INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Deteção de patologia em sons cardíacos usando deep learning

JOSÉ PEDRO INEZ DE MEIRA TORRES
Junho de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Deteção de patologia em sons cardíacos usando deep learning

José Pedro Inês Meira Torres

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de informação e conhecimento

Orientador: Elsa Maria de Carvalho Ferreira Gomes

Co-orientador: Jorge Oliveira

Porto, 06 2021

Dedicatória

Dedico esta dissertação aos meus pais e irmãos por todo o incentivo e ajuda ao longo deste percurso.

À minha namorada por todo o apoio.

Resumo

A auscultação é a principal técnica utilizada pelos profissionais de saúde para a deteção de doenças cardiovasculares (DCV). Esta técnica identifica padrões patológicos inerentes no som, o que permite determinar o estado de saúde do coração.

Os profissionais de saúde estão capacitados para realizar diagnósticos clínicos a partir da avaliação da auscultação dos sons cardíacos. No entanto, os diagnósticos podem estar sujeitos a erros devido a fatores intrínsecos ao profissional (erro humano) ou devido a fatores externos como ruídos do meio ambiente. Estes erros por parte dos profissionais de saúde podem originar a realização de tratamentos desnecessários ou por outro lado a sua inexistência.

Por este motivo, a criação de um sistema que consiga apoiar os profissionais de saúde na deteção de anormalidades nos batimentos cardíacos é considerada uma mais-valia.

Esta dissertação tem como objetivo criar uma metodologia que consiga detetar automaticamente a existência de DCV através de sons cardíacos. Com a finalidade de desenvolver esta metodologia, foram propostas duas abordagens distintas. A primeira, é baseada em *transfer learning* de modelos pré-treinados. Esta abordagem permite explorar diferentes arquiteturas de *Convolutional Neural Networks* e aplicá-las ao contexto de sons cardíacos. Na segunda abordagem, é proposta uma arquitetura de *Convolutional Neural Network* para a classificação dos dados.

Para treinar e avaliar estas abordagens foram usados vários conjuntos de dados balanceados e não balanceados, com a finalidade de testar as abordagens de forma eficiente e compará-las com as abordagens existentes.

O melhor resultado obtido foi pela segunda abordagens utilizando *SMOTE*, com um *overall* de 86.69% e uma *accuracy* de 90.74%.

Palavras-chave: PCG, deep learning, doenças cardíacas, *CNN*, *transfer learning*, *SMOTE*

Abstract

Auscultation is the main technique used by health professionals to detect cardiovascular diseases (CVD). This technique identifies pathological patterns inherent in sound, which allows determining the health status of the heart.

Health professionals are trained to perform clinical diagnoses based on the assessment of auscultation of cardiac sounds. However, diagnoses can be subject to errors due to factors intrinsic to the professional (human error) or due to external factors such as environmental noise. These errors on the part of health professionals can lead to unnecessary treatments or, on the other hand, their non-existence.

For this reason, the creation of a system that can support health professionals in detecting abnormalities in the heartbeat is considered an asset.

This dissertation aims to create a methodology that can automatically detect the existence of CVD through cardiac sounds. To develop this methodology, two different approaches have been proposed. The first approach is based on transfer learning of pre-trained models in ImageNet images. In this approach, different architectures of Convolutional Neural Networks are explored and applied to the context of cardiac sounds. In the second approach, a Convolutional Neural Network architecture was proposed for data classification.

To train and evaluate the approaches, several balanced and unbalanced datasets were used. This allows you to test approaches efficiently and compare them with existing approaches.

The best result obtained was by the second approach using SMOTE, with a total of 86.69% and a precision of 90.74%.

Keywords: PCG, deep learning, heart disease, CNN, transfer learning, SMOTE

Agradecimentos

Gostaria de agradecer a todos aqueles que me apoiaram para a finalização deste percurso académico.

À professora Elsa Ferreira Gomes pela oportunidade de me orientar na conclusão deste trabalho. Agradeço toda a disponibilidade e toda a orientação que me deu durante toda a dissertação.

Ao professor Jorge Oliveira por todas ajuda e sugestões dadas.

À minha namorada por todo o apoio que me deu durante o meu percurso académico.

Principalmente, tenho de agradecer à minha família por todo o apoio que me deram e a ajuda que permitiu terminar o percurso académico.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

DCV	Doenças Cardiovasculares
MCD	Meios Complementares de Diagnóstico
AHP	Avaliação Assistida por Computador
PVD	Países em Vias de Desenvolvimento
IA	Inteligência Artificial
PCG	Fonocardiograma
ECG	Eletrocardiograma
MFCC	*Mel-frequency Cepstral Coefficient*
RNN	*Recurrent Neural Network*
WT	*Wavelet Transform*
PANN	*Propose Pretrained Audio Neural Network*
FD	*Fractal Dimension*
BLSTM	*Bidirectional Long Short-Term Memory*
LSTM	*Long Short-Term Memory*
GRU	*Gated Recurrent Unit*
HSMM	*Logistic Regression-Hidden Semi-Markov Models*
SGD	*Stochastic gradient descent*
FFN	*Feedforward neuronal network*
PMV	*Prolapsed Mitral Valve*
RQA	*Recurrence Quantification Analysis*
FAST	*Function Analysis System Technique*
O2	Oxigénio

CO2	Dióxido de Carbono
ML	*Machine Learning*
SMOTE	*Synthetic* Minority Oversampling Technique
TF	*Time Features*
NPD	*New Product Development*
FEI	*Front End of Innovation*
SP	Soma Ponderada
ROC	*Receiver Operator Characteristic*
AUC	*Area Under the Curve*
TPR	*True Positive Rate*
FPR	*False Positive Rate*

1 Introdução

1.1 Contextualização

As Doenças Cardiovasculares (DCV) são consideradas a principal causa de morte no mundo. Estima-se que 17.9 milhões de pessoas morreram com DCV em 2019, o que representa 32% do número de mortes a nível mundial (World health organization, 2021).

O método mais usual para deteção de doenças consiste na análise clínica, que inclui a auscultação cardíaca através da utilização de um estetoscópio (Mustafa *et al.*, 2020). O estetoscópio permite ao profissional de saúde identificar informações cardíacas, tais como, o ritmo cardíaco, válvulas e outras anomalias do coração. Com o avanço da tecnologia, os estetoscópios têm evoluído, principalmente, no ganho de qualidade dos sons captados. Atualmente, está disponível para especialistas o estetoscópio digital, o qual permite ampliar o som e reduzir o ruído, melhorando a perceção do utilizador em diagnósticos mais precisos (Prodoctor, 2019).

O processo de auscultação está diretamente relacionado com o Fonocardiograma (PCG), tratando-se de uma representação gráfica dos diferentes sons originados pelo coração e seus grandes vasos. Desta forma, a deteção de anomalias do coração pode ser realizada através da análise dos sinais do PCG, o que permite encaminhar o paciente para análises posteriores mais específicas (Nogueira, Ferreira and Jorge, 2017).

Outro método utilizado na triagem cardíaca é o Eletrocardiograma (ECG), que é um exame que regista a atividade elétrica do coração. É considerado um dos métodos de eleição para deteção de anomalias cardíacas, uma vez que qualquer acontecimento mecânico reflete-se a nível elétrico (Papadaniil and Hadjileontiadis, 2014). Este é um dos Meios Complementares de Diagnóstico (MCD) mais utilizado ao nível da emergências e urgências, assim como um frequente exame de rotina. A sua vantagem passa por ser rápido, fácil e versátil, no entanto, em comparação com a auscultação é menos económico, com mais demanda de *hardware* e é difícil implementar em recém-nascidos (Singh and Cheema, 2013; Papadaniil and Hadjileontiadis, 2014).

Com o surgimento dos *smartphones* e estetoscópios digitais, o PCG tem sido um tópico muito abordado ao longo da literatura (Potes *et al.*, 2016). Deste modo, a tecnologia proporciona um aumento da eficiência na área da saúde, o que permite agilizar vários processos e suprimir limitações existentes. Neste sentido, a investigação sobre sons cardíacos para o reconhecimento de DCV torna-se uma ferramenta fundamental para a deteção e intervenção precoce.

1.2 Problema

A deteção precoce de DCV por parte de um profissional de saúde é fundamental para elaborar um plano de tratamento mais eficaz e, desta forma, reduzir custos e melhorar a esperança média de vida do seu paciente (Singh and Cheema, 2013; Latif *et al.*, 2018).

Assim sendo, como supracitado, a auscultação constitui o primeiro passo para a deteção de DCV. A sua utilização por profissionais de saúde poderá ser influenciada pelo estado emocional da pessoa, pelo ruído causado pelo organismo (por exemplo, ruído pulmonar) e pelo ambiente envolvente. Isto aumenta a probabilidade de erro em diagnósticos por parte do profissional de saúde, o que pode originar a realização de exames e tratamentos desnecessários.

Neste trabalho, foram estudadas alternativas para desenvolver um sistema que consiga determinar de forma eficiente as DCV através da utilização dos sons cardíacos, e assim auxiliar os profissionais de saúde na deteção precoce destes problemas.

1.3 Objetivos

O principal objetivo deste projeto consiste na criação de um sistema que sirva de apoio à triagem na deteção de patologias cardiovasculares. Para isso ser possível, é necessário definir e aplicar um conjunto de métodos que consigam extrair informação relevante dos sons cardíacos para posteriormente serem aplicados modelos de *deep learning*.

Pretende-se também que os resultados obtidos sejam comparados com resultados de trabalhos anteriormente desenvolvidos.

1.4 Motivações

A grande motivação para a escolha deste tema foi a possibilidade de ajudar os profissionais de saúde da área cardíaca na deteção de doença. Através da diminuição do tempo de deteção de DCV é possível atuar rapidamente no seu tratamento, em que o tempo é uma peça fundamental para o prognóstico da doença.

1.5 Resultados esperados

O resultado esperado do trabalho que se descreve nesta dissertação é a criação de uma metodologia que permita a deteção automática de DCV através do som, assim como, a diminuição de erros humanos. Para a sua concretização, foram desenvolvidos vários modelos com o objetivo de selecionar o que apresenta maior fiabilidade na deteção de doenças cardíacas.

1.6 Análise de valor

As DCV representam 32% do número de mortes a nível mundial, sendo que, Portugal segue esta tendência, em que a doença corresponde a 29.5% das mortes (Direção Geral de Saude, 2016; World health organization, 2021).

Os Países em Vias de Desenvolvimento (PVD) são responsáveis por três quartos do número de mortes por esta doença. Isto acontece devido a serviços de saúde pouco eficazes e equitativos, à pobreza da população e aos custos elevados dos exames (World health organization, 2021).

A auscultação é frequentemente utilizada pelos profissionais de saúde para a deteção de DCV. Contudo, a taxa de acerto do diagnóstico é de 80% para um cardiologista experiente, enquanto que, para um estudante ou médico em início de carreira é de apenas 20%-40% (Noponen *et al.*, 2007; Latif *et al.*, 2018).

1.7 Abordagem preconizada

Numa primeira fase foram estudados os vários artigos científicos sobre a classificação de doenças cardíacas. Esses estudos permitiram selecionar as melhores abordagens e métodos que vão de encontro ao objetivo do trabalho e também selecionar os conjuntos de dados mais estudados nesta área.

