nó. Segundo esta lógica, é então possível percecionar a ativação do nó da seguinte camada segundo o seguinte sistema (Nielsen, 2021):

$$output = \begin{cases} 0 & \text{se, } \sum_j w_j x_j \leq limite \\ 1 & \text{se, } \sum_j w_j x_j > limite \end{cases}$$

Este processo é repetido até serem atravessadas todas as camadas, culminando numa resposta identificada por um dos nós existentes na camada de saída (IBM, 2023).

### Artificial Neural Network

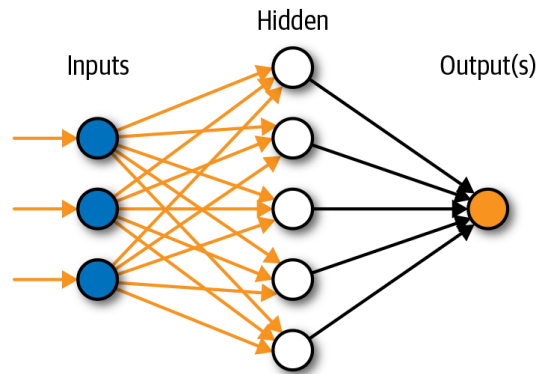


Figura 7 - Rede neuronal artificial (Castrounis, 2022)

#### 3.3.3.1 Função de Ativação

As funções de ativação são funções que durante o processo de classificação ou predição de uma informação estão responsáveis por decidir se um determinado nó deve ser ativado ou não. Para tal, são aplicadas funções matemáticas que permitem tomar uma decisão quanto à importância de um determinado nó na classificação a efetuar. A função mais simples de ativação é a função linear, podendo o valor assumido pelo nó ser qualquer valor entre menos e mais infinito. Outros exemplos de funções de ativação são (Sharma, et al., 2020):

- **Função Sigmoid:** Comprime os valores do nó entre 0 e 1 sendo bom para determinar a probabilidade de existir uma determinada classe. É bastante utilizada em problemas de classificação binários.
- **Função TanH:** Função semelhante à função sigmoide e que apresentando simetria na origem, variando os seus valores entre -1 e 1.
- **Função ReLU:** Função que permite um limite máximo de nós ativados de uma só vez. Tal implica que um nó será de desativado apenas quando o resultado da transformação linear for 0. Os seus valores variam entre 0 e mais infinito.
- **Função Leaky ReLU:** Versão melhorada da função ReLU permitindo valores compreendidos entre menos e mais infinito. Deste modo torna-se mais fácil distinguir valores que apresentem resultado negativos.
- **Função Softmax:** Combinação de múltiplas funções sigmoide, sendo ideal para problemas que apresentam múltiplas classes de classificação.

### 3.4 Aprendizagem Profunda

Aprendizagem profunda (AP) trata-se de uma subárea de investigação das redes neuronais sendo ela mesma uma rede neuronal. O que a distingue de uma rede neuronal convencional é a necessidade de grandes quantidades de informação de modo a conseguir treinar modelos que consigam realizar classificações ou previsões (IBM, 2023).

No que toca a um algoritmo de aprendizagem automática, no caso de se tratar de um modelo de classificação, de um modo geral é necessário a recolha de informação rotulada para que seja possível realizar previsões. Para além disto, deste conjunto de informação recebida é necessário extrair um conjunto de características de forma que possa esta ser utilizada para a realização das classificações. Este fenómeno de extração de características é denominado de pré-processamento, sendo necessária intervenção humana para a sua realização.

Por outro lado, a aprendizagem profunda surge como um modo de conseguir realizar pré-processamento sem a necessidade de intervenção humana e mesmo sem a necessidade de ter informação rotulada. Tal como se pode observar pela Figura 8, um algoritmo de aprendizagem profunda é capaz, à semelhança de qualquer rede neuronal, de realizar este pré-processamento e classificação de um modo automático não necessitando de qualquer intervenção humana. Tal só é possível através da capacidade das camadas escondidas da rede de perceber quais são as características mais relevantes de se avaliar, criando múltiplos níveis de abstração para a representação da informação (NetApp, 2023). Para que o sistema seja capaz de criar estes níveis de abstração é, no entanto, necessário recorrer a uma grande quantidade de dados levando a que estes sistemas exijam um longo período de treino.

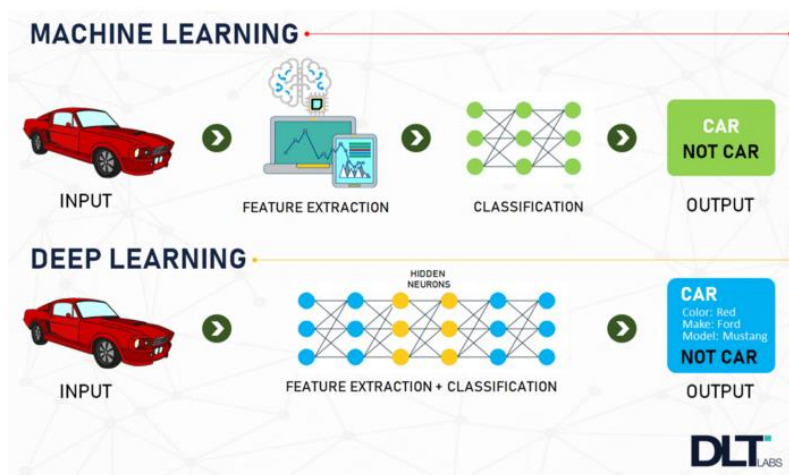


Figura 8 - Aprendizagem automática vs aprendizagem profunda (DLT Labs, 2020)

Por fim, este tipo de algoritmo recorre ao processo de *back-propagation* que permite o ajuste dos pesos das várias ligações e os limites de cada nó, de modo a obter resultados mais precisos no futuro.

Existem diversos algoritmos de aprendizagem profunda existentes sendo que cada um tem as suas aplicações e as suas vantagens e desvantagens.

### 3.4.1 Redes Neurais Convolucionais (CNN)

As redes neuronais convolucionais tipo de rede neuronal que visa a simulação da capacidade humana de processar dados visuais. Estas são capazes de aplicar filtros de modo em dados visuais de modo a manter a relação de vizinhança entre os pixels da imagem ao longo da rede (Vargas, et al., 2016).

Uma rede neuronal convolucional é composta por 3 camadas, sendo elas a camada de convolução, a camada de pooling e por fim a camada de completamente conectada. As primeiras camadas são responsáveis por conseguir identificar características simples como cores e limites enquanto camadas mais avançadas acabam por ser capaz de identificar elementos maiores e formas. Apesar de tipicamente esta ser a ordenação pela qual são criadas estas redes, estas podem apresentar mais do que uma camada de convolução ou mais do que uma camada de pooling, tal como podemos verificar pela Figura 9. Quanto maior for o número de camadas definidas na rede maior a complexidade da mesma e melhor a identificação e classificação de entidades presentes nas imagens (IBM, 2023).

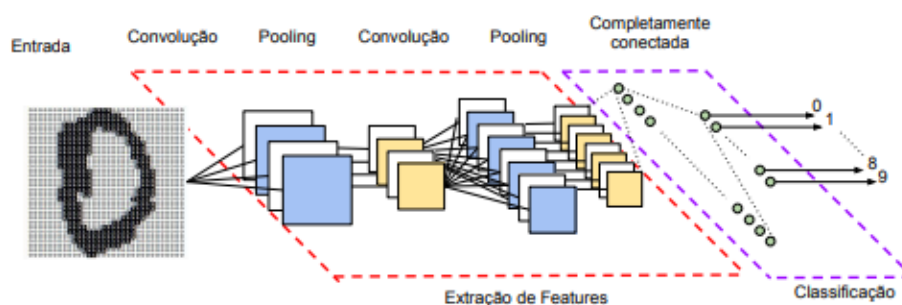


Figura 9 - Diferentes camadas de rede neuronal convolucional (Vargas, et al., 2016)

#### 3.4.1.1 Camada de Convolução

A camada de convolução é uma camada chave para o algoritmo deste tipo de redes sendo que é onde a maioria da computação ocorre. É através desta camada que se define quais são as principais características a extrair de uma imagem recebida, através da aplicação de filtros. O filtro, tal como apresentado na Figura 10, é caracterizado por uma matriz de uma dimensão menor que a imagem que está a ser recebida e que irá deslizar por toda a imagem a fim de conseguir recolher uma característica convolucional (Pekhrel, 2019).

É importante ter em atenção no número de canais existentes na imagem uma vez que o filtro a ser aplicado deverá ter o mesmo número de canais. No caso de uma imagem em escalas de cinzento, estas apenas possuem um canal sendo necessário aplicar filtros com também 1 canal. Por outro lado, uma imagem colorida, como por exemplo RGB, apresenta 1 canal por cada uma das cores presentes na imagem. Sendo assim, o filtro a ser aplicado também deverá apresentar 3 canais.

Caso seja necessário recolher mais do que uma característica de uma determinada imagem é necessário recorrer à utilização de diferentes filtros, cada um responsável por conseguir identificar a sua característica. É importante referir que para que este processo ocorra corretamente é necessário que todos os filtros envolventes apresentem as mesmas dimensões.

Por fim, é aplicada uma função de ativação sobre os filtros recolhidos de modo a conseguir produzir o resultado de saída desta camada. O produto resultante desta função de ativação deverá ser uma matriz com as mesmas dimensões que os diversos filtros aplicados e uma quantidade de canais idêntica às da imagem original.

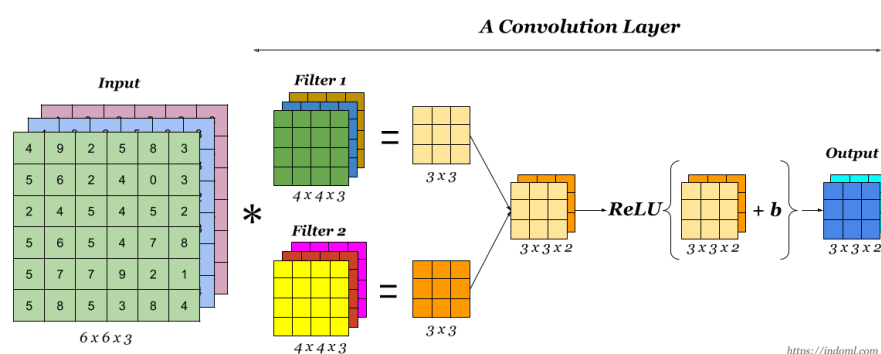


Figura 10 - Camada de convolução (Pohrel, 2019)

### 3.4.1.2 Camada de pooling

A camada de *pooling* é responsável por reduzir o tamanho da imagem que é recebida sendo geralmente precedida por uma camada de convolução. À semelhança do que ocorre na camada de convolução, a camada de *pooling* também necessita de filtros que auxiliam na redução do tamanho da imagem com o intuito de melhorar o desempenho do algoritmo. Para que esta redução aconteça, são aplicados métodos de pooling sendo os mais comuns o pooling máximo e o pooling da média, presentes na Figura 11, sendo o primeiro responsável por retornar o maior valor detetado pelo filtro e o segundo a média dos valores englobados pelo filtro (IBM, 2023).

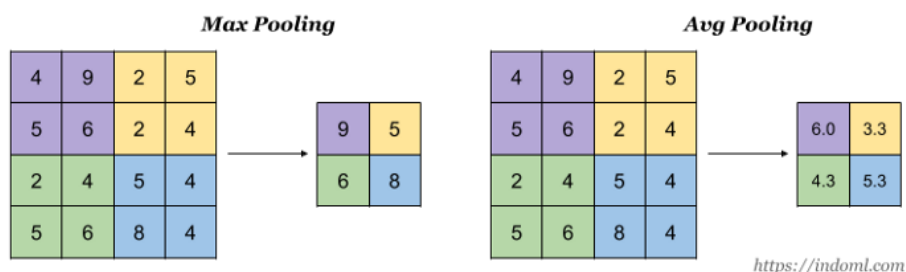


Figura 11 - Métodos de pooling mais comuns (Pohrel, 2019)

### 3.4.1.3 Camada Completamente Conectada

Por fim, a camada completamente conectada agrega o mapa de características que foram recolhidas durante as camadas anteriores de modo a conseguir capturar relações complexas entre características de alto nível, tal como apresentado na Figura 12. É através dela que são efetuadas classificações de elementos detetados na imagem recebida através da utilização de uma função de ativação (como por exemplo) softmax atribuindo valores probabilidade entre 0 e 1 (IBM, 2023).

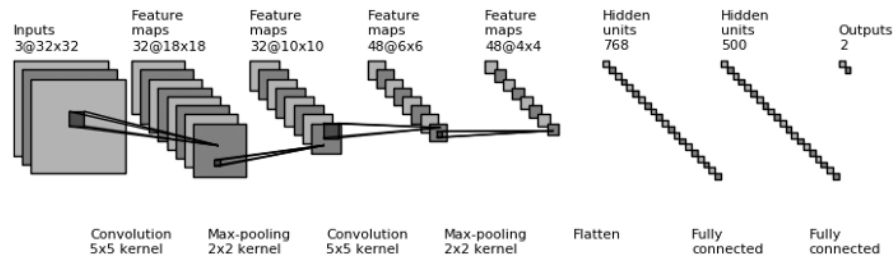


Figura 12 - Fase de classificação de uma rede convolucional (Pohrel, 2019)

### 3.4.2 Rede Neuronal Recorrente (RNN)

Redes neuronais recorrentes são um tipo de redes neuronais que usa como valores de entrada informação sequencial ou séries temporais de informação. Geralmente este tipo de redes é utilizado para a resolução de problemas relacionados com a tradução linguística, processamento de linguagem natural, deteção de discurso e mesmo reconhecimento de imagens (IBM, 2023).

No que toca a algoritmos de redes neuronais de aprendizagem profunda, tipicamente estes assumem que os valores de entrada são independentes dos valores de saída. No caso das RNN os valores de saída estão dependentemente de valores de saída de entradas que surgem previamente na sequência de informação recebida (IBM, 2023).

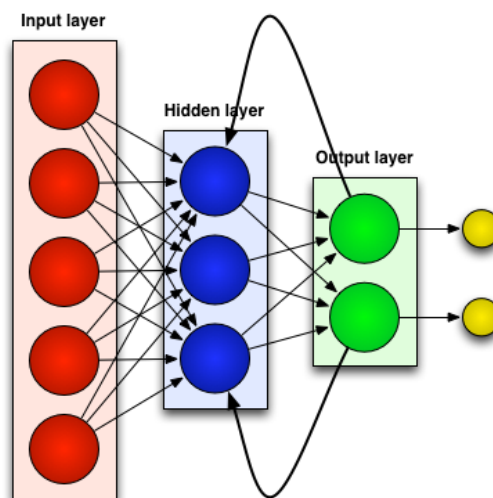


Figura 13 - Exemplo de rede neuronal recorrente (Roell, 2017)

Durante o processamento de informação, ao dar como exemplo a tentativa de previsão de uma determinada palavra, a rede recorrente recebe como entrada a sequência de letras sobre a qual se pretende realizar uma previsão. De modo a conseguir prever qual a palavra que se pretende prever e, tal como é possível verificar pela Figura 13, os resultados recebidos na camada de saída são utilizados pela camada escondida de modo a conseguir em conjunto com a nova informação descobrir qual a palavra que se pretende obter.

### 3.5 Extração de características de Áudio

Quando comparado com o processamento de imagens, o processamento de ficheiros áudio é bastante mais complexo de se analisar uma vez que é difícil realizar a sua representação em forma de matriz. No caso de imagens, podem ser utilizadas matrizes em que cada um dos seus valores representa um pixel da imagem que se pretende analisar. Por outro lado, sons são uma mistura de frequências sonoras de diferentes intensidades, havendo a necessidade de realizar a conversão da informação em um formato mais simples de modo a conseguir tirar conclusões sobre a mesma. Para além disto, existe ainda uma componente temporal associado a um determinado som, sendo a sua interpretação comparável à de um vídeo do que necessariamente uma simples imagem (Korstanje, 2021).

#### 3.5.1 Fast Fourier Transform (FFT)

Um sinal auditivo é composto por diversas ondas de apenas uma frequência. A transformação de Fourier é uma fórmula matemática que permite decompor um sinal auditivo presente no domínio do tempo em as várias frequências e amplitudes sendo o seu resultado um espectro que representa o sinal (Roberts, 2020).

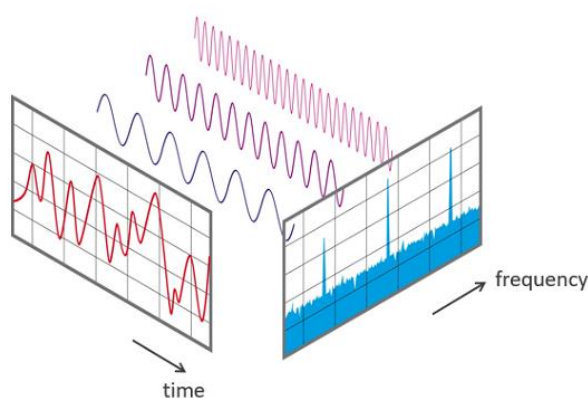


Figura 14 - Aplicação da transformação de Fourier (NTI Audio, 2023)

A transformação rápida de Fourier, apresentada na Figura 14, é um algoritmo que permite de forma eficiente aplicar e calcular a aplicação da fórmula matemática da transformação de Fourier (Roberts, 2020).

### 3.5.2 Espectrograma de Mel

O espectrograma de mel, presente na Figura 15, é um tipo de espectrograma baseado na escala de mel que pretende descrever o modo com o ser humano é capaz de perceber as características sonoras. Este espectrograma permite visualizar sons de baixas frequências, de acordo com que o ser humano é capaz de perceber melhor (Mahanta & Arvin, 2021).

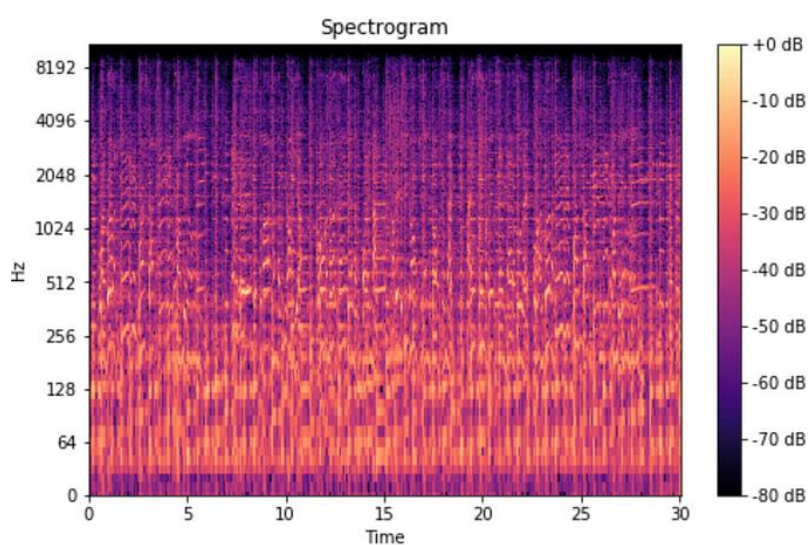


Figura 15 – Exemplo de espectrograma de mel (Roberts, 2020)

A escala de Mel utilizada facilita a percepção dois sons que apresentem a mesma diferença de frequências, quer estes sejam agudos e apresentem frequências elevadas, quer sejam graves e apresentem frequências baixas (Roberts, 2020). Através da frequência de um determinado som, é então possível realizar a conversão para esta escala recorrendo à seguinte fórmula:

$$m = 2595 * \log \left( 1 + \frac{f}{700} \right)$$

Onde m representa o valor na escala de mel e f é a frequência do som a converter (Mahanta & Arvin, 2021).

### 3.5.3 Mel Frequency Cepstral Coefficients (MFCC)

Os MFCCs, à semelhança do espectrogramas de mel são um tipo de representação sonora bastante utilizados para a extração de atributos para problemas relacionados com processamento de áudio, sendo bastante uteis na resolução de problemas de reconhecimento de voz (Muda, et al., 2010) e classificação de sons (Vimal, et al., 2021). Este tipo de representação é geralmente representado pela quantidade de coeficientes utilizados para a

caracterização do som a analisar durante um determinado intervalo de tempo, tal como pode ser observado a partir da Figura 16.