Após efetuada a revisão bibliográfica, foram definidas duas abordagens diferentes. A primeira abordagem consiste na aplicação de *transfer learning* de modelos pré-treinados (Boulares, Alafif and Barnawi, 2020). Isto possibilita a criação de modelos mais hábeis, que não são possíveis criar com os conjuntos de dados existentes. A segunda abordagem baseia-se na criação de modelos, utilizando uma arquitetura de *Convolutional Neural Network* (*CNN*) (Khan *et al.*, 2020). Para a extração dos atributos do som foram selecionados três algoritmos, *STFT Spectrogram*, *Mel-Frequency Cepstral Coefficients* (*MFCC*) e *Mel Spectrogram*.

Para testar as abordagens definidas, os conjuntos de dados foram combinados, permitindo assim, criar vários conjuntos de dados diferentes para treinar os modelos.

1.8 Estrutura do documento

Esta dissertação é organizada pelos seguintes capítulos:

- No primeiro capítulo (Introdução), pretende-se dar uma pequena explicação sobre o projeto, onde é explicado o contexto, o problema, os objetivos, as motivações, os resultados esperados, a análise de valor e a abordagem preconizada;
- No segundo capítulo (Enquadramento Teórico), é apresentado um enquadramento teórico dos conhecimentos relevantes sobre o coração;

- No terceiro capítulo (Análise de Valor, é apresentada a análise de valor do sistema desenvolvido;
- No quarto capítulo (Estado de Arte), é apresentado um contexto mais aprofundado do sistema desenvolvido, incluindo conjunto de dados a ser utilizado, exemplos de artigos relevantes e tecnologias utilizadas;
- No quinto capítulo (Design), são descritos os passos a serem utilizados para o desenvolvimento do projeto, a metodologia de avaliação e as abordagens adotadas;
- No sexto capítulo (Avaliação), são apresentados os conjuntos de dados, a comparação das abordagens definidas e a formulação da hipótese.

2 Enquadramento Teórico

Neste capítulo são apresentados os principais conceitos deste trabalho. É explicado o funcionamento do coração, a origem dos sons cardíacos e uma breve descrição do estetoscópio.

2.1 Coração

Há aproximadamente 370 anos deu-se a descoberta de que o bombeamento cardíaco é essencial para a circulação do sangue por todos os vasos sanguíneos do organismo. No entanto, os conhecimentos que hoje dispomos do seu funcionamento, regulação e dos seus tratamentos são muito mais recentes (Seeley, Stephens and Tate, 2011).

O coração possui cinco superfícies: base (posterior), diafragmática (inferior), esternocostal (anterior), e superfícies pulmonares direita e esquerda. Internamente, é dividido em quatro cavidades, dois átrios ou aurículas, o direito e o esquerdo, e dois ventrículos, o direito e o esquerdo. Como é visível na Figura 1, a parte direita do coração é responsável por receber o sangue com altos níveis de Dióxido de Carbono (CO2), denominado de sangue venoso, das veias cavas inferior e superior, que transportam este tipo de sangue vindo de todas as partes do corpo. O sangue venoso, que entra no coração, é conduzido da aurícula direita para o ventrículo direito, através da válvula tricúspide, responsável por impedir o refluxo do sangue. Seguidamente, o sangue é impulsionado para os pulmões, onde irá ser oxigenado, através da artéria pulmonar, que se ramifica em esquerda e direita, dirigindo-se para o pulmão esquerdo e direito, respetivamente. Após as trocas gasosas que ocorrem nos pulmões, o sangue, que passa a ser rico de Oxigénio (O2), é encaminhado para a aurícula esquerda pelas veias pulmonares passando a válvula bicúspide ou mitral para o ventrículo esquerdo. Quando o sangue está no ventrículo esquerdo é novamente impulsionado, neste caso, para o resto do corpo através da artéria aorta que depois se ramifica com esse objetivo. Para além das válvulas já referidas, existem ainda, no coração, a válvula da veia pulmonar e a válvula aórtica. Importante referir que por todo o sistema circulatório existem válvulas que impedem o refluxo de sangue, ou seja, impedem que o sangue volte para trás indo contra o seu fluxo normal. Assim sendo, estas tornam-se peças importantes no correto funcionamento da bomba cardíaca (Seeley, Stephens and Tate, 2011).

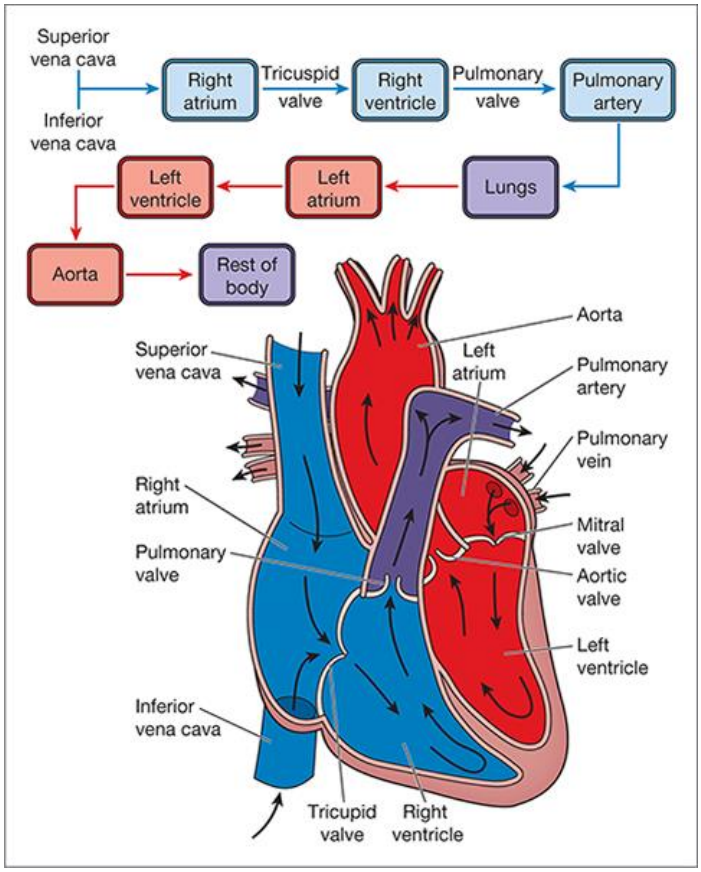


Figura 1 - Ciclo cardíaco (Flamm *et al.*, 2020)

2.2 Sons Cardíacos

Os sons cardíacos são originados pela contração das válvulas que bombeiam o sangue para todo o corpo. Esse sangue segue uma única direção se o coração estiver a funcionar corretamente, o que é garantido pelas válvulas atrioventriculares (tricúspide e mitral). Isso é possível ao abrir e fechar em coordenação exata com a ação de bombeamento do coração, garantindo que flua das aurículas para os ventrículos e nunca ao contrário. Os sons cardíacos são produzidos por este evento específico. Por fim, o sangue flui dos ventrículos para fora do coração através das válvulas semilunares conhecidas por válvula pulmonar e válvula aorta.

Os batimentos cardíacos têm dois sons básicos, nomeadamente S1(lub) e S2(dub). Cada som corresponde a um período chamado sístole e diástole. Estes formam a sequência do batimento cardíaco normal “lub dub, lub dub”, em que o tempo de “lub” para o “dub” é menor que de “dub” para o próximo “lub” (Getz, 2012).

A sístole é causada pela pressão ventricular sobre as válvulas tricúspide e mitral. Desta forma, o sangue é forçado a sair do ventrículo mantendo fechadas as válvulas atrioventriculares, o que não permite o refluxo para as aurículas. O som S1(lub) é produzido pelo fecho das válvulas atrioventriculares.

A diástole acontece à medida que os músculos dos ventrículos relaxam. Este relaxamento origina que as pressões das aurículas sejam superiores às dos ventrículos, forçando a abertura das válvulas atrioventriculares e o encerramento das válvulas semilunares. Consequentemente, o fecho das válvulas produz o som S2(dub) (Gomes, Jorge and Azevedo, 2013).

Embora S1 e S2 sejam os sons mais reconhecíveis do ciclo cardíaco, existem outros sons produzidos pelo sistema elétrico como é o exemplo de murmúrios, velocidade do sangue, terceiro som (S3) e quarto som (S4). O som S3 resulta das vibrações das válvulas ventriculares como resultado do seu rápido enchimento, enquanto que, o som S4 ocorre quando os átrios se contraem durante a segunda fase de enchimento ventricular (Singh and Cheema, 2013).

2.3 Sons anormais do coração

Quando existe uma anomalia física a nível cardíaco, o fluxo do sangue é afetado e reflete-se no som do batimento cardíaco. Esta anomalia do fluxo pode ser devido a várias razões incluindo arritmias cardíacas, sinais de insuficiência e defeitos nas válvulas.

Os corações humanos, durante o funcionamento, podem libertar sons que podem ou não ser origem de uma doença, um exemplo é a ocorrência do som cardíaco S3. Este pode não ter origem patológica, pois é frequentemente encontrado em crianças, ou se encontrado num adulto, pode ser sinal de insuficiência cardíaca. O som cardíaco S4 geralmente está associado a uma patologia. Outro exemplo, são os murmúrios fisiológicos ou patológicos, causados pela velocidade do fluxo sanguíneo (Nogueira, Ferreira and Jorge, 2017). Estes podem ser sistólicos, quando ocorre durante o tempo de contração, ou diastólicos, permitindo que algum sangue flua de volta para o ventrículo.

A contração prematura do coração (Extrassístole) pode ser identificada através do som por um batimento extra no ritmo cardíaco “lub lub dub” ou pela falta de um” lub dub dub” como pode ser observado na Tabela 1. Contudo, pode não se tratar de uma doença cardíaca, pois pode ser facilmente encontrado em crianças e, por vezes, em adultos. (Gomes and Pereira, 2012).

Tabela 1 - Representação dos diferentes sons do sistema cardíaco (Getz, 2012)

Tipo de sons cardíacos	**Característica dos sons cardíacos**
Normal	….lub…dub……lub…dub……lub…dub……lub…dub……
Murmúrio	…lub..****...dub……..lub..****..dub……..lub..****..dub ….. ou …lub…….dub…******….lub……. dub…******….lub….
Extrassístole	…lub.lub…….dub……..lub.lub…….dub……..lub.lub……..dub……. ou …lub……dub.dub……..lub…….dub.dub……..lub……..dub. dub..

2.4 Estetoscópio

A auscultação cardíaca é uma prática já conhecida antes do século XVII. Segundo Corvisart, em 1818, é a capacidade de ouvir um som ao se colocar o ouvido no tórax de um paciente (Cristian *et al.*, 1995). Este método tinha o nome de ausculta direta. Devido a ser um método pouco prático, René Laennec, aprendiz de Corvisart, em 1819, construiu o primeiro estetoscópio. Era constituído por um tubo de madeira com duas extremidades, uma delas era colocada no ouvido do médico e a outra no tórax do paciente. Isso possibilitou uma audição mais clara dos batimentos cardíacos e assim nasceu a técnica de auscultação indireta.

Após a invenção de René Laennec, vários modelos foram construídos desde então até à chegada dos estetoscópios digitais (Cristian *et al.*, 1995). Os estetoscópios digitais foram desenvolvidos na década de 1950 e foram introduzidos em contexto clinico pela 3M em 1995 (Silverman and Balk, 2019). Estes instrumentos melhoraram a captação do som e possibilitaram a gravação dos sons para consulta posterior em computadores. Desde então, continua a existir enormes pesquisas na área com a motivação de chegar a uma solução ideal.

A auscultação cardíaca relaciona-se com vibrações transmitidas através das várias estruturas sobrepostas como os músculos, pele, vasos sanguíneos, estruturas ósseas, entre outras (Cristian *et al.*, 1995). O registo dessas ondas vibratórias é feito através de um PCG.

O estetoscópio digital converte as ondas sonoras acústicas em sinais eletrónicos (Silverman and Balk, 2019), permitindo aplicar filtros no som de forma a reduzir o ruído, ampliar, reproduzir e gravar os sons e por fim representar essas informações em gráficos que auxiliam na análise.

Figura 2 - Estetoscópio digital da 3M (Antiques, 2019)

3 Análise de Valor

Neste capítulo, é apresentado o projeto num contexto mais aprofundado utilizando a análise de valor. A análise de valor é uma parte fundamental para as inovações, em que o seu principal objetivo é avaliar como aumentar o valor de um produto ou serviço minimizando os custos, sem sacrificar a sua qualidade (HUGHES, 1996).

3.1 *New Concept Development Model* (*NCD*)

O processo de inovação pode ser dividido em três passos: *Front End of Innovation* (*FEI*), *New Product Development* (*NPD*) e Comercialização. O *FEI* é um componente crítico do processo de inovação, em que as escolhas feitas neste passo determinam as opções de inovação disponíveis para o desenvolvimento e comercialização (Koen, Bertels and E. J. Kleinschmidt, 2014). Por este motivo, foi desenvolvido o modelo *NCD*, que fornece uma terminologia e vocabulário comum para entender as atividades que ocorrem no *FEI*. Este modelo divide-se em três partes: motor, roda e aro. O motor representa a visão, estratégia e cultura. A roda, parte interna do modelo, contém os cinco elementos da atividade do *FEI*. O aro, consiste nos fatores ambientais externos que influenciam o motor e os elementos de atividade. O modelo tem um formato circular, como é visível na Figura 3, para indicar que as ideias iteram entre os cinco elementos. As setas que apontam para o modelo representam pontos de partida e indicam que os projetos podem começar na identificação de oportunidades ou na geração e enriquecimento de ideias. A seta que sai do modelo representa os projetos que saem entrando no *NPD* (Koen, Bertels and E. Kleinschmidt, 2014)

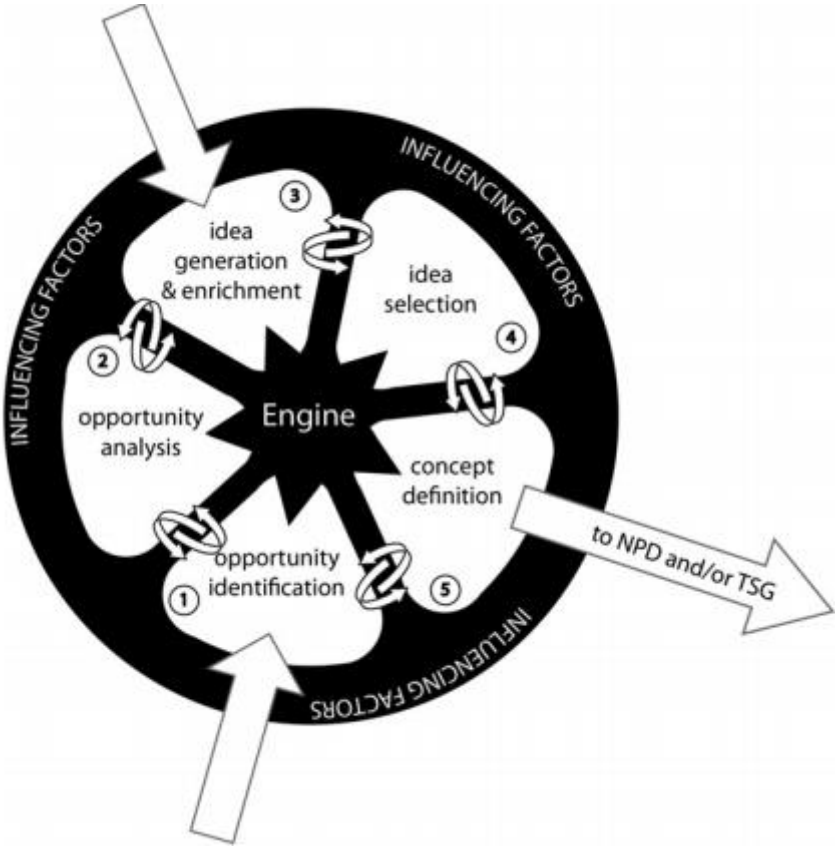


Figura 3 - Modelo NCD (Koen, Bertels and E. J. Kleinschmidt, 2014)

3.1.1 Identificação da oportunidade

Como referido anteriormente, as DCV são consideradas uma das mais prevalentes e a principal causa de morte a nível mundial. Em 2019, foram contabilizadas cerca de 17.9 milhões de pessoas mortas, o que representa 32% do número de mortes a nível mundial. Pelo menos três quartos dessas mortes ocorrem nos PVD (World health organization, 2021).

Em Portugal, as DCV também são a principal causa de morte (Figura 4), perfazendo uma percentagem de 29.5. O segundo lugar é ocupado pelos tumores malignos, e em terceiro encontram-se os problemas do aparelho respiratório. No entanto, Portugal encontra-se abaixo da percentagem média de mortes da União Europeia (Direção Geral de Saude, 2016).

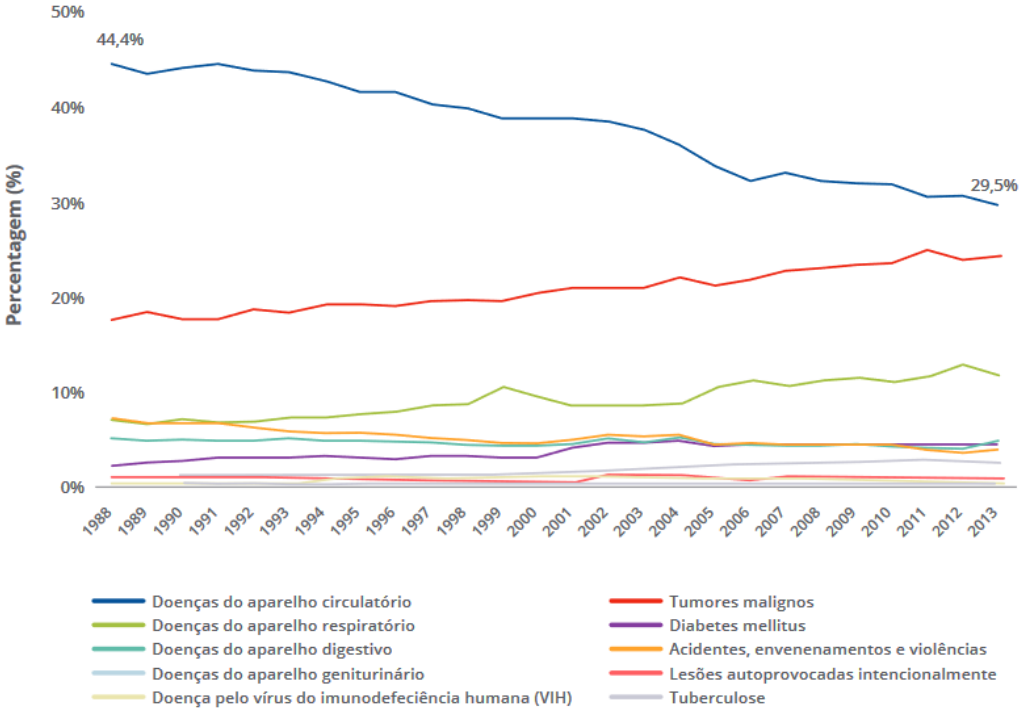


Figura 4 -Percentagem das causas de morte em Portugal (Direção Geral de Saude, 2016)

Comparativamente com os países desenvolvidos, nos PVD, as pessoas não têm acesso a um serviço de saúde eficaz e equitativo que atenda às suas necessidades. Consequentemente, as doenças não são detetadas em tempo útil, o que dificulta a criação de planos de tratamento adequados (World health organization, 2021).

Outra realidade nos PVD é o contributo das DCV para a pobreza da população, devido aos gastos catastróficos que as pessoas têm com a saúde (World health organization, 2021).

A deteção de DCV é concretizada através do ECG, ecocardiograma e a auscultação através do estetoscópio. A auscultação é considerada o primeiro passo para a deteção da doença cardíaca e, por vezes, o único método presente em meio hospitalar. O ECG e o ecocardiograma são exames mais dispendiosos que necessitam de profissionais capacitados, sendo o preço médio de 158 dólares e 1500 dólares, respetivamente (Latif *et al.*, 2018; Mdsave, 2021).

A auscultação apesar de ser um método económico, trata-se de um método complexo que depende, predominantemente, da experiência e do conhecimento do médico e da sua capacidade auditiva. Desta forma, um diagnóstico realizado por um cardiologista experiente consegue ter uma taxa de acerto que ronda os 80%, enquanto um estudante ou médico em início de carreira apresenta uma taxa de acerto entre 20%-40% (Noponen *et al.*, 2007; Latif *et al.*, 2018).

3.1.2 Análise de oportunidade

A oportunidade para a realização deste projeto surge devido à falta de um sistema no meio hospitalar que permita apoiar os profissionais no processo de auscultação.

De acordo com a Secção 3.1.1, existe uma necessidade de criar opções mais económicas que possam ser introduzidas nos PVD, substituindo assim, métodos mais caros como o ECG e o ecocardiograma, que necessitam de meios humanos capacitados para a sua utilização. O sistema desenvolvido neste projeto, possibilita a sua utilização por qualquer pessoa, apesar de ser recomendado que a sua interpretação seja realizada por pessoas capacitadas.

3.1.3 Geração e enriquecimento de ideias

A geração de ideias iniciou-se com a análise de vários trabalhos publicados. Esta análise permitiu a identificação dos diversos métodos utilizados no carregamento e processamento do som e as técnicas utilizadas. Foram também estudadas técnicas para a geração do resultado, bem como as linguagens de programação utilizadas.

3.1.4 Seleção de ideias

Dada a análise realizada anteriormente, através dos resultados obtidos pelos trabalhos científicos, foram definidas as fases principais: pré-processamento, extração de atributos e classificação. Para cada uma das fases foram analisados e definidos quais os métodos a utilizar.

Para a criação dos modelos de classificação, é adotada a abordagem *transfer learning*. Isto deve-se há falta de conjuntos de dados com a quantidade necessária para a criação de algoritmos de *deep learning*.

3.1.4.1 *Analytic Hierarchy Process* (AHP)

O *Analytic Hierarchy Process* (AHP) é uma metodologia introduzida por Thomas. L. Saaty em 1980, tendo como objetivo fornecer uma teoria geral de medição. A teoria baseia-se num conjunto de cálculos matemáticos que possibilitam a avaliação da força relativa a um conjunto de vários fatores, comparando-os, fornecendo assim resultados que apoiam a decisão (Saaty, 1987). Neste sentido, a aplicação desta metodologia tem como objetivo a escolha da *framework* de *deep learning* a ser utilizada.

Para isso, foram avaliadas e selecionadas as principais *frameworks* de *deep learning*, Tensorflow, Pytorch, Caffe, CNTK (Opala, 2019; ODSC, 2020; Pathmind, 2020; Shivanandhan, 2020; Goyal, 2021). De seguida, foram definidos os seguintes critérios: aprendizagem, *performance*, conhecimento e popularidade. Considerando os critérios definidos, foi desenvolvido o diagrama hierárquico (Figura 5).