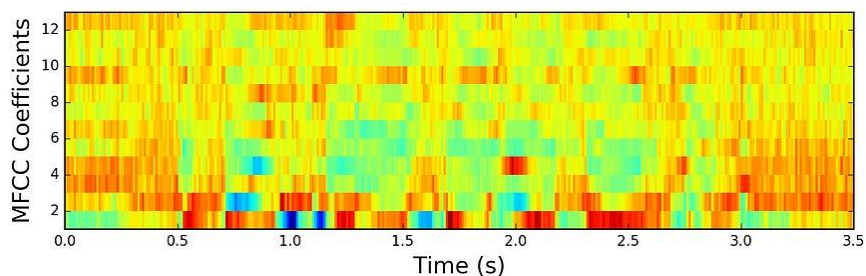


Figura 16 - Exemplo de MFCC (Nair, 2018)

A computação destes coeficientes, tal como apresentado na Figura 17, consiste num conjunto de 7 passos, começando pelo aumento da energia do sinal através da aplicação de um filtro e terminando com a obtenção dos espectro com os diferentes coeficientes (Bezoui, et al., 2017).

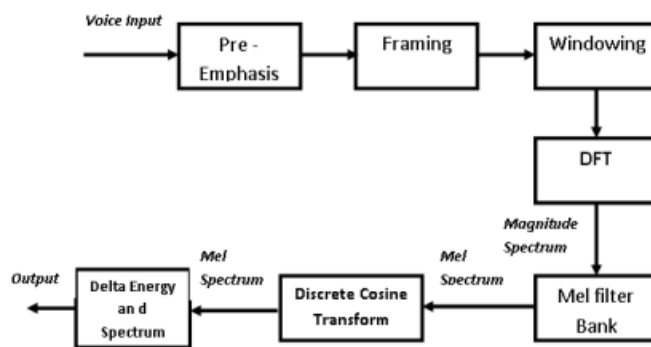


Figura 17 – Diagrama de passos de computação de MFCC (Bezoui, et al., 2017)

### 3.6 Métodos de Avaliação

A forma como o conjunto de dados é fornecido ao modelo gerado pode influenciar o desempenho do modelo. É importante ponderar o processo que será utilizado na avaliação deste modelo de modo a conseguir ter uma visão real do seu desempenho na classificação dos dados recebidos. Para além disto, este tem um papel importante na comparação com diferentes trabalhos sendo o ideal a utilização do mesmo método.

### 3.6.1 Holdout

O método de avaliação holdout, tal como apresentado na Figura 18, consiste em dividir o conjunto de dados em uma proporção  $p$  para treino e  $(1-p)$  para realização de testes (Devi, 2021).

Apesar desta divisão ser a abordagem mais comum, também é possível adicionar uma terceira divisão dos dados, sendo está o conjunto de validação. Este último conjunto poderá ser útil em casos que se pretenda encontrar dentro de um conjunto de modelos, qual o mais indicado para responder à hipótese de estudo.

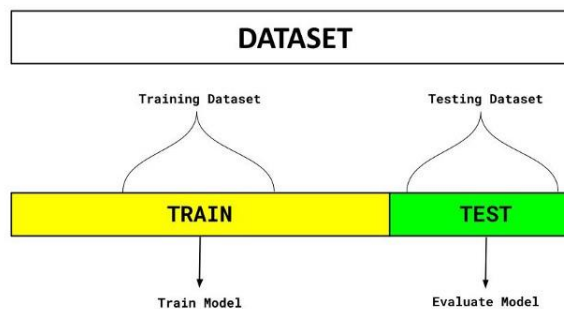


Figura 18 - Método de avaliação holdout (Devi, 2021)

### 3.6.2 K-Fold Cross Validation

A amostragem K-Fold Cross Validation, consiste numa fase inicial em segmentar o conjunto de dados da avaliação aleatoriamente em  $K$  parte iguais. De seguida, para cada segmentação, uma parte da informação deverá ser utilizada para treinar o modelo e outra parte da informação deve ser preservada para a realização de testes sobre o modelo treinado, tal como apresentado na Figura 19. Terminadas as várias avaliações, devem ser analisados os dados obtidos em cada um dos  $k$  folds e realizada a sua média, sendo posteriormente retiradas conclusões sobre a eficácia do modelo (Brownlee, 2018).

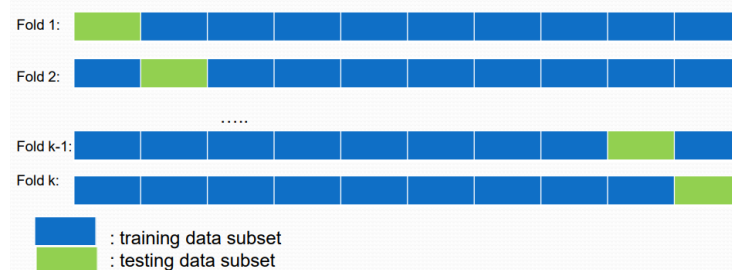


Figura 19 - K-Fold Cross Validation (Japkowicz, 2023)

## 3.7 Tecnologias

### 3.7.1 Python

Python é uma linguagem de programação de alto nível que é interpretada, orientada a objetos e com uma semântica dinâmica. Como tal, esta torna-se uma das linguagens mais utilizadas para desenvolvimento rápido de aplicações, bem como construção de scripts permitindo conseguindo ainda garantir a alta manutenibilidade do código produzido na mesma devido à simplicidade de leitura da sintaxe (Python Software Foundation, 2023).

Apesar da sua simplicidade, o Python não deixa de ser uma das linguagens mais influentes do mercado atual, sendo a linguagem mais popular da atualidade e tendo contado 5 anos em que recebeu o prémio de linguagem de programação do ano (Tiobe - The Software Quality Company, 2023).

Uma das suas grandes vantagens face às demais linguagens é a quantidade de bibliotecas e frameworks existentes que permitem auxiliar o seu utilizador no desenvolvimento dos seus projetos. Estas bibliotecas incluem bibliotecas de aprendizagem automática como por exemplo PyTorch, TensorFlow e Caffe ou mesmo outras que permitam desenvolvimento de projetos web como por exemplo Django, CherryPy e Pyramid.

Por fim, é ainda importante salientar que o Python se posiciona no mercado como sendo a linguagem de eleição para projetos relacionado com *aprendizagem automática*. Tal como é possível verificar pela Figura 20, o Python nos últimos anos tornou-se progressivamente a linguagem mais popular desde 2019, sendo a linguagem com mais perguntas efetuadas na plataforma StackOverflow mensalmente.

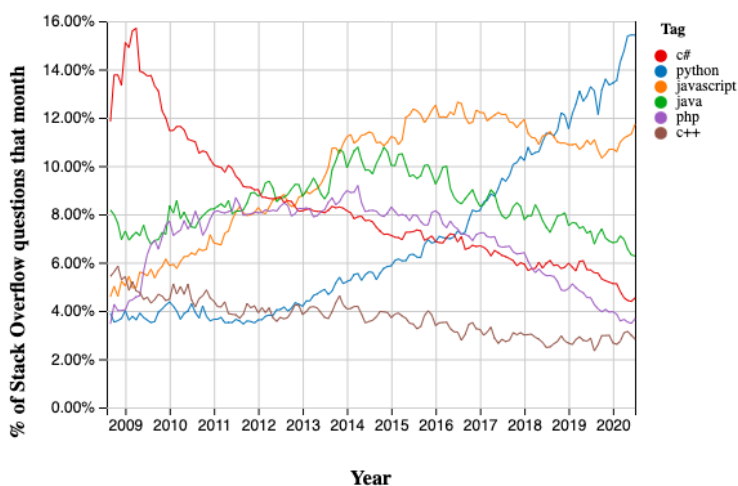


Figura 20 - Quantidade - Linguagens mais populares na plataforma StackOverflow (Gupta, 2021)

### 3.7.2 TensorFlow/Keras

O TensorFlow é uma plataforma aberta ponta a ponta utilizada para o desenvolvimento de projetos da área de aprendizagem automática e que providencia soluções para acelerar tarefas em qualquer etapa do processo de aprendizagem. Como tal, o TensorFlow ainda oferece um conjunto de ferramentas que auxiliam a consolidação do pré-processamento de dados como *datasets* padrão, pipelines de informação, processamento de diferentes camadas e ferramentas de validação e transformação (TensorFlow, 2023).

Uma mais-valia existente nesta tecnologia é a sua compatibilidade com a API de aprendizagem profunda “Keras” que é executado sobre a estrutura de *aprendizagem automática* já estabelecida pelo TensorFlow. Esta API tem como principais focos a simplicidade de modo a permitir reduzir o custo de compreensão da ferramenta, a sua flexibilidade para trabalhar com fluxos simples ou complexos e por fim a sua grande capacidade de operar a nível organizacional de um modo fiável (Keras, 2023). Na Tabela 1 é possível verificar o levantamento das vantagens e desvantagem da utilização da ferramenta TensorFlow.

Tabela 1 - Vantagens e desvantagens da ferramenta TensorFlow (Geeks for Geeks, 2023)

Vantagens	Desvantagens
Compatibilidade com a API Keras	Falta de suporte para windows
Visualização da informação (suporte gráfico)	Suporte de GPU (Graphics Processing Unit)
Grande comunidade	Execução lenta
Escalabilidade	Atualizações frequentes
Ferramenta open-source	

### 3.7.3 PyTorch

O PyTorch é uma framework open-source de aprendizagem automática baseada na linguagem Python e na biblioteca Torch que permite acelerar o processo de desenvolvimento de redes neurais profundas conseguindo agilizar o processo de construção de protótipos e de implantação dos mesmos. Esta ferramenta é principalmente utilizada por cientistas de dados na realização de pesquisas no âmbito da área de inteligência artificial, tendo sido alvo de um grande aumento de popularidade que continua a subir até à atualidade. No que toca aos seus casos práticos, esta framework é bastante utilizada no processamento de linguagem natural, aprendizagem reforça e classificação de imagens (Yasar & Lewis, 2023). O levantamento do prós e contras desta ferramenta está presente na Tabela 2.

Tabela 2 - Vantagens e desvantagens da ferramenta PyTorch (Educba, 2023)

Vantagens	Desvantagens
Facilidade de aprendizagem	Framework recente havendo pouco suporte da comunidade
Facilidade de realizar debug	Falta de ferramentas de monitorização e visualização
Bibliotecas úteis	
Flexível e rápido	
Suporte para GPU e CPU	

### 3.7.4 Caffe

Caffe é uma framework de aprendizagem profunda que suporta uma variedade de arquiteturas de linguagem profunda como por exemplo CNN e RCNN (Isbilen, 2022). Com ela é possível definir modelos, solucionadores e otimizadores através de em ficheiros de configuração devido à expressividade da sua arquitetura. Para além disto, permitem trocar entre a utilização de GPU e CPU culminando numa ferramenta que permite eliminar problemas relacionados com código estático. Esta ferramenta é ainda uma das mais rápidas no que toca a implementações CNN sendo este um dos seus principais pilares (Caffe, 2023). Algumas das vantagens e desvantagens da ferramenta Caffe estão presentes na Tabela 3.

Tabela 3 - Vantagens e desvantagens da ferramenta Caffe (Isbilen, 2022)

Vantagens	Desvantagens
Facilidade de aprendizagem	Escassez de documentação
Open-source	Dificuldade de modelação para alguns tipos de redes
Rápido	Treino Multi-GPU é parcialmente suportado
Suporte para GPU e CPU	

## 3.8 Trabalhos Relacionados

De modo a compreender as abordagens mais utilizadas para a resolução do problema a resolver, foi realizado um estudo das diversas abordagens utilizadas pelos vários participantes do concurso George B. Moody PhysioNet Challenge 2022.

Na Tabela 4 apresentam-se algumas das soluções que foram submetidas e posteriormente avaliadas pelos próprios autores. As métricas analisadas são métricas relacionadas com as medidas de avaliação definidas no concurso, não descorando, no entanto, métricas derivadas da análise de uma matriz de confusão.

Tabela 4 – Submissões de concurso George B. Moody PhysioNet Challenge 2022

<b>Autor</b>	<b>Método</b>	<b>Extração de features</b>	<b>Métricas Analisadas</b>
(Lu, et al., 2022)	Lightweight CNN Random Forest	Mel-spectrogram, Socio-demographic data and statistic	Murmur Score: 0.780 Clinical Score: 12110 Recall: 66% Especificidade: 30.6% F1-Score: 48.3% Overall Accuracy: 55.45%
(Mcdonald, et al., 2022)	Hidden Semi Markov Model (HSMM)	Phonocardiogram segmentation, Crop Hz range (0-800Hz)	Murmur Score: 0.776 Clinical Score: 11144 Recall: 84% Especificidade: 31% Overall Accuracy: 57.5%
(Xu, et al., 2022)	Hierarchical Multi- scale Convolutional Neural Network (HMS-Net)	Downsample frequency, Frequency Density distribution, Phonocardiogram	Murmur Score: 0.776 Clinical Score: 12069 Recall: 90.56 Overall Accuracy: 62.96%
(Walker, et al., 2022)	Dual Bayesian ResNet (DBRes), XGBoost	Log Mel Spectrogram	Murmur Score: 0.771 Clinical Score: 12637 Exatidão: 76.2%
(Lee, et al., 2022)	CNN, Light CNN	Phonocardiogram, Log mel-spectrograma, Short-time Fourier Transformation, Constant Q Transformation	Murmur Score: 0.767 Clinical Score: 11933 Exatidão: 76.7%
(Alkhodari, et al., 2022)	Transformer-based neural network	Wavelet features transformation	Murmur Score: 0.757 Clinical Score: 14410 Exatidão: 76.1

Para além dos trabalhos relacionados com o concurso, também foram recolhidos alguns trabalhos relacionados com o problema em questão, mas que não se encontram diretamente relacionados com o concurso. Como tal, os resultados destes trabalhos não se focam em tentar atingir as medidas definidas na prova, mas sim obter as medidas de desempenho que se conseguem obter através de uma matriz de confusão, tal como se pode observar na Tabela 5.

Tabela 5 - Submissões de soluções que utilizam o *dataset* do concurso George B. Moody  
PhysioNet Challenge 2022

<b>Autor</b>	<b>Método</b>	<b>Extração de features</b>	<b>Métricas Analisadas</b>
(Chen, et al., 2022)	CNN BiLSTM PCG-A Resnet	Phonocardiogram Segmentation Band-pass Butterworth 25-250Hz	Exatidão: 94.58% Recall: 76.68% F1-Score: 92.2%
(Patwa, et al., 2023)	1D-CNN LSTM-RNN CRNN	Time-Frequency Wavelet scattering transform	Exatidão: 87.05% Recall: 88.02% F1-score: 87.72%
(Vimalajeewa, et al., 2023)	Logistic Regression Support Vector Machine Neural Network	Self-similarity e wavelet entropy	Exatidão: 76.61% Recall: 54.03 Overall Accuracy: 68.07%
(Nogueira, et al., 2019)	SVM Random Forest CNN	Time-domain features MFCC Motis	Recall: 87.37% Especificidade: 79.07% Overall Accuracy: 83.22%

## 4 Análise de Valor

Neste capítulo é apresentada a análise de valor efetua da solução sendo abordados o estado atual da área das doenças cardiovasculares para identificação e análise de oportunidades. É também nesta secção são realizadas a análise e abordagem do problema com recurso à técnica de análise funcional FAST e são comparadas as frameworks TensorFlow, PyTorch e Caffe para recorrendo ao método de análise hierárquica.

### 4.1 Processo de Inovação

Quando se fala em inovação esta é vista como um meio de conseguir gerar uma ideia, sendo geralmente uma vista de alto nível de como resolver um problema. No que toca ao processo de realização de inovação, este pode ser subdividido em 3 áreas sendo elas o *Fuzzy Front End* (FFE), o desenvolvimento do novo produto (NPD) e por fim a comercialização do mesmo (Koen, et al., 2002).

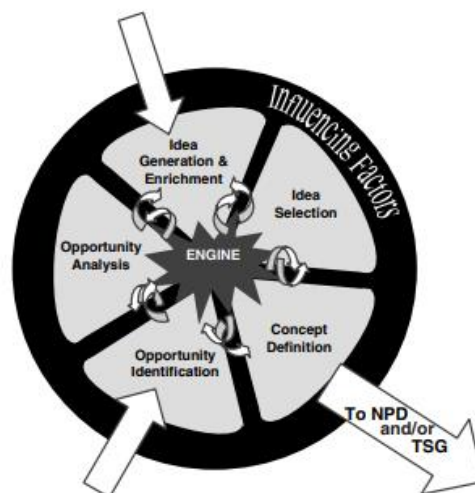


Figura 21 - Modelo de desenvolvimento de novo conceito (Koen, et al., 2002)

Este modelo de desenvolvimento do novo conceito que, tal como é possível observar pela Figura 21, aborda 3 conceitos chave importantes no desenvolvimento de um novo conceito. Em primeiro lugar o motor que é caracterizada pelo modo de operar da organização ou empresa que vai desenvolver o produto. De seguida, temos as 5 atividades chaves que controlam a atividade dos vários elementos do FFE. Por fim, existem os fatores de influência que podem ser tanto internos (capacidade organizacional) ou externos (canais de distribuição, leis, competição) e que afetam o processo de inovação durante todo o percurso (Koen, et al., 2002).

### 4.1.1 Identificação de Oportunidade

Tal como já referido anteriormente, atualmente as doenças cardiovasculares são a principal responsável pela maioria das mortes que acontece a nível mundial. Na Figura 22 é possível verificar que em 2019 cerca de 55.4 milhões de pessoas morrem, sendo que destes cerca de 18.56 milhões aconteceram devido a doenças cardiovasculares (Ritchie & Roser, 2019). Isto constitui cerca de 33% de todas as mortes que ocorreram durante o ano, levantando à preocupação recorrente de tentar resolver o problema.

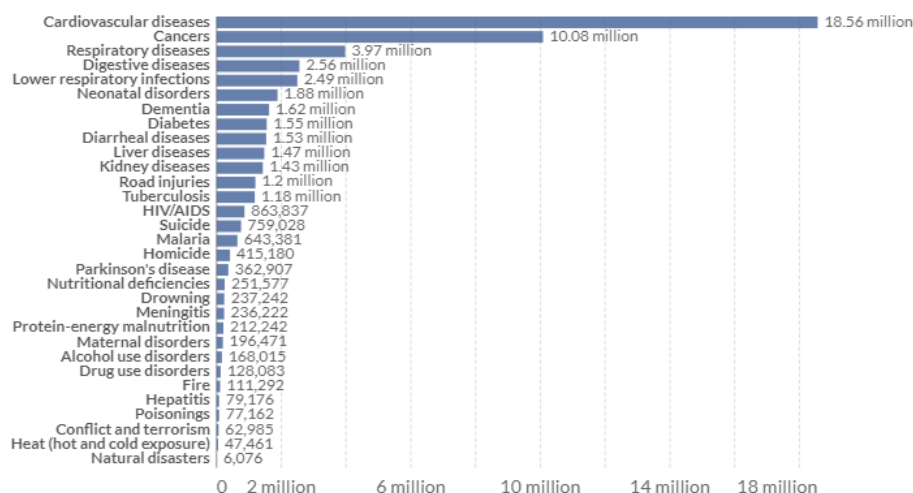


Figura 22 – Causas de morte no ano de 2019 (Ritchie & Roser, 2019)

Com o avançar da tecnologia e através do aumento da carga de trabalho no dia a dia, grande parte das pessoas acaba por adotar um estilo de vida mais sedentário ou mesmo praticar hábitos de alívio de stress que constituem um grave perigo para a própria saúde (Dowd, 2020). Como tal, a melhor forma de conseguir prevenir doenças cardiovasculares é através da prática recorrente de um estilo de vida saudável. No entanto, nem sempre é possível prevenir estes tipos de problemas sendo na mesma necessário conseguir compreender quando estes se tornam perigosos para a vida de cada indivíduo.

Atualmente, existem diversas abordagens para conseguir detetar problemas cardiovasculares, no entanto, estes não só se apresentam como sendo bastante dispendiosos a nível monetário como também necessitam de um período demasiado extenso para poder ser alargado a toda

a população. É neste momento que entra a utilização do estetoscópio que para além de ser fácil de utilizar, consegue realizar um despiste inicial deste tipo de doenças, tal como já foi referido anteriormente (Samuel, 2022).

Apesar desta ser uma boa abordagem para o despiste inicial de uma anomalia no coração, este não deixa de ter alguns inconvenientes como a dependência da experiência da pessoa que o está a utilizar, levando a potenciais erros humanos, como também a presença de sons externos pode influenciar o correto funcionamento do instrumento. Como tal, torna-se imprescindível a procura de soluções que permitam de forma eficaz detetar este tipo de doenças atempadamente.

#### **4.1.2 Análise da Oportunidade**

De acordo com o estado da arte atual, já possuímos conhecimento de que apesar do ser humano possuir a capacidade de efetuar corretamente um elevado número de tarefas, qualquer máquina desenhada para realizar uma determinada tarefa a conseguirá fazer com bastantes melhores resultado. Sendo assim, o mercado da área de informática e consecutivamente de inteligência artificial tem vindo a crescer exponencialmente de modo a conseguir tornar a vida das pessoas mais facilitada através da construção de máquinas autónomas (Ramachandran, et al., 2022).