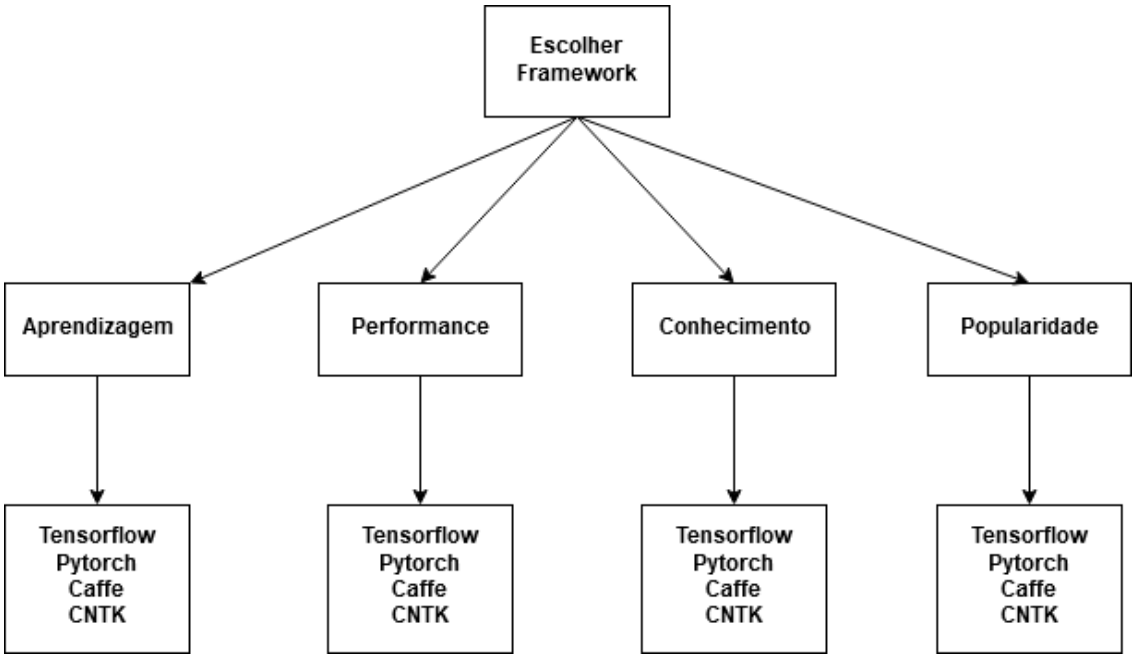


Figura 5 - Critérios de seleção

Conforme os resultados presentes no Anexo A, a *framework* mais indicada para o desenvolvimento do projeto trata-se do *Keras/Tensorflow* (49%).

3.1.5 Definição do conceito

O sistema desenvolvido deve ser capaz de receber uma gravação do som cardíaco, eliminar o ruído, identificar os ciclos cardíacos (S1 e S2) e extrair a informação relevante do som. Por fim, através desses dados, deve conseguir identificar se o som corresponde a um batimento normal ou anormal.

3.2 Proposta de Valor

A proposta de valor é uma declaração que descreve a forma como o seu produto ou serviço terá mais valor para o cliente comparativamente a ofertas semelhantes da concorrência. Esta proposta descreve de forma concisa os benefícios que o produto ou serviço oferece, bem como a forma como este ajuda a resolver o problema do cliente (Casey Newman, 2018).

A proposta de valor desenvolvida nesta dissertação passa pela criação de uma metodologia que consiga detetar automaticamente a existência de DCV através de sons cardíacos. Esta metodologia permite que os profissionais de saúde consigam identificar doenças de forma rápida, não evasiva e conveniente, sem que seja necessário proceder a exames específicos. O modelo CANVAS (3.2.1) demonstra a proposta a ser desenvolvida.

3.2.1 Proposta de valor CANVAS

A proposta de valor CANVAS foi desenvolvida pelo Dr. Alexander Osterwalder, cujo objetivo é a criação de uma ferramenta que permita garantir que um produto ou serviço esteja posicionado em torno dos valores e necessidades do cliente. Este modelo, é utilizado quando existe a necessidade de refinar uma oferta de um produto ou serviço existente ou quando uma nova oferta está a ser desenvolvida do zero (B2b International, 2014). Na Figura 6 é apresentado o modelo desenvolvido.

Figura 6 - Modelo *value proposition* CANVAS

O presente modelo é composto pelo produto e o cliente. O cliente é formado pelos ganhos, que representam os benefícios que este espera e precisa, e os tópicos que podem aumentar a probabilidade de adotar uma proposta de valor; as dores, correspondem às experiências negativas, emoções, riscos que o cliente experiência na sua utilização; o trabalho do cliente, representa, as tarefas funcionais, sociais e emocionais que este tenta realizar, os problemas a resolver e as necessidades que pretende satisfazer.

O produto é formado pelos criadores de ganho, os analgésicos e o produto/serviço. Os criadores de ganho identificam tópicos que possibilitam um produto ou serviço oferecer ganhos e um valor agregado ao cliente. Os analgésicos, apresentam pontos que permite aliviar as dores do cliente. Por último, os produtos e serviços geram ganho e sustentam a criação de valor para o cliente.

3.3 *Function Analysis System Technique* (*FAST*)

O modelo *FAST* foi desenvolvido por Charles Bytheway em 1964 e permite a criação de diagramas rápidos, de forma objetiva e orientados à função. Este modelo é construído de forma horizontal, da esquerda para a direita, utilizando formas gráficas que definem as funções

básicas, secundárias e os seus objetivos. As funções inseridas por ordem devem responder à lógica das perguntas “Como?” e “Porquê?" (Rains, 2002).

Através do modelo *FAST* representado na Figura 7 é possível entender que existe um som que passa por um conjunto de funções até ser possível determinar se existe alguma anormalidade.

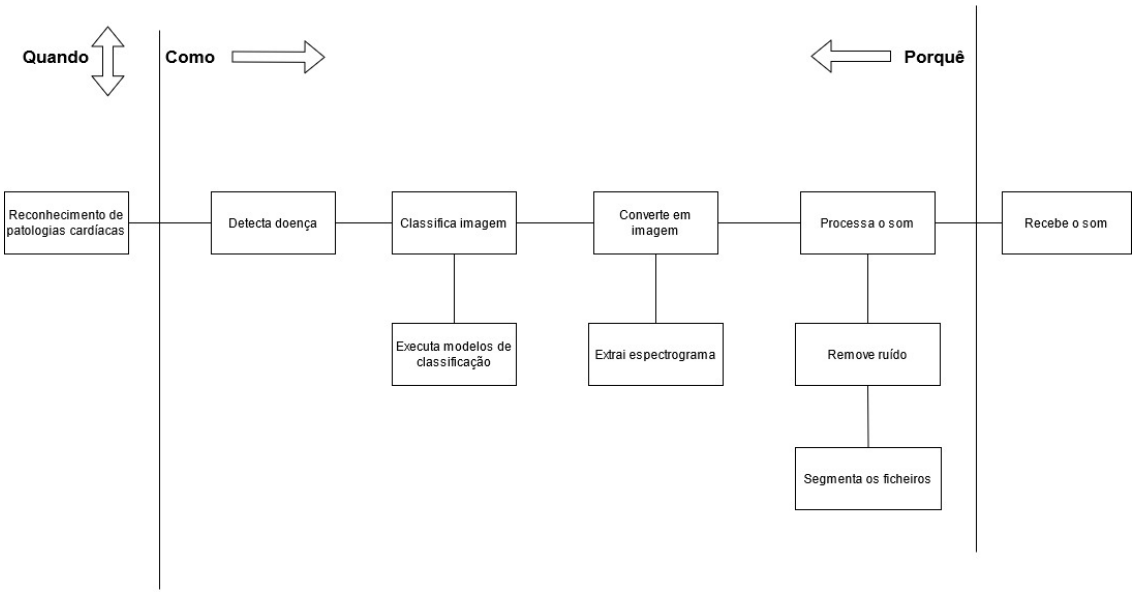


Figura 7 - Modelo *FAST*

Através da análise do modelo *FAST* é possível visualizar que do lado direto é apresentado uma função de baixa ordem “Receber o som” e no seu lado esquerdo a função de ordem superior “Reconhecimento de patologias cardíacas”. Essas funções são conectadas e organizadas pelo caminho crítico pela lógica “Como?/Porquê?”. Neste caminho, é possível identificar a função básica “Detecta doença” que descreve o objetivo do sistema descrito e todas as funções secundárias que descrevem como uma função básica é processada. Também podemos visualizar as funções secundárias “Executa modelos de classificação”, “Extrai espectrograma”, “Remove ruído” e “Segmenta os ficheiros”. Estas funções são definidas no caminho de cima para baixo e definem quando é que uma função básica deve acontecer.

4 Estado de Arte

Neste capítulo são apresentados os conjuntos de dados a serem utilizados para o desenvolvimento do trabalho descrito nesta dissertação. Nas secções seguintes, são apresentados dois trabalhos no qual o trabalho aqui descrito se insere. São também descritos vários estudos desenvolvidos por diversos autores e, por último, são apresentadas as tecnologias a serem utilizadas.

4.1 Conjunto de dados

4.1.1 PASCAL

O conjunto de dados (*dataset*) PASCAL é composto por um total de 312 gravações de sons cardíacos divididos em dois conjuntos separados denominados "*dataset* A" e "*dataset* B" (Getz, 2012). Estes sons foram captados na unidade de Cardiologia Unidade Materno-Fetal do Hospital Real Português no Recife, Brasil. As gravações foram feitas em crianças tendo uma duração de 1 a 10 segundos, em que foram anotadas algumas informações. Cada gravação foi caracterizada numa das três classes: Normal, *Murmur* e *Extrasystole*. A classe Normal (N) tem uma contagem de 200, a classe *Murmur* (M) conta 97 casos e a *Extrasystole* (E) 46 (Gomes, Jorge and Azevedo, 2014).

Como os sons foram recolhidos num contexto hospitalar, é possível identificar diversos tipos de ruídos, respiração do paciente e barulhos produzidos pelo movimento do microfone contra a pele ou roupa. Os sons foram gravados por aparelhos diferentes, o *dataset* A foi recolhido através de uma aplicação chamada *iStethoscope* apresentando uma qualidade bastante inferior aos do *dataset* B, recolhidos através de um *DigiScope*. Na Figura 8 é apresentado no lado esquerdo a aplicação *iStethoscope* e do lado direito o *DigiScope*.

Figura 8 - Sistema *collector iStethoscope* e *DigiScope* (Gomes *et al.*, 2013)

4.1.2 *PhysioNet Computing in Cardiology Challenge*

O conjunto de dados *PhysioNet* foi disponibilizado para um desafio de classificação de fonocardiogramas em 2016. Este conjunto é composto pela junção de 6 bases de dados (A - F), contendo um total de 3240 sons cardíacos (2575 normal e 665 anormal). As gravações foram efetuadas por equipas diferentes e conta com pacientes de diferentes faixas etárias, incluindo crianças, adultos e idosos. A duração das gravações varia de 5 segundos até pouco mais de 120 segundos (Liu *et al.*, 2016).

No trabalho que se descreve nesta dissertação foram utilizados os seis conjuntos de dados contendo os registos de um total de 2575 sons normais e 665 anormais (Chengyu Liu, 2016).

4.2 Abordagens anteriores

Nesta secção, são apresentados os dois artigos mais relevantes do projeto em que esta dissertação se insere. O trabalho descrito no primeiro artigo de Gomes *and* Pereira (2012) consiste na segmentação dos sons cardíacos (S1 e S2) e na sua classificação. No segundo artigo de Nogueira *et al.* (2019), o trabalho é baseado em duas abordagens diferentes, usando *MFCC* e *Motifs* em conjunto com *Time Features* (*TF*) para a extração de atributos.

4.2.1 *Classifying heart sounds using peak location for segmentation and feature construction*

O projeto descrito neste artigo foi desenvolvido no âmbito da participação num concurso chamado "*Classifying Heart Sounds* PASCAL *Challenge*" (Gomes and Pereira, 2012). O objetivo deste concurso consistia na criação de um mecanismo que conseguisse identificar patologias cardíacas. Este mecanismo poderia ser utilizado tanto num ambiente hospitalar como pelo paciente na sua casa. Por este motivo, foram fornecidos dois conjuntos de dados distintos, em que o primeiro (*dataset* A) foi recolhido através de um telemóvel e o segundo (*dataset* B) por um estetoscópio digital.

A metodologia apresentada no concurso por parte dos candidatos era dividida em duas etapas. Na primeira etapa, foi desenvolvido um método que permite a segmentação dos sons do tipo normal determinando o período da sístole (S1) e o período da diástole (S2). A segunda etapa, tratava-se da criação de métodos capazes de classificar a gravação de batimentos cardíacos em quatro classes para o *dataset* A (*Normal, Murmur, Extra Heart Sound e Artifact*) e em três classes para o *dataset* B (*Normal, Murmur and Extrasystole*).

4.2.1.1 Segmentação dos sons cardíacos

Como referido na Secção 4.2.1, a primeira etapa consiste na identificação dos períodos S1 e S2 nas gravações do tipo normal, fornecidas pelo concurso referido. Na primeira imagem da Figura 9 é possível ver um exemplo de um PCG gerado através de um dos ficheiros.

Antes de dar início à segmentação dos dados, foi realizado o pré-processamento dos mesmos. O pré-processamento inicia-se com a redução da frequência do sinal utilizando a função *decimate* do *MatLab.* Este processo pode levar a que alguns sinais se tornem indistinguíveis, conhecido na área de processamento de sinal como *aliasing*. Para a eliminação destes sinais indesejáveis, foi utilizado um filtro passa-faixa *Chebyshev* tipo I de 5ª ordem com frequência de corte inferior de 100Hz e frequência de corte superior de 882 Hz. Este processo é apresentado na Figura 9 (a segunda imagem consiste ao processo de *decimate* e a terceira imagem corresponde à aplicação do filtro passa-banda). Por fim, o sinal é normalizado relativamente ao máximo absoluto do sinal.

A segmentação dos dados inicia-se com o cálculo do envelope *Shannon* do sinal normalizado. Esse processo consiste no cálculo da média da energia de *Shannon* numa janela contínua de 0.02 segundos e com 0.01 segundos de sobreposição (Liang, 1997), resultando numa curva normalizada da energia *Shannon*. A essa curva é aplicada uma função disponibilizada pelo *Matlab* que permite determinar os valores máximos de onda, que correspondem aos picos S1 e S2. A terceira imagem da Figura 9 apresenta a deteção dos picos na curva normalizada da energia de *Shannon.*

Através deste processo foi possível identificar com sucesso os picos S1 e S2 no som original (quarta imagem da Figura 9), sendo o método eleito pelo júri e que ganhou o concurso. Contudo, o método possui ainda algumas falhas em discriminar corretamente o pico S1 do S2 devido à dificuldade de identificar a que corresponde o primeiro pico (S1 ou S2) (Gomes and Pereira, 2012).

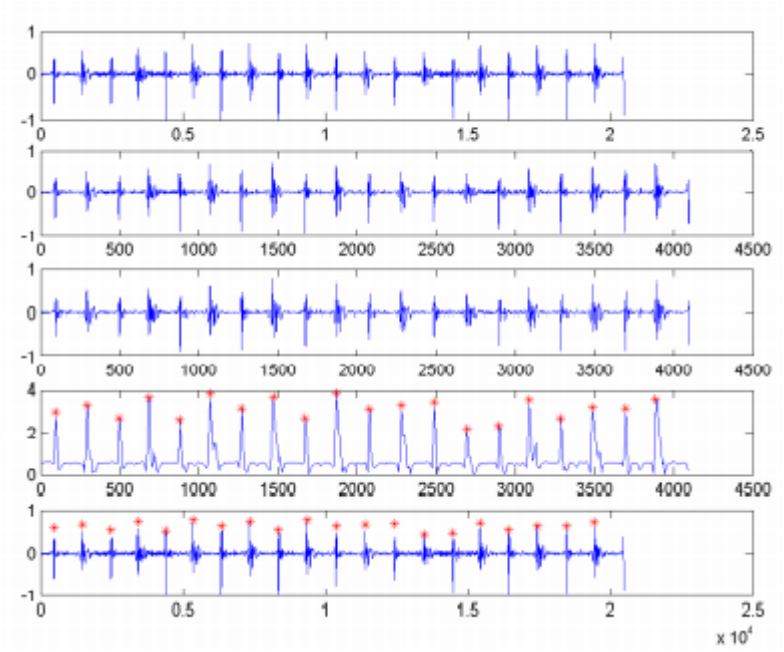


Figura 9 - Deteção dos picos no sinal do som cardíaco (Gomes and Pereira, 2012)

4.2.1.2 Classificação dos sons cardíacos

A segunda etapa do concurso consistia na criação de um método capaz de diferenciar os sons cardíacos. Para tal, foi fornecido um conjunto de dados de teste sem identificação em que o método proposto deveria ter a capacidade de identificar a sua classe (*Normal, Murmur e Extrasystole).* Esta etapa iniciou-se com a extração de seis atributos dos sinais processados na etapa anterior, que correspondem à média, mediana e desvio padrão das distâncias entre S1 e S2. Foi assumido que S2 é maior que S1, para batimentos cardíacos normais (Kumar, 2006). Para determinar cada ciclo cardíaco e o seu intervalo sistólico, foi calculada a distância entre S1 e S2 para cada segmento (Gupta *et al.*, 2007). O maior intervalo entre dois sons foi considerado o correspondente à diástole e o som do lado direito foi designado como S1 e o som do lado esquerdo foi designado como S2. A Figura 10, ajuda a compreender as distâncias reais entre os ciclos cardíacos e distâncias entre S1 e S2.

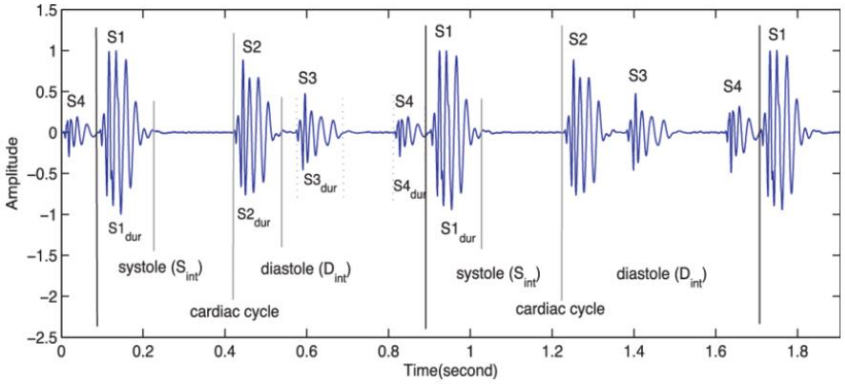


Figura 10 - Distâncias entre ciclos cardíacos S1 e S2 (Varghees and Ramachandran, 2014)

Depois de os atributos serem extraídos, foram utilizados dois métodos de classificação: J48 e *Multi-Layer Perceptron* (*MLP*) recorrendo ao *Weka* (Azuaje, 2006).

4.2.2 *Classifying Heart Sounds Using Images of Motifs, MFCC and Temporal Features*

O objetivo do trabalho descrito neste artigo foi desenvolver um modelo de classificação que utilizando os coeficientes *MFCC*, *Motifs* e *TF* para extração de atributos, fosse capaz de diferenciar sons normais de anormais, utilizando o conjunto de dados do *Physionet/ Computing in Cardiology Challenge* (Chengyu Liu, 2016).

A metodologia descrita neste trabalho começa pela segmentação dos sons cardíacos utilizando o algoritmo *Springer's segmentation* (Springer, Tarassenko and Clifford, 2015). Este algoritmo utiliza como referência os sinais ECG de modo a prever a duração esperada de cada estado do som cardíaco. Desta forma, é possível identificar cada um dos seus quatro estados fundamentais, S1, sístole, S2 e diástole no fonocardiograma. Essa segmentação dos estados possibilita a divisão do som em segmentos de 3 segundos, tendo como ponto inicial o estado S1. Após a divisão dos sons em segmentos, foram considerados três técnicas para a extração dos seus atributos. As *TF* foram extraídas através das distâncias entre S1 e S2 (médias, medianas e desvios padrão) e os atributos de frequências utilizando *MFCC* e *Motifs*. De seguida, para cada segmento, foi criado um *heatmap* que resulta da junção de *MFCC* e *TF* e outro da junção de *Motifs* com *TF*, formando grupos distintos. Na Figura 11 ilustra um exemplo, de um PCG de três segundos, em que uma linha vermelha identifica os quatro estados fundamentais e o seu *MFCC heatmap* correspondente.

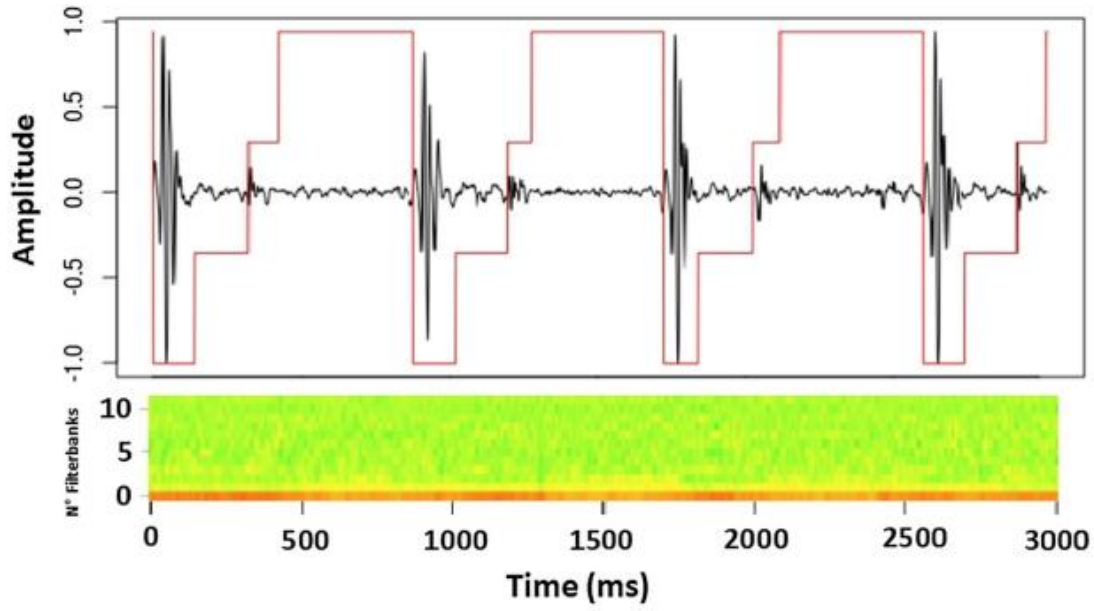


Figura 11 - PCG e *MFCC* com 3 segundos de duração (Nogueira *et al.*, 2019)

Posteriormente foram avaliados algoritmos como *SVM* e *RF*, usando diversas combinações de atributos, quanto à capacidade de discriminar batimentos cardíacos normais de anormais. Através da utilização desses algoritmos, foi possível determinar a que classes os segmentos de 3 segundos correspondem. De seguida, as previsões dos segmentos são agrupadas por paciente onde é classificado o som original. Esta classificação é feita considerando a percentagem de vezes que este é considerado como normal. Se essa percentagem estiver acima de um pré-definido *threshold,* é considerada como normal, senão anormal. Para a definição do *threshold,* foram criados oito grupos tendo em conta a frequência cardíaca. Na Figura 12, são apresentados os vários conjuntos de dados criados para testes presentes na coluna "*Type of features"*, os modelos utilizados na coluna "*Model"* e os grupos com os respetivos *threshold*.