No caso das doenças cardiovascular, e tal como referido na secção 4.1.1, quanto mais rápido for detetada uma anomalia a nível do coração, mais rápido se podem tomar medidas que ajudem a mitigar ou resolver o problema antes que este apresente repercussões irreversíveis. Sendo assim, com o auxílio de tecnologias de inteligência artificial torna-se possível o desenvolvimento de um algoritmo de inteligência artificial que consiga autonomamente receber um segmento de som de batimentos cardíacos de um indivíduo e finalmente aferir um resultado sobre o estado do coração. Deste modo, não só seria possível garantir uma maior precisão na deteção de doenças cardiovasculares, como também se tornaria uma ferramenta de auxílio à tomada de decisão do próprio profissional de saúde.

## **4.2 Valor da solução**

Apesar de atualmente se tentar resolver o problema de deteção de doenças cardiovasculares com um grande grau de confiança, a solução a desenvolver surge da existência de um desafio que promove a elaboração de soluções para deteção de anomalias no batimento cardíaco do coração (Reyna, et al., 2022). Apesar do valor atingido com a solução não ser um valor monetário, este consegue obter-se através do desenvolvimento de uma solução que consiga trazer benefícios não só para a comunidade científica ao criar caminho para a elaboração de outras soluções como para os pacientes com doenças cardíacas que conseguem despistar mais cedo o seu problema e, por conseguinte, arranjar uma solução.

### 4.2.1 Proposta de valor

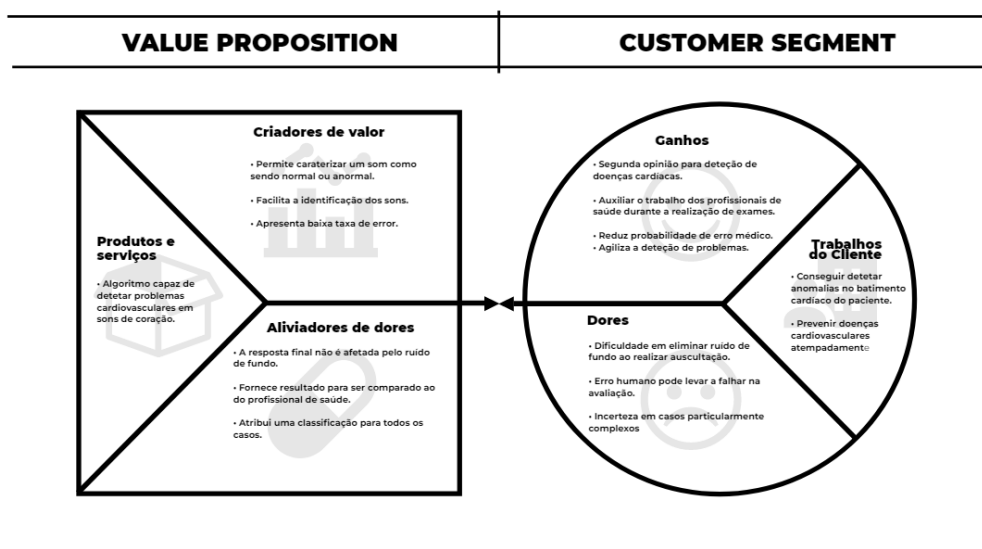


Figura 23 – Proposição de valor

Tal como podemos verificar pela análise da Figura 23, a proposta de valor pensada tem como objetivo conseguir desenvolver um sistema capaz de receber vários sons provenientes dos diversos pontos de auscultação dos pacientes e conseguir deduzir se estes produzem sons normais ou anormais. Este sistema tem como mercado alvo os profissionais de saúde que realizam rastreios na população de modo a conseguir detetar atempadamente quaisquer anomalias que devem ser posteriormente estudadas de um modo mais prudente e especializado.

Este sistema trará como vantagem principal o facto de conseguir detetar com alta eficácia e precisão a presença de uma anomalia à qual se deverá tomar medidas. Deverá também ser capaz de conseguir eliminar os ruídos de fundo provenientes dos pontos de auscultação, permitindo assim garantir que o resultado não será afetado por elementos indesejados.

Estas vantagens aliadas à opinião do profissional de saúde permitem então o despiste inicial de algum problema que deva ser estudado com um maior pormenor. Esta solução deverá então poder ser algo passível de ser incorporado num auscultador médico, permitindo que seja utilizado tanto como ferramenta manual (utilização da audição humana para avaliação) como de uma deteção automática, efetuada pelo modelo a desenvolver.

### 4.3 Técnica Sistemática de Análise Funcional

A técnica sistemática de análise funcional (FAST) é uma técnica de análise funcional que tem como propósito desenvolver uma representação gráfica que demonstre a relação lógica entre as funções que englobam o projeto que se pretende elaborar, o produto ou serviço e o processo com base em questões de “como” e “porquê” (Nicola, 2023).

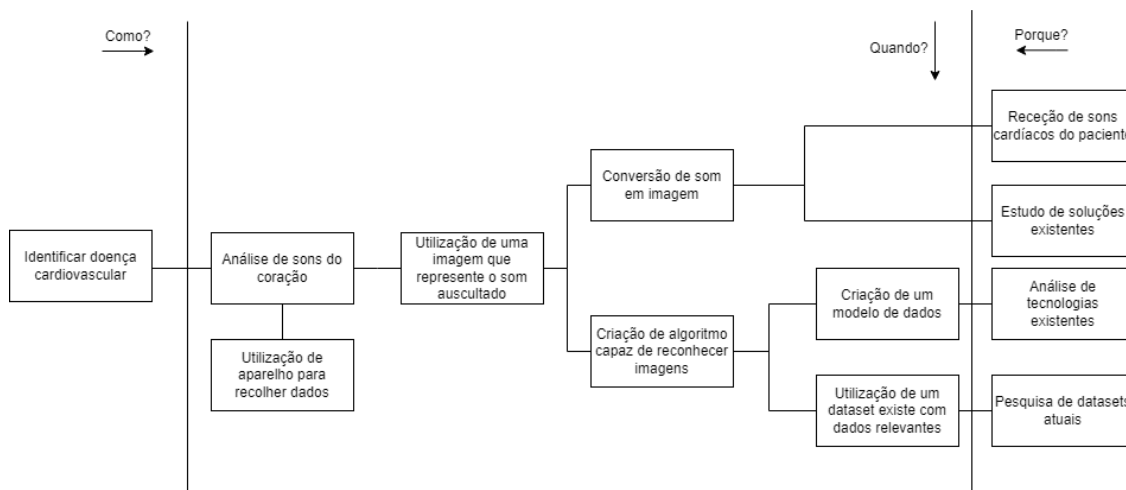


Figura 24 - Diagrama FAST

Através da elaboração deste diagrama, tal como apresentado na Figura 24, torna-se mais fácil compreender o propósito da aplicação uma vez que se compreende não só as funções que compõem a solução a desenvolver, bem como a relação lógica entre elas. Com a utilização desta abordagem é possível ainda fomentar a criatividade dos envolvidos, identificar funções em falta, melhorar a comunicação e acima de tudo partilhar o conhecimento do projeto com todos os envolvidos.

No caso do presente projeto, torna-se fácil de compreender que o objetivo final deste documento é a concretização de uma aplicação que seja capaz de identificar a presença de doenças cardíacas. Para tal, são necessários como base todos os objetivos enumerados na coluna da extremidade direita, sendo estes a receção dos sons que pretendemos classificar, o estudo de como se irá realizar conversão de sons em imagens, a análise de tecnologias existentes para que se consiga desenvolver um modelo fiável e a pesquisa de *datasets* com dados relevantes para que seja possível treinar os modelos a desenvolver.

#### 4.4 Método de Análise Hierárquico

O método de análise hierárquico (AHP) trata-se de um método utilizado para a tomada de decisões multicritério discretos sendo possível a utilização tanto de critério qualitativos como de critérios quantitativos. A ideia do método tem como base dividir o problema de decisão em vários níveis hierárquicos, agilizando assim a sua compreensão e consequentemente avaliação (Saaty, 1990).

Em primeiro lugar é necessário definir a pergunta à qual queremos responder com a utilização deste método. Com base nela, deverá ser então repartido o problema num conjunto de critérios que têm influência na tomada de decisão. Para além disso, deverão também ser estabelecidos os vários subcritérios associados a cada um dos critérios, sendo estes o resultado do método utilizado (Saaty, 1990). Na Figura 25 está representada a árvore

hierárquica que foi pensada para o problema em questão. Este trata-se de decidir qual a framework mais indicada para o desenvolvimento do algoritmo proposto no presente documento. Para tal, foram especificados 4 critérios, sendo estes:

- **Facilidade de utilização:** Facilidade de aprendizagem na utilização da framework e o quão simples é criar modelos a partir da mesma.
- **Comunidade e suporte:** A popularidade das frameworks nos últimos anos, bem como a quantidade de comunidades online e recursos de aprendizagem disponíveis.
- **Desempenho:** A capacidade de obter resultados a partir dos algoritmos desenvolvidos na menor janela temporal possível.
- **Dimensão dos dados:** A capacidade de receber grandes quantidades de dados provenientes de *datasets* existentes para o seu processamento.

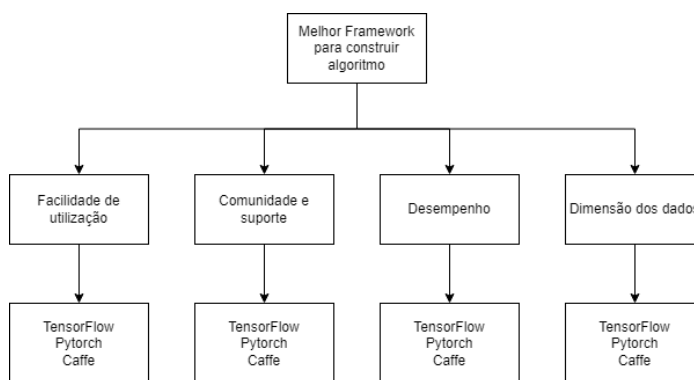


Figura 25 - Árvore hierárquica de decisão da melhor framework para construção do algoritmo

Após estabelecidos os critérios é então necessário a construção de uma matriz capaz de discriminar as diferentes prioridades de cada um dos critérios estabelecidos face aos demais. Para tal, foi utilizada uma escala de importância com valores de 1 a 9, onde o 1 significa que os critérios a comparar apresentar um nível de prioridade idêntico e 9 significa importância absoluta sobre o critério a comparar (anexos, Figura 56).

Tabela 6 - Comparação dos diferentes critérios

	<b>Facilidade de Utilização</b>	<b>Comunidade e Suporte</b>	<b>Desempenho</b>	<b>Dimensão dos dados</b>
<b>Facilidade de Utilização</b>	1	3	1/4	1/6
<b>Comunidade e Suporte</b>	1/3	1	1/5	1/4
<b>Desempenho</b>	4	5	1	1/2
<b>Dimensão dos dados</b>	6	4	2	1
<b>Somatório</b>	34/3	13	69/20	23/12

Tendo os critérios estabelecidos estabelecido, foi então construída a Tabela 6 onde foram estabelecidas as várias prioridades dos critérios definidos para a resposta que se pretende obter. Tendo todas as prioridades definidas foi então possível a construção de uma tabela que normaliza os vários valores definidos nas tabelas ao dividir cada um dos valores pelo somatório de todos os valores da coluna em que este se insere (Saaty, 1990), tal como observável na Tabela 7. Por fim, através da realização da média das várias células correspondentes a cada linha, é possível obter a prioridade relativa de cada um dos critérios. Como tal, podemos concluir que para os critérios enumerados, o critério com uma maior prioridade é a dimensão dos dados que a framework suporta e o menor é a comunidade e suporte existentes para a mesma.

Tabela 7 - Normalização dos valores e respetivas prioridades relativas

	<b>Facilidade de Utilização</b>	<b>Comunidade e Suporte</b>	<b>Desempenho</b>	<b>Dimensão dos dados</b>	<b>Prioridade Relativa</b>
<b>Facilidade de Utilização</b>	3/34	3/13	5/69	2/23	<b>0.1196</b>
<b>Comunidade e Suporte</b>	1/34	1/13	4/69	3/23	<b>0.0737</b>
<b>Desempenho</b>	6/17	5/13	20/69	6/23	<b>0.3221</b>
<b>Dimensão dos dados</b>	9/17	4/13	40/69	12/23	<b>0.4846</b>

Após obtidas as prioridades relativas é necessário compreender a fiabilidade dos vários valores obtidos. Para tal, deve ser calculado a razão de consistência (RC) de modo a mensurar o quanto os julgamentos são consistentes face a grandes amostras de juízos aleatórios. A fim de que os dados obtidos sejam confiáveis é necessário que este índice seja inferior a 10%, o que significa que os valores definidos na Tabela 6 estão muito perto de serem aleatórios. (Saaty, 1990). Após realizados os cálculos, foi obtido um RC de 0.097 ou 9.7% o que significa que os valores dados aos critérios são consistentes e fiáveis.

De seguida, é necessário a realizar a comparação das várias alternativas definidas segundo os vários critérios estabelecidos inicialmente. As seguintes tabelas indicam então as diversas comparações que foram necessárias de se realizar.

Tabela 8 - Propriedades relativas de facilidade de utilização

<b>Facilidade de Utilização</b>	<b>PyTorch</b>	<b>TensorFlow</b>	<b>Caffe</b>	<b>Prioridade Relativa</b>
<b>PyTorch</b>	1/7	1/5	1/13	<b>0.1399</b>
<b>TensorFlow</b>	3/7	3/5	9/13	<b>0.5736</b>
<b>Caffe</b>	3/7	1/5	3/13	<b>0.2864</b>

Tabela 9 - Propriedades relativas de comunidade e suporte

<b>Comunidade e Suporte</b>	<b>PyTorch</b>	<b>TensorFlow</b>	<b>Caffe</b>	<b>Prioridade Relativa</b>
<b>PyTorch</b>	3/13	5/23	1/3	<b>0.2604</b>
<b>TensorFlow</b>	9/13	15/23	5/9	<b>0.6333</b>
<b>Caffe</b>	1/13	3/23	1/9	<b>0.1062</b>

Tabela 10 - Propriedades relativas de desempenho

<b>Desempenho</b>	<b>PyTorch</b>	<b>TensorFlow</b>	<b>Caffe</b>	<b>Prioridade Relativa</b>
<b>PyTorch</b>	6/11	4/7	1/2	<b>0.5390</b>
<b>TensorFlow</b>	3/11	2/7	2/6	<b>0.2973</b>
<b>Caffe</b>	2/11	1/7	1/6	<b>0.1638</b>

Tabela 11 - Propriedades relativas de dimensão de *datasets*

<b>Dimensão de <i>datasets</i></b>	<b>PyTorch</b>	<b>TensorFlow</b>	<b>Caffe</b>	<b>Prioridade Relativa</b>
<b>PyTorch</b>	3/7	3/7	3/7	<b>0.4286</b>
<b>TensorFlow</b>	3/7	3/7	3/7	<b>0.4286</b>
<b>Caffe</b>	1/7	1/7	1/7	<b>0.1429</b>

Terminados os cálculos das várias prioridades relativas referentes a cada um dos critérios e de cada uma das alternativas face aos diferentes critérios explorados. Como tal, na Figura 26, conseguimos visualizar cada uma das prioridades relativas referentes a cada critério.

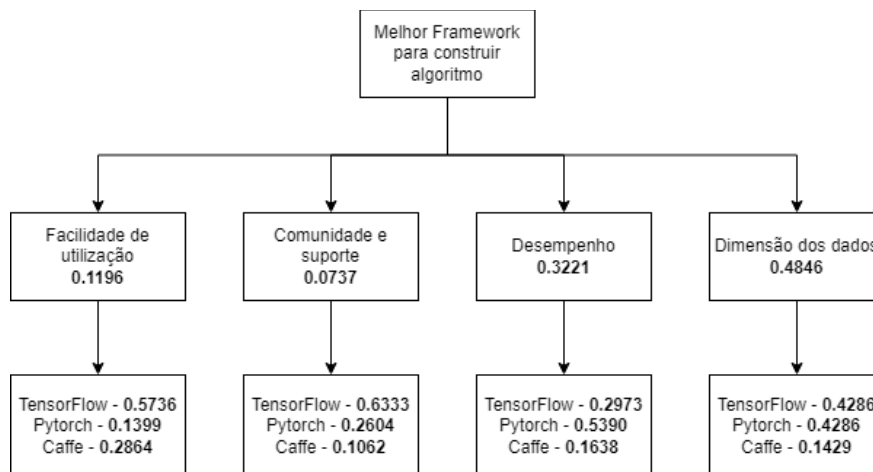


Figura 26 - Árvore de decisão de escolha de framework com prioridades relativas

Por fim, apenas falta calcular a partir das várias prioridades relativas o valor de preferência de cada uma das opções. Sendo assim a seguir são apresentados os valores finais de preferência de cada uma das frameworks enumeradas:

$$\text{TensorFlow: } (0.1196 * 0.5736) + (0.0737 * 0.6333) + (0.3221 * 0.2973) + (0.4846 * 0.4286) = \mathbf{0.4106}$$

$$\text{PyTorch: } (0.1196 * 0.1399) + (0.0737 * 0.2604) + (0.3221 * 0.5390) + (0.4846 * 0.4286) = \mathbf{0.4204}$$

$$\text{Caffe: } (0.1196 * 0.2864) + (0.0737 * 0.1062) + (0.3221 * 0.1638) + (0.4846 * 0.1429) = \mathbf{0.1689}$$

Com estes resultados é então possível concluir que, através da utilização do método de análise hierárquico, a melhor framework a utilizar para o desenvolvimento da aplicação proposta é o PyTorch, ficando ligeiramente atrás da framework TensorFlow.

## 5 Design da Solução

Nesta secção serão abordadas tanto as tecnologias utilizadas durante o desenvolvimento da solução, bem como a arquitetura das diferentes abordagens propostas. É de notar que o presente trabalho surge com o propósito de dar continuidade e melhorar os resultados de um trabalho de mestrado anterior, sendo o título da prova “Detection of cardiac pathology using machine learning” (Feliz, 2022).

### 5.1 Linguagem e Framework de aprendizagem profunda

De acordo com a análise efetuada no capítulo do estado da arte, a linguagem mais popular e mais utilizada atualmente para o desenvolvimento de soluções ligadas à área de inteligência artificial é a linguagem Python (Tiobe - The Software Quality Company, 2023). Deste modo, foi a linguagem escolhida para o desenvolvimento da solução bem como a aposta na utilização de frameworks que apresentam APIs suportadas por esta linguagem.

Segundo o estudo realizado durante a elaboração da secção 4.4, a framework mais indicada para o desenvolvimento da presente investigação seria o PyTorch, em detrimento da framework TensorFlow. Sendo que o trabalho legado foi desenvolvido recorrendo à framework TensorFlow/Keras e a diferença de preferência entre as duas ferramentas foi muito próxima, foi decidido manter a mesma framework durante a continuação da investigação.

No que toca ao tipo de algoritmo de aprendizagem profunda que é utilizado nas diversas alternativas exploradas, a decisão foi recorrer a redes neuronais convolucionais (CNN). Este é a escolha feita uma vez que este tipo de algoritmo é apropriado para a realização de classificações de imagens bem como, uma vez herdadas as várias soluções implementadas anteriormente (Feliz, 2022), continua a ser usado o mesmo tipo de rede.

## 5.2 Dataset disponível

Para o desenvolvimento de um modelo de classificação capaz de classificar sons cardíacos como normais ou anormais, é necessário recorrer a um dataset especializado no problema e a um algoritmo de aprendizagem.

O *dataset* utilizado no trabalho que aqui se descrever foi disponibilizado pela George B. Moody PhysioNet Challenge (Reyna, et al., 2022). Este correspondendo a um conjunto de áudios recolhidos em duas campanhas de rastreio cardiovascular em populações infantojuvenis no estado de Pernambuco, Brasil em 2014 (CC2014) e em 2015 (CC2015) (Oliveira, et al., 2022).

Os dados a utilizar apenas correspondem a 60% do total de dados existentes, devendo estes ser utilizados numa fase inicial para treino dos diversos modelos. Este conjunto de treino corresponde a 3163 áudios recolhidos de 942 pacientes durante a execução das duas campanhas de rastreio. Os outros 40% dos dados foram apenas fornecidos mais tarde, não tendo estes sido utilizados para a elaboração do trabalho. (Reyna, et al., 2022).

Na Tabela 12 e na Tabela 13 consta informação referente às características dos sons cardíacos que foram recolhidos de cada um dos pacientes. Estes são importantes para que fosse realizado um estudo inicial dos diferentes tipo de som e as suas características, de modo a conseguir definir uma estratégia de processamento dos dados existentes.

Tabela 12 - Informação disponível no *dataset* - 1 (Reyna, et al., 2022) (Oliveira, et al., 2022)

Variável	Descrição	Valores Possíveis
Sopro	Indica presença de sopro	<b>Present (19.5%)</b> – Ondas de sopro presentes em pelo menos 1 áudio <b>Absent (73.0%)</b> – Ondas de sopro não presente em nenhum áudio <b>Unknown (7.5%)</b> – Presença ou ausência não é clara.
Localização do sopro	Indica qual o ponto de auscultação onde pelo menos um sopro foi observado.	<b>PV</b> – Válvula pulmonar <b>TV</b> – Válvula tricúspide <b>AV</b> – Válvula aórtica <b>MV</b> – Válvula Mitral <b>Phc</b> – Outra localização
Localização mais audível	Indica o ponto de auscultação com a onda de sopro mais audível	<b>PV, TV, AV, MV, Phc</b>
Período do sopro	Indica o período sistólico em que ocorre o sopro	<b>Sistólica</b> – Early, Mid, Late, Holosystolic <b>Diastólica</b> – Early, Mid, Holosystolic

Tabela 13 - Informação disponível no dataset – 2 (Reyna, et al., 2022) (Oliveira, et al., 2022)

Variável	Descrição	Valores Possíveis
Formato do sopro	Descreve o formato do sopro que pode ser descrito como intensidade ao longo do tempo	<b>Sistólica</b> – Crescendo, Decrescendo, Diamond, Plateau <b>Diastólica</b> – Decrescendo, Plateau
Tom do sopro	Indica o gradiente de pressão sentido nas câmaras do coração	<b>High, Medium, Low</b>
Gradiente do sopro	Descreve a intensidade do sopro de acordo com a escala de Levine (Kazemnejad, et al., 2021)	<b>Grade I/VI:</b> Se praticamente inaudível e não presente ou não registado em todos os pontos de auscultação <b>Grade II/VI:</b> Suave, mas facilmente audível em todos os pontos <b>Grade III/VI:</b> Moderadamente alto ou alto.
Qualidade do sopro	Descreve a qualidade das ondas observadas	<b>Sistólica:</b> Blowing, Harsh, Musical <b>Diastólica:</b> Blowing, Harsh
Resultado	Indica a avaliação efetuado por um cardiologista	<b>Normal, Abnormal</b>

É importante ainda salientar que durante a criação dos conjuntos de treino, validação e teste os vários pacientes foram subdivididos de modo a que áudios do mesmo paciente não sejam colocados em grupos diferentes.

## 5.3 Abordagem

Existem diversas alternativas para a classificação de sons cardíacos com a finalidade classificar o paciente final como apresentando um batimento cardíaco normal ou anormal.

Sendo assim, foram utilizadas 2 abordagens distintas que servem como base para a classificação dos diversos áudios presentes no *dataset* referido na secção 5.2.

### 5.3.1 Abordagem 1

A primeira abordagem, contemplada na Figura 27, começa pela recolha de dados provenientes do *dataset* PhysioNet 2022 e culmina com a avaliação final do modelo treinado a partir destes mesmos dados.

O processo inicia-se com a realização de um pré-processamento onde são analisados, tratados e são selecionados atributos essenciais para a classificação de cada um dos pacientes. De seguida, é realizado o processo de extração de atributos com o intuito de obter um formato mais apropriado para ser interpretado pelo algoritmo de aprendizagem automática utilizado.

Posteriormente, é realizada a partição dos dados onde é realizada a separação dos dados em dados para o treino do modelo e dados para realização de testes.

De seguida, os atributos previamente processados são fornecidos ao algoritmo de classificação existente de modo a criar um modelo que consiga generalizar corretamente qualquer ficheiro de áudio referente ao batimento cardíaco de um paciente.

Por fim, é realizada a avaliação do modelo criado, a partir do conjunto de testes referido anteriormente, através da recolha e análise das várias medidas de avaliação exploradas na secção 7.2

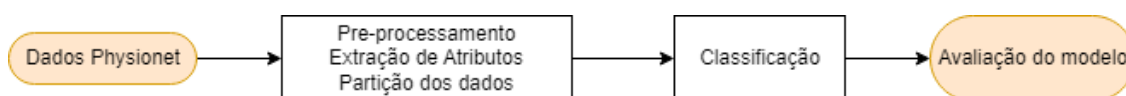


Figura 27 - Abordagem de desenvolvimento 1

### 5.3.2 Abordagem 2

No que refere à segunda abordagem, grande parte do processo é semelhante ao descrito na primeira abordagem à exceção de dois pontos, tal como se pode observar pela Figura 28.

Em primeiro lugar, a partição dos dados deixa de ocorrer apenas em dados de treino e dados de teste, sendo necessário ainda um conjunto de dados de validação de modo a ser possível avaliar o desenvolvimento no final de cada época de treino.

Em segundo lugar, é utilizado um algoritmo de Rede Neuronal Artificial para realizar a previsão da classificação a atribuir a cada paciente. Esta rede no caso do presente documento poderá ser uma CNN ou um modelo pré-treinado.

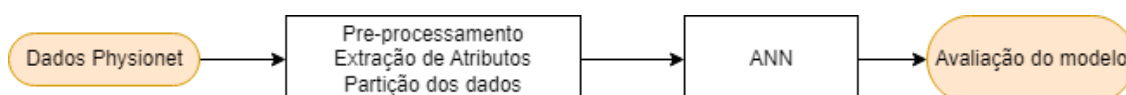


Figura 28 - Abordagem de desenvolvimento 2

## 5.4 Pré-processamento dos dados

Com o intuito de conseguir recolher informações relevantes dos sons presentes na base de dados utilizada, é necessário efetuar pré-processamento destes sons de modo a facilitar e aprimorar a aprendizagem efetuada pelo algoritmo utilizado. No presente problema, são utilizados sons cardíacos que apresentam não só um elevado grau de ruído como também apresentam diferentes tamanhos.

Deste modo, os passos de pré-processamento serão:

- Extração de ruído no segmento de áudio correspondente aos pontos de auscultação recolhidos.
- Segmentação dos áudios em segmentos de três segundos de modo a garantir que todos tenham a mesma dimensão.
- Exclusão de sons provenientes de pacientes em fase de gravidez (devido à possibilidade de múltiplos sons cardíacos) e sons em que a presença de sopro é desconhecida.

#### **5.4.1 Segmentação dos áudios**

No que toca à segmentação dos diversos ficheiros de áudio presentes na base de dados são utilizadas três abordagens distintas para o processamento dos dados.

Na primeira abordagem, são apenas divididos os ficheiros de áudio em segmentos de 3 segundos, sendo possivelmente descartado o último segmento se este não atingir uma duração de 3 segundos. Deste modo, conseguimos garantir que não existem segmentos de áudio que contêm o mesmo sinal sonoro e que são extraídos o maior número de segmentos possível.

A segunda abordagem, à semelhança da primeira, pretende mais uma vez segmentar o ficheiro de áudio em vários segmentos de 3 segundos apresentado a nuance de todos os segmentos recolhidos começarem no sinal S1. O primeiro segmento começa a partir do primeiro sinal S1, apresentando uma duração de 3 segundos. O segundo segmento começa no primeiro sinal S1 que ocorra num período imediatamente a seguir ao primeiro segmento criado. Isto permite criar um padrão (todos os segmentos de 3 segundos criados começam na mesma fase do ciclo cardíaco), que facilita ao algoritmo a análise dos segmentos criados.

A terceira abordagem pressupõe a segmentação dos diversos ficheiros de áudio em todos os segmentos de 3 segundos que comecem no sinal S1, mesmo que estes estejam sobrepostos. Ao contrário da segmentação anterior, o segundo segmento começa no segundo sinal S1 presente no áudio e não no primeiro sinal imediatamente a seguir ao primeiro segmento. Isto implica que apesar de existir uma maior quantidade de segmentos criados, todos eles vão incluir partes de outros segmentos.

### **5.5 Extração de atributos**

Após a conclusão da fase de processamento dos dados, segue-se a modelação desses dados, a fim de viabilizar a capacidade do algoritmo em detetar os padrões que servirão de base à classificação de cada um dos dados recebidos. Neste sentido, a extração de atributos assume a responsabilidade de transformar os dados processados em informação passível de ser apresentada ao algoritmo de aprendizagem.

A escolha das abordagens de extração de dados desempenha um papel crítico nesse processo, pois determina a qualidade e a relevância das informações disponibilizadas ao algoritmo. Neste contexto, optamos por utilizar duas abordagens distintas: o espectrograma de mel (Lu, et al., 2022) e o MFCC (Nogueira, et al., 2019). A seleção dessas técnicas baseou-se na comprovada eficácia na extração de atributos de áudio (Muqing, et al., 2020), garantindo uma representação robusta dos dados que possibilita a detecção precisa de padrões.

O espectrograma de mel e o MFCC são amplamente reconhecidos pela capacidade de capturar informações essenciais relacionadas com a frequência e a forma do sinal de áudio, tornando-os ideais para aplicações de processamento de áudio e reconhecimento de padrões. No entanto, é importante destacar que, após a extração desses atributos, é indicado a aplicação de uma etapa adicional de normalização (Jo, 2019).

A normalização dos valores é realizada para ajustar vetores de valores para uma escala consistente, o que pode melhorar o desempenho dos algoritmos de aprendizagem, especialmente aqueles sensíveis à escala dos dados. Essa normalização não só facilita a comparação e o processamento dos dados, como também contribui para uma representação mais eficaz no contexto do nosso sistema.

A escolha criteriosa dessas técnicas visa assegurar a precisão e a confiabilidade do nosso sistema, desempenhando um papel fundamental na consecução dos nossos objetivos de análise de dados.

Em suma, a extração de atributos, por meio da abordagem do espectrograma de mel e do MFCC, representa um elemento crítico na nossa metodologia, permitindo que os dados processados sejam transformados em informações pertinentes para o algoritmo de aprendizagem.

## **5.6 Partição dos dados**

Após a conclusão da etapa de processamento de dados, segue-se a sua partição em grupos para o posterior treino dos modelos gerados pelos diversos algoritmos utilizados neste documento.

Na primeira abordagem, conforme detalhado na secção 5.3.1, optou-se por dividir o conjunto de dados em duas partes: 70% dos dados são destinados ao treino, enquanto os restantes 30% são reservados para testes.

Na segunda abordagem, mencionada na secção 5.3.2, adotou-se uma abordagem diferente. Além da divisão em conjunto de treino e teste, também foi incluído um conjunto de validação para avaliar o desempenho do modelo em diferentes fases de aprendizagem. Neste caso, o conjunto de testes anteriormente mencionado é subdividido em dois grupos, cada um compreendendo 15% do conjunto de dados total.

É fundamental salientar que, em todas as divisões realizadas, foi assegurado rigorosamente que a proporção de dados relacionados com a presença de pacientes com batimentos cardíacos normais e anormais permanece constante em todos os conjuntos. Esta abordagem é crucial para garantir a imparcialidade e a precisão dos modelos resultantes, permitindo uma correta generalização por parte dos algoritmos.

Estas divisões estratégicas dos dados são essenciais para avaliar o desempenho dos modelos de forma fiável e fornecem uma valiosa compreensão para a análise final dos resultados do estudo.

## **5.7 Alternativas de Design**

De modo a conseguir perceber qual a melhor abordagem para a resolução do problema desta dissertação, foram ponderadas duas alternativas de implementação para a classificação dos pacientes. Estas alternativas diferem não só na quantidade de modelos necessários para a classificação de um paciente, como também a forma dos dados utilizada para o treino destes mesmos modelos.

Como ponto de partida, é apresentada também a alternativa que corresponde ao trabalho desenvolvido anteriormente (Feliz, 2022), sendo de seguida apresentadas duas alternativas diferentes à implementação da solução.

### **5.7.1 Alternativa 1 – Proposta Legado**

Tal como referido neste capítulo, esta primeira alternativa resulta de uma abordagem explorada anteriormente (Feliz, 2022). Esta abordagem consiste em criar um modelo de aprendizagem profunda por cada um dos pontos de auscultação existentes. Estes modelos, tal como apresentado na Figura 29, devem ser capazes de classificar o som do ponto de auscultação específico para o qual foram treinados, obtendo assim uma resposta sobre a presença de anomalias no som analisado. Depois de obtida uma classificação para cada ponto de auscultação, as várias classificações serão analisadas por um classificador geral que terá como finalidade a deteção de padrões para a classificação final de anomalia ou não no paciente analisado.

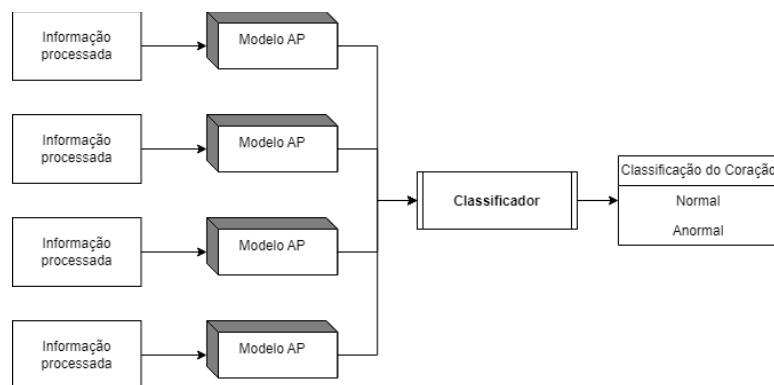


Figura 29 - Arquitetura do sistema alternativa 1 – proposta legado

Vantagens:

- É criado um modelo para cada um dos pontos de auscultação simplificando o processo de implementação e melhorando a eficácia para esse ponto.

Desvantagens:

- Necessidade de mais tempo para implementar múltiplos modelos.
- O classificador pode reduzir a importância de um ponto de auscultação, podendo induzir em falsos negativos.

### 5.7.2 Alternativa 1 – Nova Proposta

Apesar desta abordagem ser bastante semelhante à descrita na secção 5.7.1, o objetivo desta proposta é utilizar a mesma abordagem definida na alternativa 1, mas descartando a utilização do classificador final para a classificação de anomalias no paciente, como apresentado na Figura 30. Sem a utilização do classificador geral, um paciente é classificado como anormal caso pelo menos um dos modelos tenha classificado o seu ponto de auscultação como anormal. Para que um paciente seja classificado como “Normal”, o resultado dos modelos dos quatro pontos de auscultação tem de ser “Normal”.

Tal alternativa provém da ideia de que é preferível a realização de estudos mais a fundo do paciente, mesmo podendo este não ter qualquer problema do que o paciente apresentar de facto uma anomalia, mas ser finalmente classificada como um falso negativo.

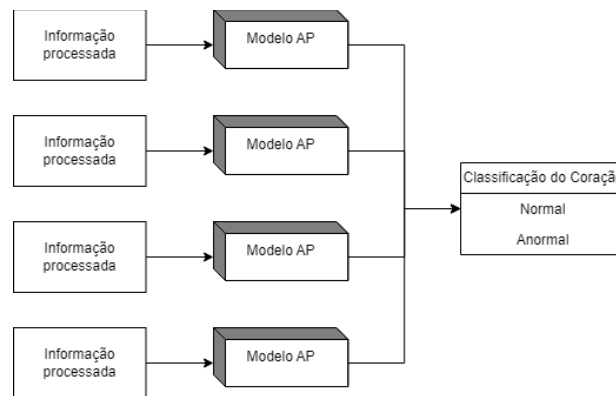


Figura 30 - Arquitetura do sistema alternativa 1 – nova proposta

Vantagens:

- Apresenta uma maior segurança para o paciente.
- Reduz alguma complexidade da classificação final permitindo perceber os resultados mais facilmente.

Desvantagens:

- Poderá incorrer num maior número de falsos positivos.
- Descartada qualquer possibilidade de detecção de padrão nos resultados obtidos pelos diferentes pontos de auscultação.

### 5.7.3 Alternativa 2 – Proposta Legado

Uma outra alternativa elaborada durante a prova de tese a dar seguimento (Feliz, 2022) é a utilização de apenas um modelo para a leitura e classificação do paciente com base nos vários sons provenientes dos diferentes pontos de auscultação. Esta proposta poderá levar a um aumento do tempo necessário para desenvolver um modelo mais complexo de classificação, não havendo, no entanto, dependências de resultados provenientes de vários modelos diferentes. A arquitetura desta alternativa está apresentada na Figura 31, começando com 4 sons diferentes e terminando com a classificação do paciente

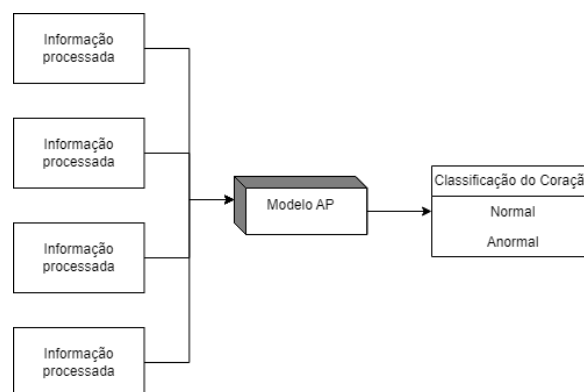


Figura 31 - Arquitetura do sistema alternativa 2 - proposta legado

Vantagens:

- Mais fácil de analisar os resultados obtidos.
- Menor complexidade temporal devido a apenas ser necessário aplicar um modelo.

Desvantagens:

- Aumento da complexidade de implementação.

## 6 Implementação

Nesta secção é explorada a implementação que foi preconizada para solução do problema. Para tal, é efetuada a descrição da implementação do pré-processamento e extração de atributos dos diversos ficheiros de áudio presentes no *dataset*, bem como a implementação das duas alternativas idealizadas.

### 6.1 Pré-processamento

Numa fase inicial, é necessário realizar o pré-processamento dos dados provenientes do *dataset* utilizado, de modo a garantir um nível de aprendizagem relevante por parte dos modelos.

Conforme referido na secção 5.4, este processo compreende a melhoria dos dados através da remoção do ruído presente nos segmentos de som, a exclusão de dados sons provenientes de pacientes que se encontrem num estado de gravidez ou que não se conheço a classificação do sopro e a segmentação dos ficheiros de áudio segundo diversas abordagens de modo a serem exploradas posteriormente.

#### 6.1.1 Remoção de Ruído

De modo a conseguir treinar corretamente os modelos desenvolvidos, tal como referido na secção 5.4, procedeu-se ao processamento dos ficheiros de áudio de modo a preparar os dados para a posterior recolha de atributos. Este processamento ocorre primeiro pela aplicação de um filtro passa-banda fase-zero com frequências de corte entre os 50Hz e os 800Hz. Tal foi decidido a partir da análise de ficheiros de áudio, recorrendo à ferramenta Audacity (AudacityTeam, s.d.), que permitiram validar as frequências com maior relevância para o problema em questão. Também foi verificado a utilização deste filtro na revisão da literatura sendo, no entanto, utilizado outro intervalo de frequências (Patwa, et al., 2023).

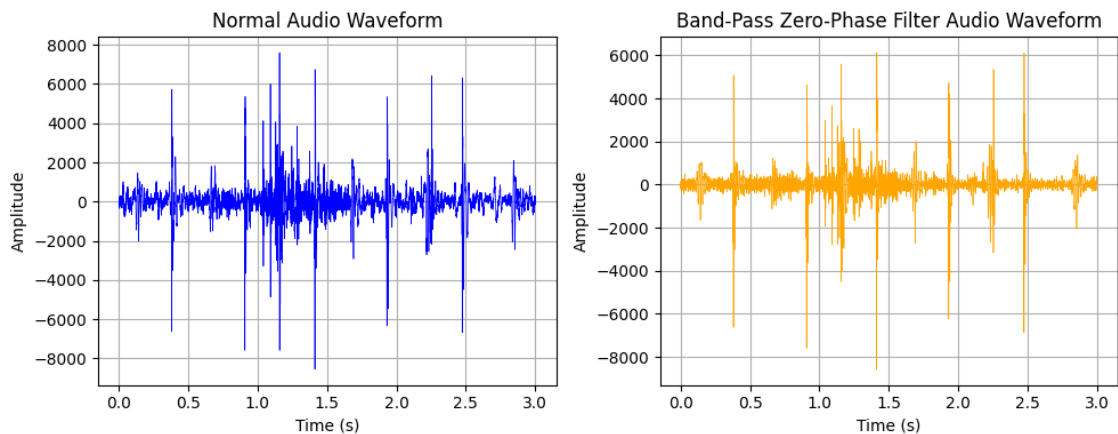


Figura 32 – Exemplo de onda de áudio normal e onda com aplicação de filtro

Na Figura 32 é possível observar dois gráficos que representam a forma da onda de áudio de um determinado segmento de um batimento cardíaco. No gráfico da esquerda está representado um exemplo de uma onda normal. No gráfico da direita é possível observar o mesmo segmento de áudio presente no gráfico da esquerda, mas ao qual foi aplicada uma filtragem para remoção do ruído.