Type of features	Model	G1	G2	G3	G4	G5	G6	G7	G8
5 MFCC + TF	SVM	0.3	0.3	0.25	0.17	0.3	0.21	0.4	0.21
6 MFCC + TF	SVM	0.3	0.3	0.37	0.25	0.24	0.21	0.24	0.2
5 MFCC + TF	RF	0.3	0.3	0.17	0.23	0.16	0.16	0.14	0.22
6 MFCC + TF	RF	0.3	0.3	0.13	0.18	0.13	0.22	0.16	0.18
Motifs (20-20) + TF	SVM	0.28	0.32	0.31	0.17	0.14	0.19	0.24	0.17
Motifs (40-40) + TF	SVM	0.36	0.32	0.21	0.17	0.16	0.18	0.23	0.17
Motifs (20-20) + TF	RF	0.3	0.3	0.24	0.14	0.13	0.16	0.32	0.18
Motifs (40-40) + TF	RF	0.3	0.3	0.19	0.15	0.15	0.23	0.29	0.18
TF	SVM	0.3	0.3	0.38	0.2	0.24	0.26	0.15	0.17
TF	RF	0.3	0.3	0.28	0.2	0.22	0.18	0.29	0.2

Figura 12 - Grupos, *thresholds* e respetivos modelos utilizados para cada conjunto de dados (Nogueira *et al.*, 2019)

O melhor resultado foi alcançado pelo *SVM* que utilizou como treino (combinação 5 *MFCC*+*TF*), e obteve um *overall* de 83.22%. Este resultado é muito satisfatório uma vez que se posiciona

dentro dos Top 10 dos melhores resultados do concurso e *PhysioNet Computing in Cardiology Challenge* (Liu *et al.*, 2016).

4.3 Abordagens existentes

Nos últimos anos, a análise de sinais dos sons cardíacos têm demonstrado potencial para a deteção de DCV, o que levou às comunidades de pesquisa a propor e projetar vários artigos científicos nesta área (Liu *et al.*, 2016).

Na Tabela 2 são apresentados exemplos de contributos na área deteção de patologias cardíacas, em que é descrito o conjunto de dados, a metodologia desenvolvida e o seu melhor resultado.

Tabela 2 - Exemplos de estudos na área de classificação de sons cardíacos

Autor	**Conjunto de dados**	**Descrição**	**Melhores Resultados e medidas de avaliação usadas**
(DeGroff *et al.*, 2001)	Foram recolhidos 69 sons cardíacos utilizando um estetoscópio eletrónico.	O método proposto inicialmente aplica o *Fast Fourier Transform* (*FFT*) que transforma o som em espectro de energia normalizados. Foram criados diferentes conjuntos de treino para intervalos de frequências (0 até 90, 0 até 150, 0 até 210, 0 até 255 e 0 até 300 Hz) e *spectral* resolutions (1, 3, e 5 Hz). O modelo utilizado foi *artificial neural network* (*ANN*) e dado a pequena dimensão dos dados, foi utilizado método *Jack-Knifing* para a sua avaliação. O método *Jack-Knifing* é uma técnica em que num conjunto de amostras N são treinados N-1 e a amostra restante serve como ponto de validação.	***ANN*:** SENS:100% SPEC:100%

(Ahlstrom *et al.*, 2006)	Um total de 36 pacientes (19 homens e 17 mulheres) com idades entre os 14 e 69 anos. Foi diagnosticado 6 pacientes com insuficiência mitral, 23 pacientes com estenose aórtica e 7 pacientes com sopros.	Utilizou *Recurrent Neural Network* (*RNN*) com o objetivo de determinar a diferença entre as três patologias. Para extrair os dados através do ECG e contruir o conjunto de treino utilizou *Wavelet Transform* (*WT*), *Shannon Energy*, *Fractal Dimension* (*FD*), *Recurrence Quantification Analysis* (*RQA*).	***Neural network**:* ACC:86%
(Noponen *et al.*, 2007)	Os dados foram recolhidos entre 1995-1999 com um total 807 crianças com idade entre o 1-16 anos. Foram recolhidas mais de 12 *murmur* diferentes.	Os sons foram inicialmente filtrados numa frequência de 75 Hz – 1500 Hz. De seguida foram projetados numa representação visual, o espectrograma dos sons recolhidos em conjunto com o fonocardiograma. O objetivo consistia em diferenciar *mumur* inocentes de patológicos.	**Método específico:** SENS:90% SPEC:91%
(Gavrovska *et al.*, 2013)	Os dados foram recolhidos na Health Center "Zvezdara", Belgrado, Serbia, usando um estetoscópio digital. Foram gravados 117 sons de 117 crianças de idades entre os 7 e os 19 anos em que 54 eram masculinas e 63 femininas. Os dados foram divididos em dois grupos, normal e P*rolapsed Mitral Valve* (PMV). Todos os dados foram validados por um cardiologista experiente.	O método proposto inicia-se com a conversão de batimentos cardíacos em forma de PCG em pontos ou curvas no espaço(x,y) utilizando *multifractal dimensions*. As *Multifractal dimensions* foram introduzidas por Mandelbrot em 1980 com o propósito de medir a velocidade do fluxo cardíaco. Posteriormente são aplicados dois algoritmos: o primeiro divide em dois conjuntos de dados utilizando como critério a área. O Segundo divide em duas classes (normal e PMV) com base da inclinação da curva para ambos os conjuntos (obtidos de acordo com valores de área na etapa anterior).	**ACC**:96.9%

(Singh and Cheema, 2013)	O conjunto de dados é composto por 60 sons em que 30 sons correspondem a patologia normal e outros 30 sons a *murmur*. Os dados foram através de um estetoscópio digital.	O método proposto inicia com a extração de 23 atributos presentes no som. Esses atributos são avaliados e de seguida é determinado que apenas 5 dos 23 eram suficientes para diferenciar uma patologia normal de um *murmur*. De seguida, *Naive Bayes*, *Bayes Net*, *Logit Boost* e *Stochastic Gradient Descent* (*SGD*) foram aplicados para a criação dos modelos de classificação.	***Naive bayes*:** ACC:93.3% SENS: 93.3% SPEC: 93.3%
(Potes *et al.*, 2016)	*PhysioNet Computing* 2016	Apresentam um método capaz de diferenciar batimentos normais de anormais. O método inicia com a aplicação de um filtro passa-banda que filtra o som de cada ficheiro numa gama de 25Hz a 400Hz. De seguida, é aplicada a segmentação identificando as posições dos batimentos S1 e S2 no som. Depois de terminado o pré-processamento e a segmentação, foram extraídos 36 atributos utilizando as posições dos batimentos obtidos no processo de segmentação e outros 13 atributos utilizando o *MFCC*. Esses dados foram combinados com o objetivo de criar um conjunto de treino para os modelos. Os modelos utilizados foram *AdaBoost-abstain*, *CNN* e a sua combinação (*ensemble classifier*).	***Ensemble classifier*:** ACC:86% SENS:94.2% SPEC:77.8%

(Banerjee and Majhi, 2020)	PASCAL *challenge dataset*.	O método inicia com o pré-processamento do som. Neste passo, o som é *downsample* e filtrado num intervalo de 30Hz até 900Hz e de seguida normalizado entre -1 e 1. No passo de segmentação é obtido o máximo absoluto dos sinais previamente normalizados. Na extração de atributos, o método *MFCC* é utilizado para extrair a informação dos sinais e essa informação é utlizada para treinar um modelo de classificação *CNN*.	***CNN*:** ACC:83%
(Khan *et al.*, 2020)	PASCAL *challenge dataset* e *PhysioNet Computing in Cardiology Challeng.*	A metodologia proposta inicia com o pré-processamento, em que, os sinais sonoros são *resampled* numa frequência de 2000Hz, é aplicado um filtro de 20Hz até 400Hz e de seguida, esses sinais são divididos em segmentos de 8 segundos. No próximo passo, os atributos dos sinais são convertidos em espectogramas utilizando o *Short-time Fourier Transform* (*STFT*). Os espectrogramas extraídos são usados para treinar um modelo *CNN*. Para avaliar esse modelo foi usado *cross-validation* com *K-fold* (*K*=10).	***CNN*:** ACC:96.8% SENS: 95.8% SPEC: 98%
(Bourouh ou *et al.*, 2020)	PASCAL *challenge dataset*.	O método proposto inicia com o *resample* do som para 1000Hz. De seguida, o som é filtrado num intervalo de 25 HZ até 400Hz e normalizado. A extração da informação do som foi realizada recorrendo ao método proposto por Springer, Tarassenko and Clifford (2015).	***Naive bayes*:** ACC:80 %
(Mustafa *et al.*, 2020)	PASCAL *challenge dataset*.	O som é filtrado num intervalo de 50Hz a 1000Hz. De seguida, os batimentos S1 e S2 são separados utilizando um *treshold* específico. Os segmentos dos sons são definidos dinamicamente de acordo com a duração do S1 e S2 definidos no passo anterior. Os atributos dos segmentos são	***MLP*:** ACC:90% SENS: 91.1% SPEC: 90.3%

		extraídos recorrendo usando *MFCC*.	
(Latif *et al.*, 2018)	*PhysioNet Computing* 2016	Inicialmente, foi utilizado o método *Logistic Regression-Hidden Semi-Markov Models* (HSMM) proposto por (Springer, Tarassenko and Clifford, 2015) para a deteção dos ciclos cardíacos o que permitiu dividir os sons em segmentos com 5 ciclos cardíacos. De seguida, as foram extraídos os atributos desses segmentos usando *MFCC*. Por fim, foram criados modelos de *RNN*, *Gated Recurrent Unit* (*GRU*), Bidirectional Long *Short-Term Memory* (*BLSTM*) e *Long Short-Term Memory* (*LSTM*).	***RNN***: ACC:97.6% SENS: 98.3% SPEC: 98.8%
(Chowdhury, Poudel and Hu, 2020)	*PhysioNet Computing in Cardiology Challenge* 2016	A metodologia proposta inicia pelo filtro dos sinais sonoros numa frequência de 0-250Hz. De seguida, a segmentação dos dados é realizada recorrendo ao *Shannon Energy Envelop* e *Zero-Crossing Algorithm*. A extração da informação dos sinais foi realizada usando *MFCC*. O modelo utilizado foi *Feedforward Neuronal Network* (*FFN*) com 5 *layers*.	***FFN*:** ACC:97.1% SENS: 99.2% SPEC: 94.8%
(SINGH and MAJUMDER, 2020)-	*PhysioNet Computing in Cardiology Challenge* 2016 e PASCAL *challenge dataset*.	Na fase de pré-processamento, os dados são divididos em segmentos de 5 segundos, filtrados num intervalo de 25Hz a 400Hz e normalizados. Para a criação dos modelos de classificação foram utilizados atributos temporais, atributos de frequências, atributos estatísticos e entropia. Para treinar o modelo foi utilizado o modelo 5-*fold cross-validation.*	***AdaBoost*:** ACC:92.4% SENS: 94 % SPEC: 91.9%