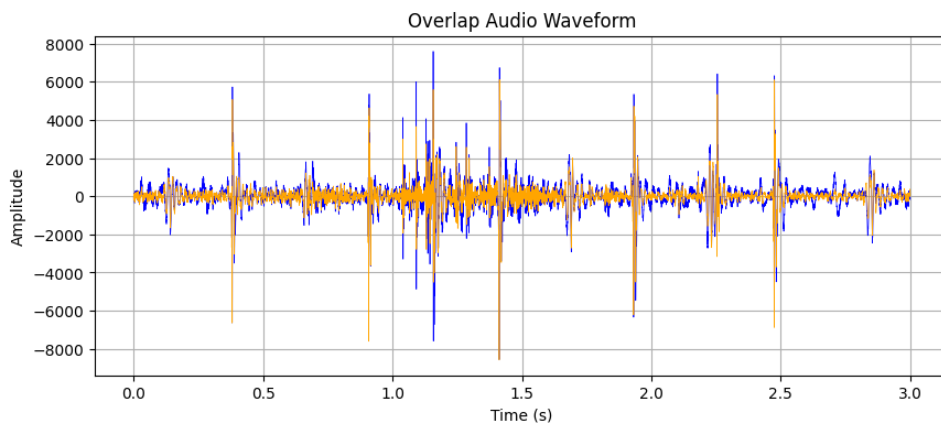


Figura 33 - Comparação das ondas

Através da comparação de ambas as ondas, é possível observar na Figura 33, uma diferença substancial na amplitude que as duas ondas apresentam. De modo a verificar esta alteração, cada uma das ondas foi novamente convertida num sinal sonoro com a finalidade de interpretar o novo som gerado. Como tal, foi possível constatar que existe menos ruído no áudio onde foi aplicado o filtro, sendo assim mais audíveis os diversos batimentos cardíacos.

### 6.1.2 Segmentação dos áudios

Após a remoção do ruído do sinal de áudio, torna-se necessário proceder à segmentação do sinal em segmentos de dimensão uniforme. Este procedimento visa não só aumentar o conjunto de dados disponíveis para o desenvolvimento de algoritmos, mas também garantir

que todas as partições possuam as dimensões necessárias para serem interpretadas pelo modelo de inteligência artificial.

Tal como referido na secção 5.4.1, a primeira abordagem utilizada consiste na tentativa de subdividir o áudio no maior número de segmentos possíveis que não se sobreponham e que consigam atingir uma duração de 3 segundos. Os ficheiros disponíveis na Physionet são ficheiros de áudio “.wav” sendo assim necessário recorrer à classe AudioSegment da biblioteca pydub para a sua leitura, como apresentado na Figura 34. Cada um dos segmentos é finalmente convertido num vetor de modo a ser posteriormente persistido como uma imagem.

```
def read_audio_file_duration_segment(audio_path, sr, duration):
    segments = []
    file_duration = librosa.get_duration(filename=audio_path)

    if file_duration > duration:
        myaudio = AudioSegment.from_file(audio_path, "wav", sr=4000)
        myaudio = myaudio[1000:]
        chunk_length_ms = duration * 1000
        chunks = make_chunks(myaudio, chunk_length_ms)

        for i, chunk in enumerate(chunks):
            if len(chunk) == len(chunks[0]):
                segments.append(audiosegment_to_ndarray(chunk))
    else:
        segments = None
    return segments
```

Figura 34 - Segmentação dos ficheiros de áudio

No que refere às restantes 2 abordagens de segmentação, foi necessário recorrer ainda a um ficheiro “.tsv” existente por cada ficheiro de áudio relacionado com os diversos pacientes. Este ficheiro contém *metadata* do ficheiro de áudio correspondente ao intervalo de tempo em que ocorre cada uma das fases do batimento cardíaco.

Tabela 14 – Exemplo de ficheiro “.tsv”

Início	Fim	Fase do Batimento
0	0.1085	0 (Sem Anotação)
0.1085	0.240176	1 (Onda S1)
0.240176	0.360176	2 (Período sistólico)
0.360176	0.500176	3 (Onda S2)
0.500176	0.700176	4 (Período diastólico)
0.700176	0.820176	1
0.820176	0.920176	2
0.920176	1.040176	3
1.040176	1.240176	4
...	...	...
5.580176	6.592	0

De modo a garantir que todos os segmentos comecem a partir da mesma fase do batimento cardíaco, foi implementada uma nova função que visa recolher todos os segmentos que apresentam a fase do batimento “1” que corresponde ao impulso da onda S1 produzida pelo coração, como indicado na Tabela 14. Deste modo, torna-se mais fácil para o modelo gerado interpretar cada um dos sons, não sendo necessário que este compreenda quando ocorre cada uma das fases do batimento cardíaco.

```
def read_tsv_file(filepath):
    with open(filepath) as file:
        tsv_file = csv.reader(file, delimiter="\t")
        tsv_file = list(filter(lambda x: x[2] == '1', tsv_file))
        tsv_file = list(map(lambda x: [float(x[0]) * 1000, float(x[1]) * 1000, x[2]], tsv_file))
    return tsv_file
```

Figura 35 - Leitura de ficheiro ".tsv"

A função implementada lê a informação existente no ficheiro “.tsv” e retorna uma lista com todos os intervalos do ficheiro de áudio em que ocorre o impulso S1, tal como é possível observar na Figura 35. Cada um dos intervalos é também convertido de segundos para milissegundos de modo a ser realizada uma correta extração dos diversos segmentos.

A terceira abordagem promove um *oversample* natural dos dados, sem necessitar recorrer a técnicas de *oversample* sintético, enquanto a segunda exclui a possibilidade de existir informação redundante nos segmentos.

Na Tabela 15 é apresentada a distribuição de amostras existentes para cada uma das abordagens de segmentação. Como podemos verificar, apesar das abordagens serem diferentes, a sua proporção mantém-se bastante semelhante em todas elas, variando apenas a quantidade de amostras existente para cada uma das classes.

Tabela 15 - Distribuição das amostras de classes por abordagem de segmentação

Classes	Abordagem		
	1ª	2ª	3ª
<b>Present</b>	3216 (17%)	1879 (18%)	9485 (19%)
<b>Absent</b>	15215 (83%)	8308 (82%)	41155 (81%)
<b>Total</b>	18431 (100%)	10187 (100%)	50640 (100%)

## 6.2 Extração de Atributos

Após a conclusão do pré-processamento, procede-se à extração de atributos. No caso do presente trabalho, esta fase inicia-se pela conversão do vetor correspondente ao sinal sonoro em espectrogramas de mel ou em MFCCs. Para a recolha destes atributos, foi utilizada a biblioteca “Librosa” recorrendo às respetivas funções de “melspectrogram” e “mfcc”.

Por fim, em cada um dos casos, os valores dos atributos foram normalizados para o intervalo de valores [0, 1], com o objetivo de facilitar ao modelo criado ajustar os seus parâmetros durante a fase de treino do mesmo.

### 6.2.1 Espectrograma de Mel

Para a obtenção do espectrograma de mel a partir de um sinal sonoro, foi utilizado o excerto de código da Figura 36 para a sua conversão.

```
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)

    scaler = MinMaxScaler()

    mel_spectrogram_scaled = numpy.mean(mfcc.T, axis=0)
    scaler.fit(mel_spectrogram_scaled)
    mel_spectrogram_normalized = scaler.transform(mel_spectrogram_scaled)

    return mel_spectrogram
```

Figura 36 - Criação do espectrograma de mel

Com este código, é possível validar a criação do espectrograma, terminando com a normalização dos dados recorrendo à biblioteca “Sklearn” para normalizar os dados do vetor. Na Figura 37 é possível verificar o resultado da extração de atributo de um espectrograma de mel segundo a aplicação do filtro abordado na secção 6.1.1.

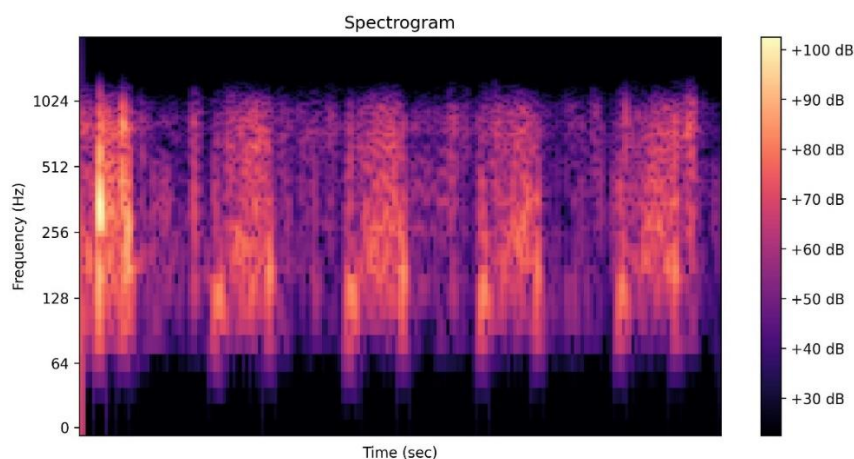


Figura 37 - Exemplo de espectrograma de mel gerado

## 6.2.2 MFCC

O segundo atributo recolhido a partir dos áudios de cada paciente foram MFCCs. De modo a gerar o atributo MFCC, foi elaborado o código apresentado na Figura 38.

```
def get_mfcc(x, sr):
    hop_length = int(numpy.round(0.01 * sr))
    sgram = librosa.stft(filtered_segment, n_fft=256, hop_length=hop_length)

    sgram_mag, _ = librosa.magphase(sgram)
    mfcc = librosa.feature.mfcc(y=numpy.swapaxes(sgram_mag, 0, 1), sr=sr, n_mels=10)

    scaler = MinMaxScaler()

    mfccScaled = numpy.mean(mfcc.T, axis=0)
    scaler.fit(mfccScaled)
    mfccs_normalized = scaler.transform(mfccScaled)

    return mfccs_normalized
```

Figura 38 - Criação do MFCC

À semelhança da criação do espectrograma de mel, é utilizada a biblioteca “Librosa” para gerar a imagem MFCC correspondente ao som recebido, sendo por fim realizada uma normalização dos valores no intervalo [0, 1] de modo a facilitar o ajuste de valores do algoritmo de aprendizagem durante a fase de treino.

O resultado é apresentado na Figura 39, sendo correspondente a um MFCC com 10 coeficientes ao longo de um intervalo de 3 segundos.

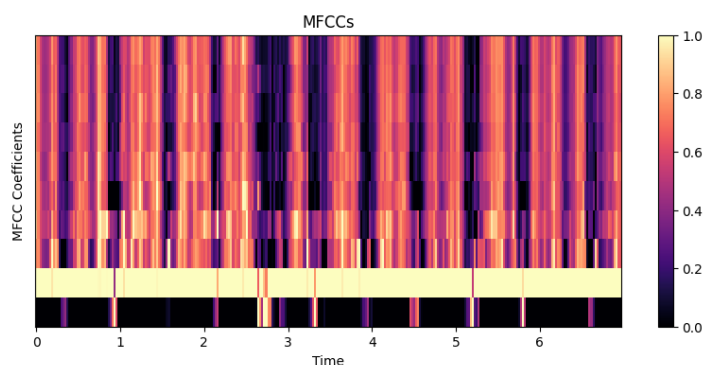


Figura 39 - Exemplo de MFCC

## 6.3 Alternativa de Implementação

Terminada a fase da extração dos atributos, é necessário agora compreender de que modo foi solucionado o problema descrito ao longo do documento. Como tal, e de acordo com o que foi descrito durante a secção 5.7, é descrita a implementação das duas novas alternativas de design e os passos comuns para a construção das várias abordagens utilizadas.

### 6.3.1 Construção de Alternativas

A construção de alternativas foi pensada não só de modo a facilitar a construção e correção dos diversos modelos de aprendizagem criados, mas também para facilitar a compreensão do leitor no processo utilizado para a conceção dos modelos de previsão.

Numa fase inicial, é necessário seleccionar os dados correspondentes aos diferentes pontos de auscultação, sendo ainda necessário fazer o agrupamento dos dados em diferentes conjuntos face ao ponto de auscultação associado de modo a facilitar a sua aplicação nas restantes fases.

Para tal, foi criada uma função que foi responsável por receber os dados relativos aos vários segmentos de áudio criados e gera os conjuntos de dados essenciais para cada um dos modelos de aprendizagem utilizados, tal como apresentado na Figura 40.

```

def generate_data_partitions(n):
    m1_df_train, m1_df_test, m1_df_val = get_alt1_train_test_val_without_unknown(n)

    m1_AV_df_train = get_extracted_features_of_location(m1_df_train, 'AV', "AV_TRAIN")
    ...
    m1_MV_df_train = get_extracted_features_of_location(m1_df_train, 'MV', "MV_TRAIN")

    m1_AV_df_test = get_extracted_features_of_location(m1_df_test, 'AV', "AV_TEST", True)
    ...
    m1_MV_df_test = get_extracted_features_of_location(m1_df_test, 'MV', "MV_TEST", True)

    m1_AV_df_val = get_extracted_features_of_location(m1_df_val, 'AV', "AV_VAL")
    ...
    m1_MV_df_val = get_extracted_features_of_location(m1_df_val, 'MV', "MV_VAL")

    m1_AV_train_X, m1_AV_train_y, m1_AV_test_X, m1_AV_test_y, m1_AV_val_X, m1_AV_val_y = get_x_y_without_unknown(m1_AV_df_train, m1_AV_df_test, m1_AV_df_val)
    ...
    m1_MV_train_X, m1_MV_train_y, m1_MV_test_X, m1_MV_test_y, m1_MV_val_X, m1_MV_val_y = get_x_y_without_unknown(m1_MV_df_train, m1_MV_df_test, m1_MV_df_val)

    m1_AV_train_X = fix_arrays(m1_AV_train_X)
    m1_AV_test_X = [m1_AV_test_X[0], fix_arrays(m1_AV_test_X[1])]
    m1_AV_val_X = fix_arrays(m1_AV_val_X)
    ...
    m1_MV_val_X = fix_arrays(m1_MV_val_X)

    AV_data = [m1_AV_train_X, m1_AV_train_y, m1_AV_test_X, m1_AV_test_y, m1_AV_val_X, m1_AV_val_y]
    ...
    MV_data = [m1_MV_train_X, m1_MV_train_y, m1_MV_test_X, m1_MV_test_y, m1_MV_val_X, m1_MV_val_y]

    return AV_data, PV_data, TV_data, MV_data

```

Figura 40 - Exemplo de partição da informação de cada foco

Com esta função, é possível não só garantir que estão a ser gerados os conjuntos de treino e teste para cada um dos modelos de treino a utilizar, como também facilita a divisão dos mesmos por cada um dos focos de auscultação. No caso de ser uma rede neuronal, esta função está ainda encarregue de gerar o conjunto de validação que é utilizado durante o processo de treino do modelo.

De modo a simplificar o processo de criação dos modelos, foi concebida uma função genérica para a criação e compilação dos modelos, sendo esta utilizada por todos os focos de auscultação. Na Figura 41, está presente um exemplo desta abordagem sendo assim possível ajustar os parâmetros dos modelos de igual modo para todos focos.

```

def build_model():
    model = Sequential()

    model.add(Conv2D(filters=16, kernel_size=3, strides=1, padding="same", activation="relu", input_shape=(301, 10, 1)))
    model.add(MaxPooling2D())
    model.add(BatchNormalization())

    model.add(Conv2D(filters=32, kernel_size=3, strides=1, padding="same", activation="relu"))
    model.add(MaxPooling2D())
    model.add(BatchNormalization())

    model.add(Conv2D(filters=16, kernel_size=3, strides=1, padding="same", activation="relu"))
    model.add(MaxPooling2D())
    model.add(BatchNormalization())

    model.add(Flatten())

    model.add(Dense(512, activation="relu"))
    model.add(Dense(256, activation="relu"))
    model.add(Dense(128, activation="relu"))
    model.add(Dense(1, activation="sigmoid"))

    metrics = ['accuracy', tf.keras.metrics.Precision(class_id=0), tf.keras.metrics.Recall(class_id=0)]

    model.compile('adam', loss=tf.losses.BinaryCrossentropy(), metrics=metrics)

    return model

```

Figura 41 - Exemplo de construção de modelo CNN

Neste exemplo, para cada um dos modelos criados, são adicionadas 3 camadas de convolução representadas pelas funções de convolução, pooling e normalização e pelas camadas totalmente conectadas representadas pelas funções “Dense”. Finalmente, são definidas as

métricas a usar durante a fase de aprendizagem e é compilado o modelo, identificando o otimizador e a função de perda a utilizar.

Terminada a construção dos diversos modelos é necessário agora ajustar os dados recolhidos ao modelo que foi gerado. Para tal, é criada uma função responsável por treinar o modelo gerado a partir dos dados recebidos de modo a obter um modelo capaz de prever a existência de um sopro através da análise da imagem de um som cardíaco.

```
def train_model(model, train_X, train_y, val_X, val_y, batch_size, epochs, class_weights, save_path, n):
    f1_checkpoint = F1ScoreCheckpoint(validation_data=(val_X, val_y))

    # Create checkout
    checkout_a = ModelCheckpoint(f'{save_path}/checkout_{n}_a.hdf5', save_best_only=True, save_weights_only=True, monitor='val_accuracy', verbose=1, mode='max')
    checkout_p = ModelCheckpoint(f'{save_path}/checkout_{n}_p.hdf5', save_best_only=True, save_weights_only=True, monitor='val_precision', verbose=1, mode='max')
    checkout_r = ModelCheckpoint(f'{save_path}/checkout_{n}_r.hdf5', save_best_only=True, save_weights_only=True, monitor='val_recall', verbose=1, mode='max')
    checkout_l = ModelCheckpoint(f'{save_path}/checkout_{n}_l.hdf5', save_best_only=True, save_weights_only=True, monitor='val_loss', verbose=1, mode='min')
    checkout_f = F1ScoreCheckpoint(validation_data=(val_X, val_y), save_path=f'{save_path}/checkout_{n}_f.hdf5')

    # 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, checkout_l, checkout_f])

    # Save model
    model.load_weights(f'{save_path}/checkout_{n}_a.hdf5')
    model.save(f'{save_path}/trained_{n}_a.h5')

    model.load_weights(f'{save_path}/checkout_{n}_p.hdf5')
    model.save(f'{save_path}/trained_{n}_p.h5')

    model.load_weights(f'{save_path}/checkout_{n}_r.hdf5')
    model.save(f'{save_path}/trained_{n}_r.h5')

    model.load_weights(f'{save_path}/checkout_{n}_l.hdf5')
    model.save(f'{save_path}/trained_{n}_l.h5')

    model.load_weights(f'{save_path}/checkout_{n}_f.hdf5')
    model.save(f'{save_path}/trained_{n}_f.h5')
```

Figura 42 - Exemplo de função de treino

Para além do treino do modelo, são também adicionados “callbacks” que monitorizam métricas associados ao treino do modelo durante as várias épocas de treino, tal como apresentado na Figura 42. Estas métricas (verificar secção 7.2.2) são a exatidão, a precisão, o recall, a perda e o f1-score associados à capacidade do modelo em prever corretamente os dados presentes no conjunto de teste (validação no caso de NN).

Por fim, é realizada a avaliação dos modelos treinados através da utilização do conjunto de teste nos diversos modelos. Deste modo, é possível conseguir relacionar as previsões realizadas pelo modelo com o valor real correspondente a cada paciente. É assim possível criar uma matriz de confusão (verificar secção 7.2.1) que permite avaliar corretamente o desempenho final dos modelos criados, tal como se pode observar na Figura 43.



Figura 43 - Matriz de confusão e métricas de avaliação dos modelos

Tal como nas fases anteriores, foi desenvolvida uma função que recebe como parâmetro as previsões geradas pelos modelos, bem como o valor original de cada paciente sendo retornado um conjunto de valores necessário à construção de uma matriz de confusão (secção 7.2.1). Na Figura 44, é possível verificar como é realizada a comparação dos diversos valores reais com os previstos pelo modelo e como é construída a matriz de confusão.

```
def getConfusionMatrix(results):
    present_present = 0
    present_absent = 0
    absent_present = 0
    absent_absent = 0
    unknown_present = 0
    unknown_absent = 0

    for result in results:
        prediction, original = result[0], result[1]

        if original == 0:
            if prediction == 0:
                present_present = present_present + 1
            if prediction == 1:
                present_absent = present_absent + 1
        if original == 1:
            if prediction == 0:
                absent_present = absent_present + 1
            if prediction == 1:
                absent_absent = absent_absent + 1

    return [[present_present, present_absent], [absent_present, absent_absent]]
```

Figura 44 - Criação de matriz de confusão

### 6.3.2 Alternativa 1 – Nova Proposta

A nova proposta de implementação, face ao desenvolvido previamente, pressupõe a criação de um modelo de aprendizagem para cada um dos pontos de auscultação correspondentes aos diferentes focos do coração. Para tal, os diversos segmentos de áudios criados, de acordo com o descrito na secção 6.1.2, são agrupados pelo seu foco de auscultação para posteriormente serem utilizados nos respetivos modelos.