(Boulares, Alafif and Barnawi, 2020)	PASCAL *challenge dataset*.	Neste trabalho, foram utilizados de 18 modelos pré-treinados, em que, unicamente foram preservadas as camadas de extração de atributos. Depois foram adicionadas quatro camadas extra para uma melhor representação da extração dos atributos. De seguida, foi utilizado *MFCC* para converter os sinais sonoros em imagem. Este estudo tem modelos para a classificação binária e *multiclass.*	***VGG19*:** ACC:77% SENS: 76 %
(Koike *et al.*, 2020)	*PhysioNet Computing in Cardiology Challenge* 2016	Neste trabalho foi utilizado *Mel Spectrogram* e *MFCC* para a extração de atributos e conversão em imagem dos sons. Foi aplicado *transfer learning* através de um modelo pré-treinado em som chamado *Propose Pretrained Audio Neural Networks* (*PANNs*).	***PANNs*:** Spec:88.6% SENS:96.9%
(Oliveira and Praca, 2021)	MELD	Este trabalho tem como principal objetivo a deteção de emoções em amostras de áudio. Para a extração de atributos foram utilizadas duas abordagens diferentes. A primeira abordagem consiste na utilização do modelo treinado em reconhecimento de fala chamado *Baidu's Deep Speech* que é treinado utilizando *Spectrograms*. A segunda abordagem consiste na extração de atributos utilizando a ferramenta *openSMILE* 2.3.0 para o treino de modelos de *machine learning*.	***TrBaidu*:** F1-*score* (*Joy* e *surprise)*: 23%

∗ACC: *Accuracy*, SENS: *Sensitivity*, SPEC: *Specificity*.

4.3.1 Conclusão

A partir da análise dos vários trabalhos, foi possível determinar que as metodologias apresentam passos semelhantes. De modo geral, o primeiro passo é o pré-processamento, seguido da segmentação, extração da informação e criação dos modelos *Machine Learning* (*ML*). Por último, os modelos previamente criados, são testados usando uma metodologia de avaliação.

No passo de pré-processamento, todos os sons são tratados de forma a tornar os batimentos cardíacos mais audíveis e consistentes entre si. Para isso ser possível, o som é feito *resample* para uma taxa de amostragem específica, o ruído é removido, normalizado e dividido em segmentos do mesmo tamanho.

No passo seguinte, o processo de segmentação dos dados consiste na identificação dos batimentos S1 e S2 no som. Este passo torna-se relevante para extrair atributos temporais do ritmo cardíaco, que posteriormente são utilizados para treinar os modelos de classificação. Nos conjuntos de dados PASCAL e *PhysioNet* é utilizado o envelope de energia *Shannon* na segmentação para deteção dos picos S1 e S2 (Chowdhury, Poudel and Hu, 2020).

O passo seguinte consiste na extração de atributos do som, através do método amplamente utilizado em trabalhos de *deep learning*, o *MFCC*. Este permite a extração da informação do som através de uma imagem ou de um conjunto pequeno de recursos que descrevem concisamente o sinal sonoro.

Na criação dos modelos são utilizados algoritmos de *deep learning* ou *machine learning* de acordo com a quantidade de dados presentes no estudo. A utilização de algoritmos de *deep learning*, surgiu maioritariamente a partir de 2016, com o aparecimento do conjunto de dados *PhysioNet* (Liu *et al.*, 2016). A partir desse momento, começaram a aparecer diversos estudos com diversos modelos e arquitetura de *deep learning.*

Para o desenvolvimento do projeto, foram escolhidas as arquiteturas dos trabalhos de Khan *et al.* (2020) e o de Boulares, Alafif and Barnawi (2020). A escolha de Khan *et al.* (2020) deve-se aos resultados alcançados na utilização de ambos os conjuntos de dados (PASCAL e *PhysioNet*), uma vez que, poucos estudos conseguem alcançar resultados semelhantes devido ao facto de os sons gravados serem provenientes de dispositivos diferentes. O trabalho de Boulares, Alafif and Barnawi (2020), foi selecionado pela utilização da técnica *fine-tunning* de modelos pré-treinados, que difere do *transfer learning* na medida em que não só são alteradas as camadas de saída, mas também são retreinadas as camadas correspondentes à extração de atributos. A identificação das camadas a serem retreinadas dependem da quantidade dos dados do novo conjunto e da sua diferença do original.

Para avaliação dos modelos desenvolvidos, é comum encontrar na bibliografia a metodologia de avaliação *K-fold cross-validation* e as métricas *accuracy*, *sensitivity* e *specificity* são frequentemente utilizadas (Nogueira *et al.*, 2019; Khan *et al.*, 2020; Mustafa *et al.*, 2020; SINGH and MAJUMDER, 2020).

4.4 Processamento do som

4.4.1 *Fast Fourier Transform* (*FFT*)

O FFT é um algoritmo que permite aplicar de forma eficiente a transformada de *Fourier*, através da decomposição do sinal do som em frequências individuais e amplitudes da frequência, ou seja, converte o sinal do domínio temporal para domínio da frequência (Figura 13). Este algoritmo é muito utilizado no processamento de sinais sonoros (Roberts, 2020).

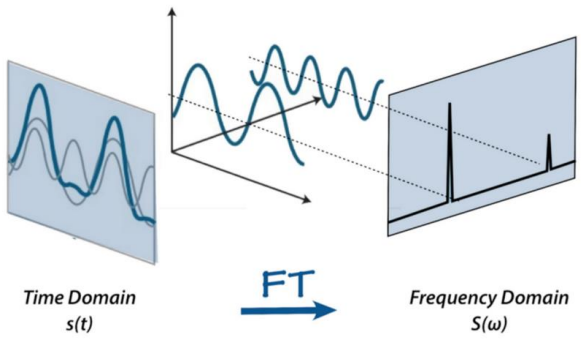


Figura 13 - Exemplo de aplicação de FFT (Roberts, 2020)

4.4.2 *Mel Spectrogram*

Um *spectrogram* representa visualmente a mudança da frequência de um sinal em relação ao tempo. Podemos ver o *spectrogram* com um conjunto de *FFTs* empilhados uns sobre os outros (Roberts, 2020).

O *mel-scale* proposto por Stevens, Volkmann e Newmann em 1937, visa a imitação da perceção não linear do som no ouvido humano, sendo mais distinta nas frequências mais baixas e menos distinta nas frequências mais altas (Chowdhury, Poudel and Hu, 2020).

O *Mel Spectrogram* é um *spectrogram* onde as frequências são convertidas para *mel-scale* (Roberts, 2020). A Figura 14 apresenta um exemplo de um *Mel Spectrogram*.

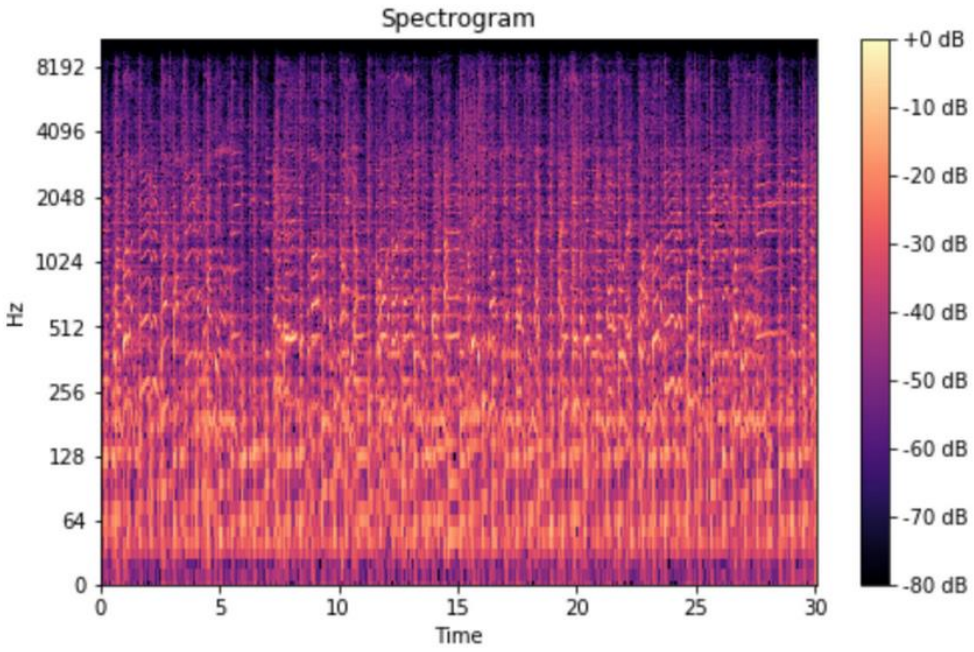


Figura 14 - Exemplo de um *Mel Spectrogram* (Roberts, 2020)

4.4.3 *Mel Frequency Cepstral Coefficients* (*MFCC)*

Os *MFCCs* foram introduzidos por Davis e Mermelstein, na década de 1980, tendo sido, desde então um dos recursos mais utilizados no reconhecimento automático de sons (Jameslyons, 2013). O *MFCC* é um algoritmo de processamento de sinal poderoso, pois possibilita a extração de atributos da frequência do som inspirado no comportamento do ouvido do ser humano (Nogueira, Ferreira and Jorge, 2017).

A criação do *MFCC* envolve a análise e o processamento do som de acordo com os passos descritos na Figura 15.

Figura 15 - Passos para a criação do *MFCC* (Nogueira, Ferreira and Jorge, 2017)

O algoritmo recebe o som e inicia a fase de *pre-emphasis* que tem a função de aprimorar os sinais e compensar as distorções do sinal. De seguida, o som é dividido em pequenas *frames* e é aplicado o *Fast Fourier Transform* a cada *frame*. O *Mel Filterbank* faz com que a operação seja idêntica à perceção humana e converte as composições da frequência do *Mel-filter* em intensidade de energia. Seguidamente, estes valores são normalizados pela fase do logaritmo (*Log*) que torna os dados sobrepostos e correlacionados. Para descorrelacionar os dados é aplicada a transformação *discrete cosine transform* (*DCT*), antes de ser criado o *MFCC* (Jameslyons, 2013; Nogueira, Ferreira and Jorge, 2017).

4.5 Algoritmos de Classificação

A Inteligência Artificial (IA) tem tido um crescimento acentuado nos últimos anos, estando, atualmente, presente em praticamente tudo o que utilizamos (Michael Copeland, 2016). A IA é uma ciência que estuda formas de construir programas ou máquinas inteligentes que consigam imitar as capacidades humanas (IBM Cloud Education, 2020a). O *ML*, subconjunto do IA, fornece um conjunto de sistemas que têm a capacidade de aprender automaticamente e melhorar ao longo do tempo. Estes algoritmos foram extensamente utilizados na classificação automática de doenças cardíacas (Oh *et al.*, 2020).