Uma vez que esta abordagem exige a criação de um modelo por cada foco de auscultação, é ainda criada outra função que permite validar para cada modelo as melhores métricas atingidas durante a fase de treino, como apresentado na Figura 45.

```
Best accuracy epoch model:
11/11 [=====] - 0s 2ms/step
```

	precision	recall	f1-score	support
Present	0.55	0.40	0.46	68
Absent	0.85	0.92	0.88	262
accuracy			0.81	330
macro avg	0.70	0.66	0.67	330
weighted avg	0.79	0.81	0.80	330

Figura 45 - Métricas de desempenho de modelo

Nesta alternativa, cada modelo está responsável por classificar cada amostra segundo as classes “presente” e “absent” sendo estas respetivamente a classe positiva e negativa. A classificação final do paciente é dada pela existência de pelo menos um dos resultados que atribua a classificação de “present” à amostra analisada.

Terminada a construção dos 4 modelos, é necessário classificar cada um dos focos de auscultação dos pacientes pertencentes ao conjunto de teste. Para tal, é realizado um pós-processamento da classificação realizada por cada modelo recorrendo à função apresentada na Figura 46.

```
def classify_patients(test_patients, models, m1_AV_test_X):
    av_model, pv_model, tv_model, mv_model = models
    results = []
    for patient in test_patients:
        AV_lines = m1_AV_test_X[m1_AV_test_X[0].Patient_ID == patient]
        PV_lines = m1_PV_test_X[m1_PV_test_X[0].Patient_ID == patient]
        ...
        all_results = []
        all_original_values = []

        AV_result = 0
        AV_True_Label = None
        PV_result = 0
        PV_True_Label = None
        ...

        for index, row in AV_lines.iterrows():
            features = row.Features
            true_result = np.argmax(row.Results);
            AV_True_Label = true_result

            prediction = av_model.predict(features)
            prediction_result = prediction[0]
            if(prediction_result == 1):
                AV_result = prediction_result
            ...

        all_results = [AV_result, PV_result, TV_result, MV_result]
        all_original_values = [AV_True_Label, PV_True_Label, TV_True_Label, MV_True_Label]
        all_original_values = list(filter(lambda a: a != None, all_original_values))

        if(AV_True_Label != None and AV_True_Label == AV_result):
            AV_corrects = AV_corrects + 1
        ...

        if(MV_True_Label != None and MV_True_Label == MV_result):
            MV_corrects = MV_corrects + 1

        if 1 in all_results:
            results.append([0, all_original_values[0]])
        else:
            results.append([2, all_original_values[0]])
    return results
```

Figura 46 - Classificação do paciente recorrendo a 4 modelos

Nesta função é utilizado cada um dos modelos criados para realizar a classificação de cada um dos focos existentes. No final da classificação, caso exista pelo menos 1 dos valores indicados com classificação “presente” o paciente é classificado como sendo anormal. Caso todos os focos apresentem uma classificação de “absent” o paciente é classificado como normal.

### 6.3.2.1 Abordagem 1 - SVM

Uma das abordagens utilizada no trabalho que aqui se descreve, foi recorrer ao algoritmo SVM para a criação de um modelo de classificação capaz de classificar os batimentos cardíacos como sendo normais ou anormais. Tal como abordado anteriormente, foram criadas funções para ajudar na preparação, conceção e treino dos dados e modelos a treinar.

No entanto, os atributos recolhidos durante a fase descrita na secção 6.2 apresentam 3 dimensões, sendo necessário no caso de uma SVM um valor de entrada de apenas 1 dimensão. Sendo assim, na Figura 47 é possível contemplar a conversão dos vários vetores de 3 dimensões em vetores de apenas uma dimensão, que posteriormente é fornecido ao modelo para realização do treino.

```
def train_model(model, x, y):
    trainX = tf.reshape(x, (x.shape[0], x.shape[1] * x.shape[2]))
    trainY = list(map(get_max_value, y))
    return model.fit(trainX, trainY)
```

Figura 47 - Função de treino de abordagem SVM

Além disto, é importante referir que também foram realizadas experiências hiperparâmetros da SVM de modo a otimizar o desempenho do modelo de classificação, tal como se pode verificar na Figura 48. As experiências foram realizadas recorrendo ao “GridSearchCV” que iterou sobre uma quantidade de parâmetros diferentes de modo a validar qual a melhor configuração para obter os melhores resultados no treino do modelo. Através desta busca constatou-se que os parâmetros que obtiveram melhor desempenho foram o “C” com 1 e o *gamma* com o valor “scale”, sendo estes utilizados para as futuras experiências.

```
def tuning(x, y, weights):
    trainX = tf.reshape(x, (x.shape[0], x.shape[1] * x.shape[2]))
    trainY = list(map(get_max_value, y))
    param_grid = {'C': [0.1, 1, 10, 1000],
                  'gamma': [1, 0.01, 0.0001],
                  'kernel': ['rbf']}
    grid = GridSearchCV(estimator=svm.SVC(class_weight=weights),
                        param_grid=param_grid, refit = True, verbose = 3,
                        scoring="recall_weighted")
    # fitting the model for grid search
    return grid.fit(np.array(trainX), np.array(trainY))
```

Figura 48 - Exemplo de utilização de GridSearchCV

Para a realização do resto das experiências foram alterados os parâmetros *kernal* utilizado para a realização da classificação e o grau da função polinomial, caso o *kernal* seja definido como polinomial.

### 6.3.2.2 Abordagem 2 – Random Forest

Na segunda abordagem utilizou-se o algoritmo Random Forest, sendo o seu método de preparação semelhante ao enunciado anteriormente.

É importante notar que neste caso foi utilizado o “RandomizedSearchCV” para selecionar os melhores parâmetros (tuning) devido à grande quantidade de parâmetros existentes para otimizar o modelo criado. Na Figura 49, está presente a função criada para realização desta seleção e onde foram utilizados 7 parâmetros com valores escolhidos aleatoriamente pela ferramenta. Através deste método, foi possível verificar qual a melhor configuração para o treino de modelos para este problema, sendo as configurações utilizadas as definidas na Tabela 16.

Tabela 16 - Parâmetros utilizados no algoritmo Random Forest

Parâmetro	Valores
max_features	sqr
max_depth	None
min_samples_split	2
min_samples_leaf	1

Tal como no caso anterior, para além dos parâmetros previamente mencionados, o parâmetro com maior impacto foi o número de árvores de decisão utilizado, sendo este analisado na secção 7.4.2.

```
def tuning(x, y, weights):
    trainX = tf.reshape(x, (x.shape[0], x.shape[1] * x.shape[2]))
    trainY = list(map(get_max_value, y))
    # Number of trees in random forest
    n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
    # Number of features to consider at every split
    max_features = ['auto', 'sqrt']
    # Maximum number of levels in tree
    max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
    max_depth.append(None)
    # Minimum number of samples required to split a node
    min_samples_split = [2, 5, 10]
    # Minimum number of samples required at each leaf node
    min_samples_leaf = [1, 2, 4]
    # Method of selecting samples for training each tree
    bootstrap = [True, False]
    # Create the random grid
    random_grid = {'n_estimators': n_estimators,
                  'max_features': max_features,
                  'max_depth': max_depth,
                  'min_samples_split': min_samples_split,
                  'min_samples_leaf': min_samples_leaf,
                  'bootstrap': bootstrap}

    rf_random = RandomizedSearchCV(estimator = RandomForestClassifier(class_weight=weights),
                                  param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)
    # fitting the model for grid search
    return rf_random.fit(np.array(trainX), np.array(trainY))
```

Figura 49 - Exemplo de utilização de RandomSearchCV

### 6.3.2.3 Abordagem 3 – Rede Neuronal Convolucional

Dentro da área das redes neuronais, foi utilizada a CNN com o intuito de reduzir a complexidade dos dados recebidos e conseguir detetar padrões nos sons provenientes dos batimentos cardíacos. É também uma abordagem bastante mais configurável, sendo possível não só realizar diferentes abordagens de convolução, deteção de características e aplicar mecanismos de prevenção de *overfitting*.

No que refere à arquitetura da CNN, esta recebe como parâmetro de entrada uma imagem com uma determinada dimensão (dependendo da extração de atributos) e retorna uma probabilidade da imagem pertencer à classe positiva a analisar.

Inicialmente são utilizadas 3 camadas de convolução que permitem não só reduzir o tamanho da imagem como também permite detetar as principais características existentes na imagem, descartando informação que o modelo não considera relevante para a correta classificação dos sons em análise. Em cada uma das camadas é aplicada uma convolução, um “max pooling” e uma normalização, condensando a imagem antes de passar a nova informação para a camada seguinte.

Por fim, os dados são convertidos em um vetor de 1 dimensão, começando assim a aplicação das camadas totalmente conectadas. Nestas camadas a rede tenta capturar as características geradas pelas camadas anteriores e criar relações que permitam a correta classificação de cada um dos sons em análise.

Na Figura 50, é possível verificar um exemplo desta arquitetura quando o atributo utilizado para treinar a rede neuronal é um MFCC. Ao percorrer cada camada convolucional, o tamanho da imagem é condensado para metade do seu tamanho original, sendo no fim apenas extraído um valor em 0 e 1.

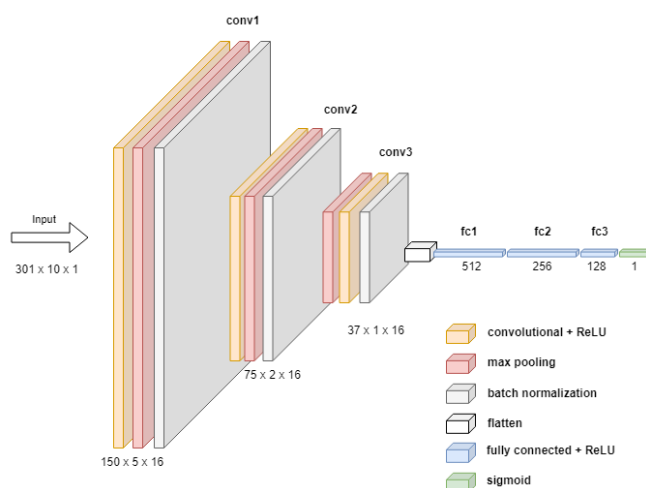


Figura 50 - Modelo RCN para cada foco

É ainda importante referir que de modo a combater problemas de *overfitting*, foram adicionadas camadas de *Dropout* no fim das várias camadas convolução e totalmente

conectadas. Isto permite ao modelo ignorar alguns dos nós presentes no modelo durante a fase de treino permitindo ao modelo realizar uma melhor generalização. Na Figura 51 está ainda presente a função de ativação “sigmoid” utilizada para a classificação final da imagem, bem como a função de perda “Binary Crossentropy”. Estas configurações foram utilizadas uma vez que o problema apenas apresenta 2 classes de classificação, sendo esta a presença de um sopro ou a sua ausência (“presente” ou “absent”).

```
def build_model():
    model = Sequential()

    model.add(Conv2D(filters=16, kernel_size=3, strides=1, padding="same", activation="relu", input_shape=(301, 10, 1)))
    model.add(MaxPooling2D())
    model.add(BatchNormalization())
    model.add(Dropout(0.25))

    model.add(Conv2D(filters=32, kernel_size=3, strides=1, padding="same", activation="relu"))
    ...

    model.add(Flatten())

    model.add(Dense(512, activation="relu"))
    model.add(Dropout(0.5))
    model.add(Dense(256, activation="relu"))
    model.add(Dropout(0.5))
    model.add(Dense(128, activation="relu"))
    model.add(Dense(1, activation="sigmoid"))

    metrics = ['accuracy', tf.keras.metrics.Precision(class_id=0), tf.keras.metrics.Recall(class_id=0)]

    model.compile('adam', loss=tf.losses.BinaryCrossentropy(), metrics=metrics)

    return model
```

Figura 51 - Construção de CNN

#### 6.3.2.4 Abordagem 4 - Modelo Pré-Treinado

A última abordagem utilizada pressupõe a utilização de modelos pré-treinados para a classificação dos vários pacientes. Deste modo, é possível recorrer à arquitetura e pesos do modelo treinado a partir do *dataset* ImageNet (ImageNet, 2021). Este *dataset* apresenta uma elevada quantidade de imagens permitindo assim aos algoritmos que a utilizam criar modelos com uma elevada taxa de acerto sobre classes presentes.

No caso do presente problema, foi pensado inicialmente a utilização dos modelos ConvNet (Liu, et al., 2022) fornecidos pelo “Keras”. No entanto, devido à grande complexidade da arquitetura do modelo e à reduzida quantidade de dados existentes, não foi possível construir modelos capazes de realizar classificações eficazes.

Sendo assim, foi utilizado o modelo pré-treinado VGG16 (Simonyan & Zisserman, 2015), também fornecido pelo “Keras” e que já apresenta um conjunto de pesos do *dataset* ImageNet para começar o treino de um novo modelo. No entanto, uma vez que o conjunto de dados da ImageNet é bastante diferente do conjunto de dados do problema a resolver, foi apenas utilizada a arquitetura do modelo pré-treinado VGG16 para a construção do modelo. Para permitir uma correta aprendizagem, foram ainda removidas o conjunto de camadas totalmente conectadas do modelo VGG16, sendo posteriormente adicionadas camadas totalmente conectadas personalizadas de modo a encaixar corretamente com o problema em questão.

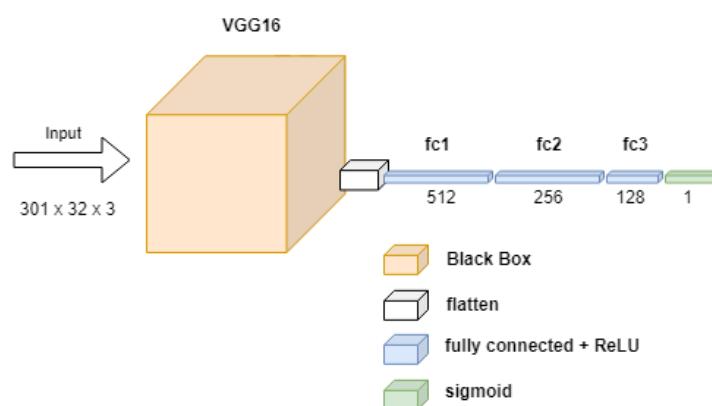


Figura 52 - Modelo Pré-Treinado para cada foco

Tal como na abordagem anterior, o valor de entrada do modelo é um MFCC, no entanto, tal como apresentado na Figura 52, este apresenta uma dimensão diferente. Tal deve-se ao facto de o modelo pré-treinado necessitar de um dimensão mínima de 32 x 32, obrigando assim a aumentar o número de coeficientes utilizados no MFCC de 10 para 32.

### 6.3.3 Alternativa 2 – Proposta Legado

No que toca a alternativa 2, apenas foram utilizadas abordagens que utilizam redes neuronais uma vez que este tipo de modelos permite a concatenação de diferentes valores de entrada para a classificação de imagens. Apesar desta restrição, a preparação dos modelos continua a seguir a mesma abordagem que foi definida durante a secção 6.3.2, diferenciando apenas a sua implementação.

É ainda importante referir que uma vez que é fornecido ao modelo 4 sons correspondente a cada ponto de auscultação de um paciente, foi necessário escolher apenas 1 dos segmentos criados para cada ficheiro de áudio. Para o presente problema, foi apenas escolhido o primeiro segmento de 3 segundos que começa numa fase S1 do batimento cardíaco.

#### 6.3.3.1 Abordagem 1 – Rede Neuronal Convolutacional

Tal como na alternativa anterior, foi utilizada a abordagem de rede neuronal convolutacional. No entanto, ao invés de ser criado 1 modelo diferente por cada foco de auscultação, foi criado apenas 1 modelo que apresenta 4 caminhos paralelos para classificação de cada um dos focos. Por fim, o resultado de cada um destes caminhos é compilado num só vetor, sendo depois aplicadas camadas totalmente conectadas aos valores de modo a conseguir extrair relações relevantes entre os vários modelos.

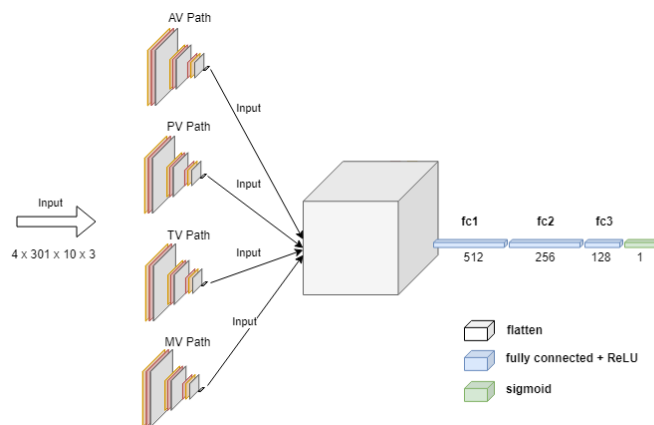


Figura 53 - Modelo CNN com 4 inputs

Na Figura 53 é possível verificar a arquitetura final desta abordagem recorrendo a uma CNN, sendo que ao contrário do referido na alternativa 1, não existem camadas totalmente conectadas em cada um dos modelos intermédios. Isto leva a que apenas sejam realizadas convoluções por cada um dos ramos da CNN, sendo posteriormente concatenados todos os vetores de cada ramo, sendo por fim aplicadas camadas totalmente conectadas, culminando numa só classificação.

Por fim, a função de implementação é ligeiramente diferente, sendo necessária a criação de 4 modelos e a sua concatenação. O resultado da concatenação é o ponto a partir do qual começa a aplicação das camadas totalmente conectadas, tal como presente na Figura 54.

```
def create_complex_model():
    modelIAV = build_model()
    modelIPV = build_model()
    modelITV = build_model()
    modelIMV = build_model()

    concatenated = tf.keras.layers.Concatenate()([modelIAV.output, modelIPV.output, modelITV.output, modelIMV.output])

    fully_connected = Dense(512, activation="relu")(concatenated)
    fully_connected = Dropout(0.5)(fully_connected)
    fully_connected = Dense(256, activation="relu")(fully_connected)
    fully_connected = Dropout(0.5)(fully_connected)
    fully_connected = Dense(128, activation="relu")(fully_connected)
    fully_connected = Dropout(0.5)(fully_connected)
    fully_connected = Dense(1, activation="sigmoid")(fully_connected)

    combined_model = tf.keras.Model(inputs=[modelIAV.input, modelIPV.input, modelITV.input, modelIMV.input], outputs=fully_connected)

    metrics = ['accuracy', tf.keras.metrics.Precision(class_id=0), tf.keras.metrics.Recall(class_id=0)]
    combined_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=metrics)
```

Figura 54 - Construção de CNN com 4 focos

### 6.3.3.2 Abordagem 2 – Modelo Pré-Treinado

A última abordagem é bastante similar à abordagem da proposta da CNN, sendo diferente na arquitetura utilizada em cada um dos ramos da CNN. Na Figura 55 está presente a arquitetura elaborada, onde foi utilizado o modelo VGG16 para cada um dos ramos, sendo cada um responsável por conseguir retirar características referentes a cada ponto de auscultação recebido.

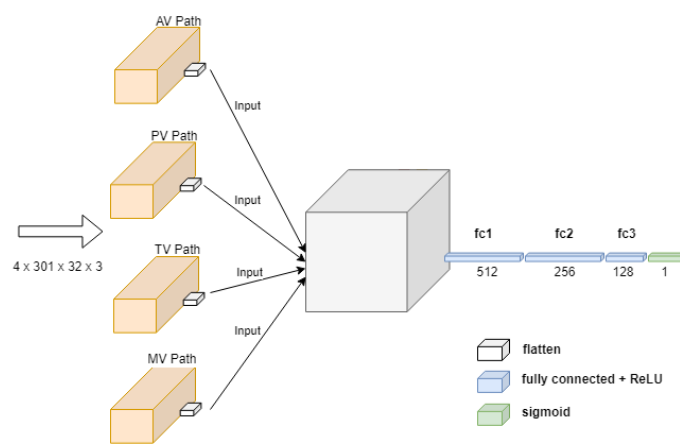


Figura 55 - Modelo CNN com 4 inputs com auxílio de modelo pré-treinado

A vantagem desta abordagem face à anterior reside na utilização de modelos com uma arquitetura já estudada que apresenta resultados favoráveis em vários artigos existentes (Demir, et al., 2019), conferindo uma maior fiabilidade à sua capacidade de generalizar corretamente para cada um dos pacientes.

## 7 Avaliação

Nesta secção é apresentada a hipótese que se pretende verificar com o desenvolvimento do projeto. Para além disso, é estudado o *dataset* a ser utilizado, bem como as medidas e métodos de avaliação que são utilizados para a classificação da solução a desenvolver.

### 7.1 Hipótese

A hipótese que se pretende alcançar com esta investigação deve ser definida de acordo com os objetivos delineados anteriormente a fim conseguir resolver o problema em questão.

O objetivo do presente documento é descrever o trabalho que consistiu em conceber uma solução automática que permita auxiliar os profissionais de saúde na análise dos batimentos cardíacos de um paciente, despistando a existência de uma doença cardiovascular.

Tal deverá ser feito através da construção de um algoritmo que analisando os sons de pontos provenientes de pontos de auscultação obtendo como resultado a classificação do som como sendo normal ou anormal.

Tendo isto em conta, a pergunta que se pretende responder através o estudo a desenvolver é “Através da utilização do conjunto de dados da George B. Moody PhysioNet Challenge (Reyna, et al., 2022) referentes a sons obtidos via pontos de auscultação, será possível realizar um despiste de doenças cardiovasculares de modo a auxiliar os técnicos de saúde?”.

Sendo assim, a hipótese que se pretende explorar com este projeto é se será possível criar um modelo de aprendizagem profunda capaz de auxiliar um profissional de saúde a realizar um despiste inicial de problema cardíaco, para um determinado paciente.

## 7.2 Medidas de Avaliação

Após desenhada uma solução para a resolução de um determinado problema é necessário avaliar a qualidade de resposta perante novos dados. No caso da área da aprendizagem profunda é fundamental perceber não só o nível de acerto que o modelo desenvolvido está a produzir, bem como este se posiciona face a outros modelos já existentes.

No problema que aqui se trata, é necessário compreender qual o nível de acerto que o modelo consegue na classificação dos sons cardíacos recebidos dos pacientes. Contudo, mesmo que o nível de exatidão possa ser bastante elevado, pode ter interesse a utilização de outras medidas de avaliação.

### 7.2.1 Matriz de Confusão

A matriz de confusão é uma matriz, em que através da observação dos seus valores é possível compreender a confusão do modelo criado ao classificar a informação recebida (Kundu, 2023). Esta é caracterizada por uma matriz N por N onde N é o número de classes existentes no modelo para atribuir a informação recebida. No seu caso mais simples, e tal como apresentado na Tabela 17, a matriz de confusão pode ser identificada apenas por 2 classes (caso binário) sendo as linhas representativas dos valores verdadeiros, e as colunas a classificação atribuída pelo modelo. No presente problema, um caso positivo é considerado um paciente com um som anormal classificado corretamente.

Tabela 17 - Matriz de confusão

		Valor Previsto	
		Positivo	Negativo
Valor Real	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

De modo a simplificar a compreensão do conceito foi desenhada a tabela supramencionada que apenas apresenta 2 tipos de classificação, sendo estas os valores “Positivo” e “Negativo”. Para além disso, a tabela apresenta ainda os seguintes termos (Kundu, 2023):

- **Verdadeiro Positivo:** Corresponde a um valor que apresenta o valor real positivo e o valor previsto pelo modelo classificado adequadamente.
- **Verdadeiro Negativo:** Corresponde a um valor que apresenta o valor negativo e o valor de previsão do modelo refere também como negativo.
- **Falso Positivo (Erro Tipo 1):** Corresponde a um valor que apresenta o valor real de negativo, mas foi previsto pelo modelo como sendo um valor positivo.
- **Falso Negativo (Erro Tipo 2):** Corresponde a um valor que apresenta o valor real de positivo, mas foi previsto pelo modelo como sendo um valor negativo

## 7.2.2 Medidas de desempenho

Apesar da matriz de confusão não se apresentar como uma métrica de desempenho, através da sua análise é possível extrair os dados necessários para o cálculo de métricas importantes para a avaliação e classificação do desempenho de um modelo. Deste modo, para o efeito da secção atual, é utilizada a matriz de confusão desenhada na secção anterior para a descrição dos vários medidores de avaliação de desempenho.

### 7.2.2.1 Exatidão - taxa de acerto

A exatidão é uma métrica que permite identificar a frequência com a qual o modelo criado acerta na previsão da informação obtida. No caso de existirem 100 dados de entrada, quando a exatidão assume um valor de 90% podemos afirmar que em 90 dos dados analisados foram classificados corretamente. A equação que representa a exatidão é:

$$\text{Exatidão} = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões efetuadas}}$$

Apesar de não ser necessário a utilização da matriz de confusão para deduzir esta métrica, através da sua análise também é possível realizar o cálculo da exatidão, obtendo então a seguinte fórmula:

$$\text{Exatidão} = \frac{VP + VN}{VP + VN + FP + FN}$$

Apesar desta métrica ser bastante importante para calcular a eficácia do modelo, sozinha pode incorrer no erro de apenas classificar corretamente parte das classificações estipuladas. Segundo o caso anterior, se a 90 dos 100 corresponderem a valores reais negativos e a classificação do modelo também for negativa, mesmo sem ter em conta os casos positivos é possível obter uma exatidão de 90%. O que leva à possibilidade de o modelo não conseguir classificar corretamente informação que seja “positiva” induzindo em erro face à sua eficácia (Yıldırım, 2020).

### 7.2.2.2 Precisão

Quando o modelo que se está a tentar avaliar apresenta apenas 2 classificações possíveis, a precisão é uma boa métrica para se ter em conta. Esta foca-se nas predições positivas conseguindo assim perceber a percentagem de informação classificada como positiva que de facto deveria ser classificada como positiva. A seguinte fórmula descreve os cálculos necessários para a obtenção da precisão (Yıldırım, 2020):

$$\text{Precisão} = \frac{VP}{VP + FP}$$

### 7.2.2.3 Recall (Sensibilidade)

A medida recall, tal como a precisão, é uma medida bastante útil quando se tenta classificar modelos que apenas apresentam 2 possibilidades de classificação focando-se apenas na classificação positiva. Através desta métrica é possível perceber a capacidade do modelo criado de classificar corretamente dados que apresentem um valor real positivo. A fórmula que representa o recall é a seguinte (Yıldırım, 2020):

$$Recall = \frac{VP}{VP + FN}$$

#### 7.2.2.4 Especificidade (taxa de acerto na classe negativa)

A especificação é uma métrica semelhante à sensibilidade. No entanto, ao contrário da sensibilidade, esta não tem como seu propósito o foco na classificação positiva, mas sim na classificação negativa. Deste modo é possível compreender a capacidade do modelo criado de classificar corretamente dados que apresentem um valor real negativo, sendo dado pela seguinte fórmula (Yıldırım, 2020):

$$Especificidade = \frac{FP}{VN + FP}$$

#### 7.2.2.5 F1-Score

O F1-score é a média dos pesos referente à precisão e ao recall. Esta medida é especialmente mais importante que a exatidão nos casos em que o modelo a ser aplicado apresenta múltiplas distribuições de classificações, uma vez que tem em conta tanto os falsos positivos como os falsos negativos. A fórmula que representa esta métrica é a seguinte (Yıldırım, 2020):

$$F1_{score} = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

#### 7.2.2.6 Overall Score

O overall score é uma métrica que mede a média da entre a taxa de acerto na classe positiva e a taxa de acerto da classe negativa de um determinado modelo. O seu cálculo é então dado pela seguinte fórmula (Clifford, et al., 2016):

$$Overall\ Score = \frac{Sensibilidade + Epecificidade}{2}$$

## 7.3 Avaliação do Sistema

A avaliação do sistema desenvolvido é muito importante para compreender de que modo é que o modelo desenvolvido consegue responder ao problema que se pretende resolver. Para tal, devem ser utilizadas as metodologias de avaliação definidas anteriormente, com as métricas adequadas para avaliar os resultados obtidos.

Para a avaliação das várias experiências utilizadas, foi utilizado o método de avaliação *Holdout*, garantindo que o paciente presente no conjunto de treino não se encontra presente no conjunto de teste. Com os valores obtidos deverá ser construída uma matriz de confusão e posteriormente recolhidas as métricas referentes à exatidão, sensibilidade, especificidade, f1-score e overall accuracy.

Estas métricas devem ser utilizadas não só para realizar a comparação entre diversas abordagens como também devem permitir comparar os resultados obtidos por outras soluções já existentes.

## 7.4 Experiências Realizadas

Nesta secção são abordadas as experiências realizadas de acordo com as várias implementações descritas no capítulo 6. Sendo assim, esta secção foi dividida em experiências realizadas para a alternativa 1 e para a alternativa 2, tendo em conta os diferentes atributos recolhidos e as estratégias de segmentação enunciadas.

Para questões de experimentação, apenas foram consideradas duas classes para a classificação, sendo estas as classes de “abnormal” e “normal” como classe positiva e classe negativa, respetivamente.

### 7.4.1 Alternativa 1 – SVM

No que toca à utilização do algoritmo SVM, podemos verificar que os melhores resultados foram obtidos pela utilização de atributos MFCC. Apesar de terem sido realizadas várias experiências com este tipo de algoritmo, foi observado que de um modo geral, os melhores resultados foram obtidos através da utilização de um *kernal* “rbf”, sendo que a Tabela 18 apenas apresenta resultados obtidos através da utilização deste *kernal*.

Para além disto, é importante reparar que para o atributo espectrograma de mel, não foi possível construir um modelo com base na 3ª segmentação devido à insuficiência recursos da máquina utilizada.

No que toca às restantes experiências, é possível verificar que cada um das abordagens apresenta melhores resultados face a diferentes métricas. Apesar das abordagens referentes ao atributo de espectrograma de mel apresentarem melhores valores de recall, demonstram uma grande dificuldade em apresentar um classificação de overall accuracy maior que 50%. Para além disto, apresentam um valor de F1-score bastante baixo, havendo uma baixa relação entre a precisão dos modelos e a sua capacidade de prever corretamente a classe positiva.

No que toca ao modelos que utilizam como atributo o MFCC, estes apresentam não só um melhor overall accuracy como também apresentam uma melhor relação entre precisão e recall dada pela métrica F1-Score.

Para esta experiência, é então possível concluir que a estratégia com melhores resultado recorre ao atributo MFCC com a 2ª estratégia de segmentação.

Tabela 18 – Comparação de diferentes abordagens nos modelos SVM

Atributos	Segmentação	Exatidão	Recall	Especificidade	F1-Score	Overall Accuracy
MFCC	Segmentação 1	59,84%	65,85%	58,62%	35,53%	62,24%
	Segmentação 2	<b>67,62%</b>	58,54%	69,46%	37,80%	<b>64,00%</b>
	Segmentação 3	56,97%	23,77%	<b>70,73%</b>	<b>54,19%</b>	35,58%
Mel Spectrogram	Segmentação 1	35,25%	75,61%	27,09%	28,18%	51,35%
	Segmentação 2	29,51%	<b>78,05%</b>	19,70%	27,12%	48,88%
	Segmentação 3					

#### 7.4.2 Alternativa 1 - Random Forest

Passando ao algoritmo de random forest, é possível verificar através da análise da Tabela 19 que este algoritmo permitiu a construção de modelos com melhor capacidade de classificação face ao algoritmo SVM.

Para a realização desta experiência, foram utilizadas mais uma vez diversas configurações para a construção do modelo, sendo que a que apresentou melhores resultados foi a utilização de 10 árvores para a classificação de cada paciente. Como tal, todos os resultados apresentados utilizaram esta configuração.

De acordo com a tabela, podemos verificar que a utilização de atributos MFCC conferiu aos modelos criados melhor desempenho na classificação dos pacientes. É possível comprová-lo através da consulta das métricas de recall, f1-score e overall accuracy que apresentam valores bastante superiores face à utilização do atributo alternativo.

Por outro lado, a utilização de espectrogramas de mel demonstra melhor desempenho no que toca às medidas de exatidão e de especificidade dos modelos criados. Tal é justificável devido ao grande desequilíbrio da classe de “present” face à classe de “absent”. Apesar de estes modelos apresentarem elevados valores de exatidão, tal deve-se em grande parte à sua capacidade de classificar corretamente amostras da classe “absent”, tal como podemos verificar pela elevada especificidade. Por outro lado, não são indicados para generalizar ambas as classes, tal é possível observar pelo baixo recall e uma overall accuracy muito perto de 50%.

Para esta experiência, a melhor combinação foi a utilização do atributo MFCC, recorrendo à segmentação número 3, atingindo uma overall accuracy de 71.74%.

Tabela 19 - Comparação de diferentes abordagens nos modelos Random Forest

Atributos	Segmentação	Exatidão	Recall	Especificidade	F1-Score	Overall Accuracy
MFCC	Segmentação 1	73,40%	66,67%	75,00%	<b>48,98%</b>	70,83%
	Segmentação 2	70,24%	<b>72,22%</b>	69,74%	48,15%	70,98%
	Segmentação 3	75,31%	66,67%	76,81%	44,44%	<b>71,74%</b>
Mel Spectrogram	Segmentação 1	75,61%	25,00%	84,29%	23,08%	54,64%
	Segmentação 2	<b>80,00%</b>	25,00%	<b>93,75%</b>	33,33%	59,38%
	Segmentação 3	70,00%	37,50%	78,13%	33,33%	57,81%

### 7.4.3 Alternativa 1 - Rede Neuronal Convolutacional

Quanto à utilização de algoritmos de aprendizagem profunda, foi utilizada a CNN explorada na secção 6.3.2.3 para a realização das experiências apresentadas na Tabela 20. Este algoritmo conseguiu atingir resultados melhores do que os resultados apresentados pelo algoritmo de random forest, sendo o primeiro a conseguir atingir uma overall accuracy superior a 75%.

Pela observação dos resultados, é possível observar que, tal como no algoritmo anterior, este algoritmo conseguiu a maior exatidão e especificidade na utilização do atributo espectrograma de mel, sendo que a justificação para este comportamento se mantém o mesmo que foi referido na secção 7.4.2.

Por outro lado, é possível observar uma elevada capacidade do algoritmo para classificar amostras da classe “presente” através da utilização do atributo MFCC. Para além disto, a segmentação número 3 foi a segmentação que apresentou o maior valor de recall dentro das várias experiências efetuadas, conseguindo atingir um valor de 90%.

Contudo, a melhor experiência efetuada para o presente algoritmo, foi a utilização 2ª segmentação, conseguindo atingir não só um valor de overall accuracy superior a 75%, como também conseguiu atingir um f1-score superior a 50%.

Tabela 20 - Comparação de diferentes abordagens nos modelos CNN

Atributos	Segmentação	Exatidão	Recall	Especificidade	F1-Score	Overall Accuracy
MFCC	Segmentação 1	62,60%	68,18%	61,39%	39,47%	64,78%
	Segmentação 2	79,67%	68,18%	82,18%	<b>54,55%</b>	<b>75,18%</b>
	Segmentação 3	29,27%	<b>90,09%</b>	15,84%	31,45%	52,97%
Mel Spectrogram	Segmentação 1	71,54%	68,18%	72,28%	46,15%	70,23%
	Segmentação 2	<b>86,18%</b>	45,45%	<b>95,05%</b>	54,05%	70,25%
	Segmentação 3	52,85%	86,36%	45,54%	39,58%	65,95%

#### 7.4.4 Alternativa 1 – Modelo Pré-Treinado

Por fim, foram realizadas experiências através da utilização de modelos pré-treinados, estando os resultados desta experiência apresentados na Tabela 21. Este algoritmo conseguiu atingir resultados bastante favoráveis, conseguindo alcançar um overall accuracy acima de 75% e um f1-score superior a 60%.

No que toca à utilização deste algoritmo, não foi possível obter bons resultados na criação de modelos recorrendo ao atributo espectrogramas de mel. Tal é possível avaliar pela medidas de desempenho recall e especificidade que são 0% e 100%, respetivamente. Tal indica que o modelo classificou todas as amostras como sendo da classe “absent” não sendo capaz de generalizar corretamente para o problema em questão. Para além disto, é importante reparar que devido à elevada dimensão das imagens extraídas com este atributo, não foi possível a criação de um modelo para a segmentação 3. Tal deve-se a limitação das máquinas utilizadas para a criação dos modelos.

Por outro lado, a utilização do atributo MFCC permitiu a construção de modelos com elevado nível de desempenho. Tal como presente na tabela, a 3ª segmentação conseguiu atingir um valor de recall de 90% sendo o ideal para classificar amostras com classificação “presente”.

Apesar disto, a melhor experiência foi conseguida através da utilização da segunda segmentação, conseguindo atingir melhores valores em todas as outras métricas utilizadas, terminando com um overall accuracy bastante superior às demais experiências.

Tabela 21 - Comparação de diferentes abordagens nos modelos Pré-treinados

Atributos	Segmentação	Exatidão	Recall	Especificidade	F1-Score	Overall Accuracy
MFCC	Segmentação 1	64,39%	79,97%	61,57%	42,72%	70,77%
	Segmentação 2	<b>86,18%</b>	63,64%	<b>91,09%</b>	<b>62,22%</b>	<b>77,36%</b>
	Segmentação	54,47%	<b>90,91%</b>	46,53%	41,67%	68,72%
Mel Spectrogram	Segmentação 1	82,11%	0,00%	100,00%	0,00%	50,00%
	Segmentação 2	82,11%	0,00%	100,00%	0,00%	50,00%
	Segmentação 3					

#### 7.4.5 Alternativa 2

Nesta última abordagem, apenas foi realizada a experiência com base numa só segmentação dos dados. Tal deve-se ao formato das amostras fornecidas ao modelo necessitarem de um segmento por cada foco de auscultação por cada um dos pacientes. Sendo assim, nesta abordagem apenas foi utilizada a segunda segmentação uma vez que nas experiências anteriores esta se apresentou como a segmentação que obteve melhores resultados.

No que toca à utilização dos atributos de espectrograma de mel, tal como apresentado na Tabela 22, para ambos os algoritmos, não foi possível obter bons resultados uma vez que o modelos apenas foi capaz de classificar a classe “normal”.

Por outro lado, através da utilização dos atributos de MFCC foi possível obter resultados bastante favoráveis, conseguindo em ambos os casos obter uma overall accuracy superior a 70% e respetivos f1-scores superiores a 60%.

Tabela 22 - Comparação de diferentes abordagens na alternativa 2

Atributos	Algoritmo	Accuracy	Recall	Especificidade	F1-Score	Overall Accuracy
MFCC	CNN	83,33%	<b>64,71%</b>	88,06%	61,11%	<b>76%</b>
	Pré-treinado	<b>86,90%</b>	52,94%	95,52%	<b>62,07%</b>	74%
Mel Spectrogram	CNN	79,76%	0,00%	<b>100,00%</b>	0,00%	50%
	Pré-treinado	79,76%	0,00%	<b>100,00%</b>	0,00%	50%

## 7.5 Comparação e discussão de Resultados

Por fim, é necessário realizar uma comparação dos modelos criados com os diferentes algoritmos e com a literatura existente. Deste modo é possível perceber qual dos modelos criados é mais indicado para a resolução do problema e de que modo é que estes se apresentam relevantes face às soluções existentes.

Tabela 23 - Melhor modelo de cada experiência realizada

Alternativas	Algoritmo	Accuracy	Recall	Especificidade	F1-Score	Overall Accuracy
Alternativa 1	SVM	67,62%	58,54%	69,46%	37,80%	64,00%
	Random Forest	75,31%	66,67%	76,81%	44,44%	71,74%
	CNN	79,67%	<b>68,18%</b>	82,18%	54,55%	75,18%
	Pré-Treinado	86,18%	63,64%	91,09%	<b>62,22%</b>	<b>77,36%</b>
Alternativa 2	CNN	83,33%	64,71%	88,06%	61,11%	76%
	Pré-treinado	<b>86,90%</b>	52,94%	<b>95,52%</b>	62,07%	74%

Através da observação das medidas de desempenho apresentadas na Tabela 23, é possível confirmar que o melhor método de classificação desenvolvido pertence à utilização de 4 modelos distintos para a classificação de um paciente recorrendo a utilização de modelos pré-treinados. Este modelo apresenta uma overall accuracy de 77.36% e um f1-score de 62.22%.

Apesar deste ser o melhor modelo, a alternativa 2 recorrendo a um modelo CNN fica imediatamente atrás com medidas de desempenho muito próximas e uma overall accuracy de 76%. Esta diferença poderá ser motivada pela escassez de pacientes, limitando em grande quantidade a capacidade do modelo aprender a generalizar corretamente.

Por fim, foi ainda realizada a comparação do modelo criado com as soluções apresentadas na literatura. A Tabela 24 apresenta as métricas de desempenho do melhor modelo criado e trabalhos que utilizam o mesmo *dataset* utilizado no concurso George B. Moody PhysioNet Challenge (Reyna, et al., 2022).

Como é possível observar pelos valores da tabela, o modelo desenvolvido conseguiu apresentar uma overall accuracy e f1-score melhor do que os trabalhos submetidos para o concurso, conseguindo ainda uma overall accuracy melhor do que um dos trabalhos não pertencentes ao concurso. No entanto, este não é o melhor modelo desenvolvido, havendo um modelo desenvolvido que apresenta um overall accuracy de 87.72 (Patwa, et al., 2023).

A única medida de desempenho que o modelo criado obteve os melhores resultados foi na especificidade apresentando um valor de 91.09%. Um dos motivos para este resultado pode ser derivado de uma grande parte dos artigos não apresentar informação relativamente a esta métrica.

Tabela 24 - Comparação da solução com a literatura

Trabalhos	Exatidão	Recall	Especificidade	F1-Score	Overall Accuracy
(Lu, et al., 2022)		66%	30.6%	48.3%	55.45%
(Mcdonald, et al., 2022)		84%	31%		57.5
(Xu, et al., 2022)		<b>90.56</b>			62.96
(Chen, et al., 2022)	<b>94.58%</b>	76.68%		<b>92.2%</b>	
(Patwa, et al., 2023)	87.05%	88.02%			<b>87.72</b>
(Vimalajeewa, et al., 2023)	76.61%	54.03%			68.07%
(Nogueira, et al., 2019)		87.37%	79.07%		83.22%
Proposta Elaborada	86,18%	63,64%	<b>91,09%</b>	62,22%	77,36%

É importante ainda referir que a comparação efetuada não é integralmente justa, uma vez que as metodologias utilizadas por diferentes trabalhos levam a que possam existir diferenças significativas nos resultados obtidos. Estas diferenças ocorrem não só na exclusão de algumas das amostras que apresentam classificações “unknown” como também variam na quantidade de dados sobre os quais os modelos foram trabalhados, uma vez que parte dos dados do *dataset* não se encontravam disponíveis.

## 8 Conclusão

As doenças cardiovasculares são um dos problemas de saúde mais predominantes da atualidade, sendo estes em 2019 responsáveis por 33% dos óbitos registados. Tem-se também que a ausência de sintomas evidentes tornam o despiste destas doenças muito mais complexo do que o normal. É pois importante investir na deteção precoce destas doenças. Existem diversos estudos efetuados que pretendem de forma eficiente detetar estas doenças antes destas se tornem um problema.

Esta dissertação tem como objetivo construir um modelo de aprendizagem profunda que seja capaz de classificar o som cardíaco de um paciente como normal ou anormal. Para tal, foi utilizado o conjunto de dados do concurso George B. Moody PhysioNet Challenge (Reyna, et al., 2022) para treinar os modelos desenvolvidos. Foi necessário realizar pré-processamento e extração de atributos dados e aplicação dos mesmos a diferentes algoritmos de classificação de modo a criar modelos de classificação.

Foram desenvolvidas 2 alternativas que se distinguem pela abordagem seguida na construção dos modelos a utilizar. Para a primeira alternativa, foi criado um modelo de classificação para cada foco de auscultação do coração, sendo a classificação final de cada paciente dada pela junção dos resultados obtidos na classificação por cada modelo. Foram assim utilizadas diversas técnicas de aprendizagem supervisionada como SVM, Random Forest e algoritmos de aprendizagem profunda. A segunda alternativa consiste na construção de apenas um modelo que recebe um segmento de áudio de cada foco de auscultação e apresenta uma classificação final do paciente. Para esta alternativa apenas foram utilizadas abordagens de aprendizagem profunda.

Através das experiências realizadas, foi possível observar que a alternativa 1 foi a melhor para a resolução do problema, apresentando uma overall accuracy de 77.36% e um F1-score de 62.22. Este resultados foram conseguidos através da utilização da 2ª segmentação (explorada na secção 6.1.2), da extração de atributos MFCC e recorrendo à arquitetura do modelo pré-treinado VGG-16.

Com estes resultados, pode observar-se que a utilização deste modelo pode ser uma mais-valia no auxílio de profissionais de saúde no despiste inicial de problemas cardiovasculares, bem como o presente trabalho poderá auxiliar em futuros desenvolvimentos na área de deteção de patologias cardíacas. Todavia, continua a ser necessário a realização de mais experiências e ajustes aos modelos criados, explorar novos algoritmos, aumentar a quantidade e qualidade de dados disponíveis, para se continuar a melhorar a performance dos modelos desenvolvidos.

## 8.1 Trabalho Futuro

No que toca à possibilidade de trabalho futuro, será importante ter em conta:

- Diferentes configurações dos diferentes algoritmos, como também otimizar a construção dos modelos de aprendizagem profunda a fim de tornar os modelos mais robustos e obter melhores resultados.
- Explorar modelos pré-treinados mais recentes e com maior capacidade a fim de explorar o seu desempenho
- Melhorar o hardware utilizado de modo a conseguir realizar experiências que exijam um maior poder computacional.
- Experimentar a utilização de diferentes atributos do som cardíaco existentes no *dataset*, como por exemplo o tom, o formato e a qualidade.
- Utilizar o *dataset* completo do concurso a fim de melhorar o treino dos modelos desenvolvidos.
- Utilizar uma abordagem diferente para a escolha do melhor segmento de som extraído de um determinado ficheiro de áudio.

# Referências

UC Berkeley - School of Information, 2020. *What Is Machine Learning (ML)?*. [Online]  
Available at: <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>  
[Acedido em 19 Fevereiro 2023].

Alkhodari, M., Kamarul Azman, S., Hadjileontiadis, L. J. & Khandoker, A. H., 2022. Ensemble Transformer-Based Neural Networks Detect Heart Murmur In Phonocardiogram Recordings.

American Heart Association, 2019. *Investing in Heart Disease and Stroke Research*. [Online]  
Available at: <https://www.heart.org/-/media/Files/About-Us/Policy-Research/Fact-Sheets/Public-Health-Advocacy-and-Research/Investing-in-Our-Hearts--NIH-Fact-Sheet.pdf>  
[Acedido em 25 Fevereiro 2023].

AudacityTeam, s.d. *Audacity*. [Online]  
Available at: <https://www.audacityteam.org/>  
[Acedido em 23 Março 2023].

Ayodele, T. O., 2010. *Types of Machine Learning Algorithms*. s.l.:s.n.

Bezoui, M., Elmoutaouakkil, A. & Beni-Hssane, A., 2017. Feature extraction of some Quranic recitation using Mel-Frequency Cepstral Coefficients (MFCC). *International Conference on Multimedia Computing and Systems -Proceedings*, Volume 0, pp. 127-131.

Bindra, R., 2023. *Why Is It Called "Logistic Regression" and Not "Logistic Classification"?*. [Online]  
Available at: <https://turbofuture.com/industrial/Why-Logistic-Regression-Why-not-Logistic-Classification>  
[Acedido em 22 Fevereiro 2023].

Bi, Q., Goodman, K. E., Kaminsky, J. & Lessler, J., 2019. What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*, 188(12), pp. 2222-2239.

Bloom, D. et al., 2011. The Global Economic Burden of Non-communicable Diseases. *World Economic Forum*.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), pp. 5-32.

Brownlee, J., 2016. *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*. 1.1 ed. s.l.:s.n.

Brownlee, J., 2018. *A Gentle Introduction to k-fold Cross-Validation*. [Online]  
Available at: <https://machinelearningmastery.com/k-fold-cross-validation/>  
[Acedido em 21 Fevereiro 2023].

Caffe, 2023. *Caffe*. [Online]

Available at: <https://caffe.berkeleyvision.org/>

[Acedido em 14 Fevereiro 2023].

Castrounis, A., 2022. *AI Explained*. [Online]

Available at: <https://www.whyofoai.com/blog/ai-explained>

[Acedido em 19 Fevereiro 2023].

Cheng, K. & Wang, J., 2019. Forest Type Classification Based on Integrated Spectral-Spatial-Temporal Features and Random Forest Algorithm—A Case Study in the Qinling Mountains. 10(7).

Chen, Y. et al., 2022. A Heart Sound Classification Method Based on Residual Block and Attention Mechanism. *Proceedings - 2022 IEEE 21st International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2022*, pp. 1060-1065.

clevelandclinic, t., 2023. *Heart: Anatomy and Function*. [Online]

Available at: <https://my.clevelandclinic.org/health/body/21704-heart>

Clifford, G. D. et al., 2016. Classification of normal/abnormal heart sound recordings: The PhysioNet/Computing in Cardiology Challenge 2016. *Computing in Cardiology*, Volume 43, pp. 609-612.

Demir, F., Şengür, A., Bajaj, V. & Polat, K., 2019. Towards the classification of heart sounds based on convolutional deep neural network. *Health Information Science and Systems*, 7(1), pp. 1-9.

Devi, K., 2021. *Understanding Hold-Out Methods for Training Machine Learning Models*.

[Online]

Available at: <https://heartbeat.comet.ml/understanding-hold-out-methods-for-training-machine-learning-models-733f5179d716>

[Acedido em 23 Fevereiro 2023].

DLT Labs, 2020. *Understanding Machine Learning & Deep Learning*. [Online]

Available at: <https://dltlabs.medium.com/understanding-machine-learning-deep-learning-f5aa95264d61>

[Acedido em 20 Fevereiro 2023].

Dornbush, S. & Turnquest, A. E., 2020. *Physiology, Heart Sounds*. s.l.:StatPearls Publishing.

Dowd, M., 2020. *Negative Effects of a Heavy Workload*. [Online]

Available at: <https://work.chron.com/importance-positive-environment-workplace-3008.html>

[Acedido em 26 Fevereiro 2023].

Educba, 2023. *What is PyTorch? | Applications, Advantages & Disadvantage of PyTorch*.

[Online]

Available at: <https://www.educba.com/what-is-pytorch/>  
[Acedido em 14 Fevereiro 2023].

Farquard, M. A. H. & Bose, I., 2012. Preprocessing unbalanced data using support vector machine. *Decision Support Systems*, 53(1), pp. 226-233.

Feliz, J., 2022. *Detection of cardiac pathology using machine learning*. [Online]  
Available at: <http://hdl.handle.net/10400.22/21718>

Geeks for Geeks, 2023. *Advantages and Disadvantages of TensorFlow*. [Online]  
Available at: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-tensorflow/>  
[Acedido em 14 Fevereiro 2023].

Gupta, S., 2021. *What Is the Best Language for Machine Learning?*. [Online]  
Available at: <https://www.springboard.com/blog/data-science/best-language-for-machine-learning/>  
[Acedido em 13 Fevereiro 2023].

Healio, 2017. *Heart Sounds Topic Review | Learn the Heart*. [Online]  
Available at: <https://www.healio.com/cardiology/learn-the-heart/cardiology-review/topic-reviews/heart-sounds>  
[Acedido em 2 Fevereiro 2023].

Healthline, 2019. *Heart Murmurs and Causes of Other Abnormal Heart Sounds*. [Online]  
[Acedido em 04 Fevereiro 2023].

IBM, 2023. *What are convolutional neural networks?*. [Online]  
Available at: <https://www.ibm.com/topics/convolutional-neural-networks>  
[Acedido em 20 Fevereiro 2023].

IBM, 2023. *What are Neural Networks?*. [Online]  
Available at: <https://www.ibm.com/topics/neural-networks>  
[Acedido em 19 Fevereiro 2023].

IBM, 2023. *What are recurrent neural networks?*. [Online]  
Available at: <https://www.ibm.com/topics/recurrent-neural-networks>  
[Acedido em 21 Fevereiro 2023].

IBM, 2023. *What is deep learning?*. [Online]  
Available at: <https://www.ibm.com/topics/deep-learning>  
[Acedido em 20 Fevereiro 2023].

IBM, 2023. *What is Machine Learning?*. [Online]  
Available at: <https://www.ibm.com/topics/machine-learning>  
[Acedido em 19 Fevereiro 2023].

- ImageNet, 2021. *ImageNet*. [Online]  
Available at: <https://www.image-net.org/>  
[Acedido em 29 Setembro 2023].
- İsbilen, E., 2022. *What Is Caffe?*. [Online]  
Available at: <https://builtin.com/machine-learning/caffe>  
[Acedido em 14 Setembro 2023].
- Japkowicz, N., 2023. *Performance Evaluation for Learning Algorithms*. [Online]  
Available at: [https://www.icmla-conference.org/icmla11/PE\\_Tutorial.pdf](https://www.icmla-conference.org/icmla11/PE_Tutorial.pdf)  
[Acedido em 21 Fevereiro 2023].
- Johns, A., 2021. Journey towards a synthetic consciousness. *Learning Outcomes of Classroom Research*, pp. 56-64.
- Jo, J.-M., 2019. Effectiveness of Normalization Pre-Processing of Big Data to the Machine Learning Performance. *The Journal of the Korea institute of electronic communication sciences*, 14(3), pp. 547-552.
- Karnath, B. & Thornton, W., 2002. Auscultation of the Heart. pp. 39-43.
- Kazemnejad, A., Gordany, P. & Sameni, R., 2021. An Open-Access Simultaneous Electrocardiogram and Phonocardiogram Database. *bioRxiv*.
- Keras, 2023. *About Keras*. [Online]  
Available at: <https://keras.io/about/>  
[Acedido em 14 Fevereiro 2023].
- Koen, P. A. et al., 2002. Fuzzy Front End : Effective Methods, Tool, and Techniques. pp. 5-35.
- Korstanje, J., 2021. *Machine Learning on Sound and Audio data*. [Online]  
Available at: <https://towardsdatascience.com/machine-learning-on-sound-and-audio-data-3ae03bcf5095>  
[Acedido em 21 Março 2023].
- Kundu, R., 2023. *Confusion Matrix: How To Use It & Interpret Results*. [Online]  
Available at: <https://www.v7labs.com/blog/confusion-matrix-guide>  
[Acedido em 21 Fevereiro 2023].
- Lee, J. et al., 2022. Deep Learning Based Heart Murmur Detection Using Frequency-time Domain Features of Heartbeat Sounds.
- Liu, Z. et al., 2022. A ConvNet for the 2020s. *CoRR*.
- Lu, H. et al., 2022. A Lightweight Robust Approach for Automatic Heart Murmurs and Clinical Outcomes Classification from Phonocardiogram Recordings.

- Mahanta, S. K. & Arvin, 2021. *Audio Feature Extraction*. [Online]  
Available at: <https://devopedia.org/audio-feature-extraction>  
[Acedido em 31 Março 2023].
- Mcdonald, A., Gales, M. & Agarwal, A., 2022. Detection of Heart Murmurs in Phonocardiograms with Parallel Hidden Semi-Markov Models.
- MEDPAGE, 2014. *CardioBuzz: More Money Needed for Heart Research?*. [Online]  
Available at: <https://www.medpagetoday.com/opinion/cardiobuzz/44015>
- Minneapolis Heart Institute Foundation, 2023. *Benefits Research*. [Online]  
Available at: <https://mplsheart.org/benefits-research>  
[Acedido em 25 Fevereiro 2023].
- M, S., 2023. *Differences Between AI vs. Machine Learning vs. Deep Learning | Simplilearn*. [Online]  
Available at: <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-vs-machine-learning-vs-deep-learning>  
[Acedido em 18 Fevereiro 2023].
- Muda, L., Begam, M. & Elamvazuthi, I., 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *Journal of Computing*, 2(3), pp. 138-143.
- Muqing, D. et al., 2020. Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. *Neural Networks*, Volume 130, pp. 22-32.
- Nair, P., 2018. *The dummy's guide to MFCC*. [Online]  
Available at: <https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd>  
[Acedido em 17 Setembro 2023].
- Narkhede, S., 2018. *Understanding Confusion Matrix*. [Online]  
Available at: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>  
[Acedido em 21 Fevereiro 2023].
- NetApp, 2023. *What is Deep Learning?*. [Online]  
Available at: <https://www.netapp.com/artificial-intelligence/what-is-deep-learning/>  
[Acedido em 25 Fevereiro 2023].
- Nicola, S., 2023. *Análise de Valor: FAST and QFD Techniques*, s.l.: s.n.
- Nielsen, M., 2021. Neural Networks and Deep Learning. *The Machine Age of Customer Insight*, pp. 91-101.
- Nogueira, D. M., Ferreira, C. A., Gomes, E. F. & Jorge, A. M., 2019. Classifying Heart Sounds Using Images of Motifs, MFCC and Temporal Features. *Journal of Medical Systems*, XLIII(6).

Noor, A. M., 2019. The heart auscultation: From sound to graphical.

NTI Audio, 2023. *Fast Fourier Transformation FFT - Basics*. [Online]

Available at: <https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft>

[Acedido em 1 Abril 2023].

Oliveira, J. et al., 2022. The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification. *IEEE Journal of Biomedical and Health Informatics*, 26(6), pp. 2524-2535.

Patwa, A., Rahman, M. & Al-Naffouri, T., 2023. Heart Murmur and Abnormal PCG Detection via Wavelet Scattering Transform & a 1D-CNN. pp. 1-9.

Pokhrel, S., 2019. *Beginners Guide to Convolutional Neural Networks*. [Online]

Available at: <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d>

[Acedido em 20 Fevereiro 2023].

Python Software Foundation, 2023. *What is Python? Executive Summary*. [Online]

Available at: <https://www.python.org/doc/essays/blurb/>

[Acedido em 12 02 2023].

Ramachandran, K. K. et al., 2022. Machine learning and role of artificial intelligence in optimizing work performance and employee behavior. *Materials Today: Proceedings*, Volume 51, pp. 2327-2331.

Reyna, M. A. et al., 2022. *Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022*. [Online]

Available at: <https://moody-challenge.physionet.org/2022/>

[Acedido em 10 Fevereiro 2023].

Ribeira, R., 2022. *Válvulas cardíacas e valvulopatias*. [Online]

Available at: <https://www.hospitaldaluz.pt/pt/dicionario-de-saude/valvulas-cardiacas-e-valvulopatias>

[Acedido em 2 Fevereiro 2023].

Ritchie, H. & Roser, M., 2019. *Causes of Death, Our World in Data*. [Online]

Available at: <https://ourworldindata.org/causes-of-death>

[Acedido em 05 Fevereiro 2022].

Roberts, L., 2020. *Understanding the Mel Spectrogram*. [Online]

Available at: <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53>

[Acedido em 31 Março 2023].

Roberts, L., 2020. *Understanding the Mel Spectrogram*. [Online]

Available at: <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53>

[fca2afa2ce53](#)

[Acedido em 1 Abril 2023].

Roell, J., 2017. *Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data*. [Online]

Available at: <https://towardsdatascience.com/understanding-recurrent-neural-networks-the-preferred-neural-network-for-time-series-data-7d856c21b759>

[Acedido em 21 Fevereiro 2023].

Saaty, T. L., 1990. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1), pp. 9-26.

Samuel, 2022. *A Stethoscope Can Detect A Variety Of Heart Problems*. [Online]

Available at: <https://www.excel-medical.com/a-stethoscope-can-detect-a-variety-of-heart-problems/>

[Acedido em 25 Fevereiro 2023].

Sharma, S., Sharma, S. & Athaiya, A., 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, IV(12), pp. 310-316.

Simonyan, K. & Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition.

TensorFlow, 2023. *Introduction to TensorFlow*. [Online]

Available at: <https://www.tensorflow.org/learn>

[Acedido em 13 Fevereiro 2023].

Tiobe - The Software Quality Company, 2023. *The Python Programming Language*. [Online]

Available at: <https://www.tiobe.com/tiobe-index/python/>

[Acedido em 13 Fevereiro 2023].

Vargas, A. C. G., Paes, A. & Vasconcelos, C. N., 2016. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. *Proceedings of the XXIX Conference on Graphics, Patterns and Images*, pp. 1-4.

Vimalajeewa, D., Lee, C. & Vidakovic, B., 2023. A Method for Detecting Murmurous Heart Sounds based on Self-similar Properties. pp. 1-15.

Vimal, B. et al., 2021. MFCC Based Audio Classification Using Machine Learning. *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*, pp. 1-4.

Walker, B. et al., 2022. Dual Bayesian ResNet: A Deep Learning Approach to Heart Murmur Detection.

World Health Organization, 2021. *Cardiovascular diseases (CVDs)*. [Online]

Available at: <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases->

(cvds)

[Acedido em 2022 02 04].

Xu, Y., Bao, X., Lam, H.-K. & Kamavuako, E. N., 2022. Hierarchical Multi-Scale Convolutional Network for Murmurs Detection on PCG Signals.

Yasar, K. & Lewis, S., 2023. *What is PyTorch?*. [Online]

Available at: <https://www.techtarget.com/searchenterpriseai/definition/PyTorch>

[Acedido em 14 Fevereiro 2023].

Yıldırım, S., 2020. *10 Must-Know Classification Metrics for Machine Learning | by.* [Online]

Available at: <https://towardsdatascience.com/10-must-know-classification-metrics-for-machine-learning-2ce3a4ad256e>

[Acedido em 21 Fevereiro 2023].

## Anexos

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2,4,6,8	Valores intermediários	Quando se procura uma condição de compromisso entre duas definições

Figura 56 - Escala fundamental – Níveis de importância de comparações (Saaty, 1990)