No trabalho descrito neste documento foram estudados algoritmos de *ML*, nomeadamente, de *deep leaning.* Estes algoritmos diferem do *ML* na forma como os dados são trabalhados e aprendidos. Os algoritmos clássicos de *ML* que aprendem com dados estruturados e devidamente identificados, necessitam de intervenção humana e são conhecidos como aprendizagem supervisionada. Por sua vez, algoritmos de *ML* também conseguem detetar características ou padrões em dados que não estão devidamente etiquetados, com o mínimo de supervisão humana, conhecidos como aprendizagem não supervisionada (Jason Brownlee, 2019; IBM Cloud Education, 2020b).

4.5.1 Funcionamento das Redes Neuronais Artificiais (*ANN*)

As redes neuronais foram inspiradas no funcionamento do cérebro humano em que copiam a forma com que os neurónios biológicos comunicam entre si. Estas redes são formadas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó ou neurónio artificial comunica com outro tendo um peso e um limite associado. Enquanto que o peso é a representação da conexão entre os neurónios, no qual indica a importância do *input*, o limite indica se a informação recebida pelo neurónio é relevante para a previsão do modelo. Esta relevância é determinada pela função de ativação que se encontra anexada em cada neurónio. Deste modo, a função de ativação consiste em equações matemáticas que determinam o *output* do neurónio. Se o resultado da função de ativação for superior ao limite do neurónio, a informação é relevante e procede para a próxima camada (função ativada). Assim sendo, existem várias funções de ativação, sendo uma das mais comuns a função Sigmoid e *Rectified Linear Unit* (*ReLU*) (Eda Kavlakoglu, 2020; MissingLink, 2020). A

função *Sigmoid* representada na Figura 16 do lado esquerdo, transforma os valores de entrada em valores entre 0 e 1. Desta forma todos os valores superiores a 1 são transformados em 1, da mesma forma que os menores que 0 são transformados em 0. Por sua vez, a função *ReLU* representada na Figura 16 do lado direito, transforma todos os valores de entrada menores que 0 em 0 e todos os valores de entrada maiores que 0 são mantidos iguais.

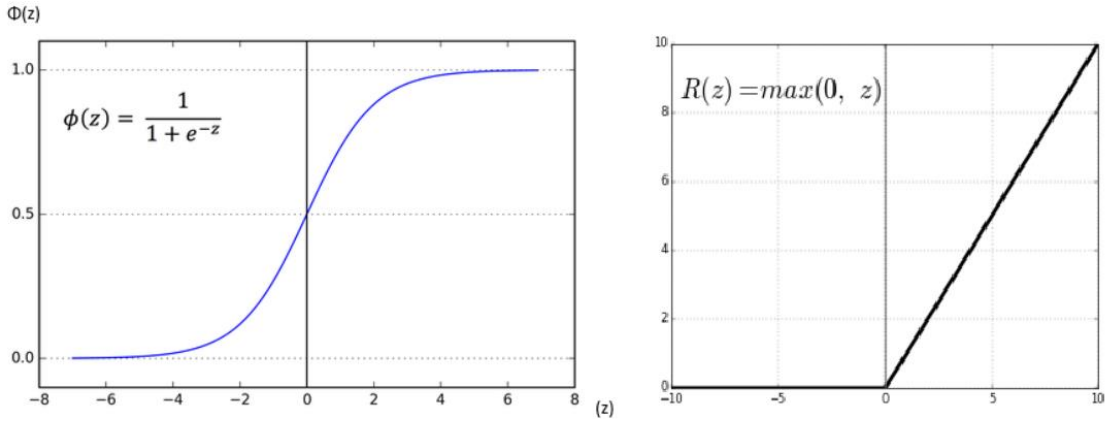


Figura 16 - Representação da função *Sigmoid* e da função *ReLU*

4.5.2 Redes Neurais Recorrente (*RNN*)

As *RNN* são redes neuronais em que, a cada etapa, os neurónios lembram-se das informações treinadas nas etapas anteriores. Desta forma, para além da influência dos pesos, contêm um vector "oculto" que representa o *context* com base nos inputs anteriores. Assim, a mesma entrada pode produzir saídas diferentes, dependendo das entradas anteriores (Venkatachalam, 2019).

4.5.3 *Long Short-Term Memory* (*LSTM*)

É um tipo especial de *RNN* na qual cada neurónio contém um bloco de memória para armazenar representações em intervalos de tempo. Um bloco de memória consiste em três portas: porta de entrada, saída e esquecer. Essas portas aprendem a controlar o fluxo de erro constante dentro de cada neurónio. Desta forma, o neurónio decide o que armazenar e quando permitir leituras, gravações ou apagar informações (Latif *et al.*, 2018). Na Figura 17 esta representado um exemplo da aplicação de *LTSM*, em que a dada altura, ao analisar uma determinada palavra e consegue prever a palavra mais provável, pois tem em consideração os elementos anteriormente processados.

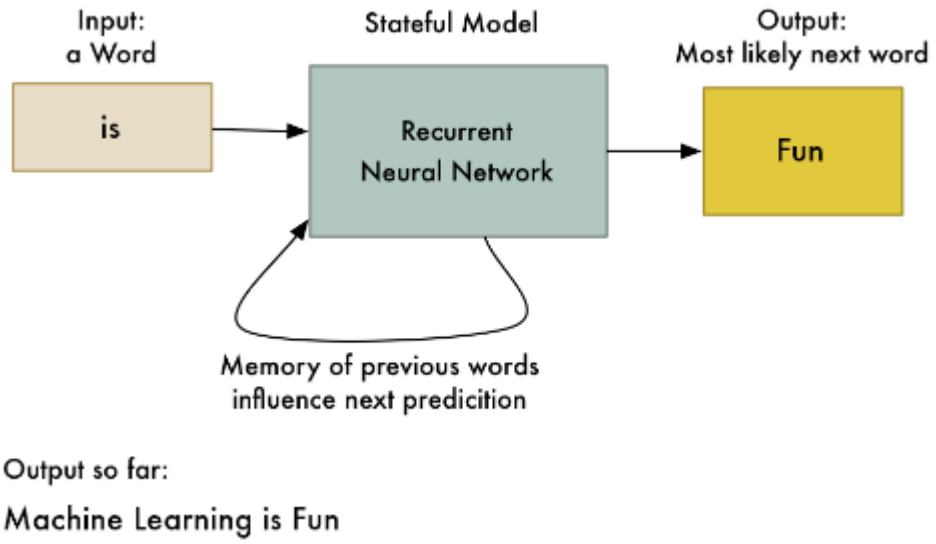


Figura 17 - Exemplo da aplicação do *LTSM* (Kishan Maladkar, 2018)

4.5.4 *Convolutional Neural Network* (*CNN*)

A *CNN* é um algoritmo de *deep learning* utilizado no processamento de sinais de imagens. Esta rede neuronal recebe como entrada uma imagem que é processada por uma série de camadas de *convolution layers, pooling layers*, *fully connected layers*. Por fim, é aplicada uma função *Sigmoid* para a classificação de um objeto com valores entre 0 e 1.

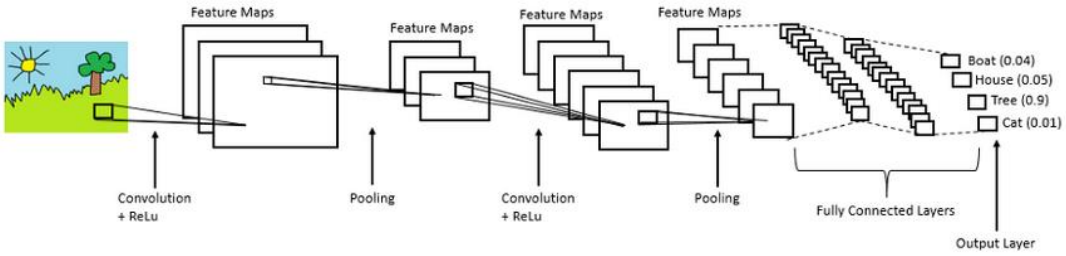


Figura 18 - Funcionamento de uma *CNN* (Prabhu, 2018)

Como é possível ver na Figura 18, a *convolution* é a primeira camada do algoritmo que realiza a extração de recursos da imagem recebida como input. A *convolution* preserva a relação entre os pixels, aprendendo os recursos da imagem usando pequenos quadrados de dados como entrada. De seguida, é aplicado o *pooling* que tem como objetivo reduzir a matriz (representação da imagem) mas retém a informação importante. A última camada corresponde às *fully connected layers,* que realizam operações lineares para aprender a relação entre os valores de entrada e saída. Por fim, a função de ativação (*Sigmoid* ou *Softmax*) permite obter as probabilidades dos valores de saída pertencente a cada classe diferente (Prabhu, 2018). Para prevenir o *overfitting* do modelo, ou seja, para prevenir que o modelo fique demasiado ajustado ao conjunto de treino e desta forma não consiga generalizar para novos dados, é comum a

utilização da camada *dropout.* Esta camada anula aleatoriamente a contribuição de alguns neurónios para a próxima camada e não altera todas as outras (Baeldung, 2020).

4.6 Avaliação

A avaliação do modelo tem como objetivo estimar o desempenho dos modelos quando são usados para fazer previsão sobre dados desconhecidos. Esta informação permite avaliar e comparar os modelos desenvolvidos, obtendo informação sobre a viabilidade dos mesmos.

4.6.1 *K-fold cross validation*

Esta metodologia de avaliação, divide sistematicamente os dados em *K-folds*, treina o modelo com *K-1 folds*, e o *fold* que resta é usado para teste. Este processo é repetido *K* vezes, garantindo que existem *K* modelos diferentes e *K* previsões com precisões diferentes. A precisão final do modelo é obtida pela média de todas as previsões, tornando o desempenho do modelo mais próximo do desempenho real (Brownlee, 2020a). A Figura 19, representa um exemplo da aplicação da técnica descrita.

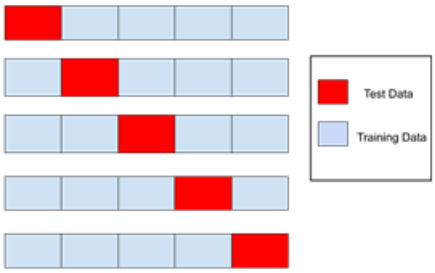


Figura 19 - Separação em 5 *folds* (Mltut, 2020)

4.6.1.1 *Stratified K-fold cross-validation*

Este método é uma variação do *cross-validation*, sendo que a diferença consiste na possibilidade ajustar o *fold* de acordo com a proporção do valor categórico de cada classe no conjunto de treino total. Assim, é garantido que cada *fold* tem a mesma distribuição do conjunto de treino completo (Brownlee, 2020b).

4.6.2 *Hold-out*

A metodologia *hold-out* consiste na divisão do conjunto de dados num conjunto de treino e outro de teste (por exemplo 70%, 30%). O conjunto de treino consiste nos dados com que o modelo é treinado e o conjunto de teste são os dados desconhecidos para o modelo, com o qual este será testado.

4.6.3 Matriz de confusão

A matriz de confusão é uma tabela frequentemente utilizada para descrever o desempenho de um modelo de classificação em um conjunto de testes para quais os valores verdadeiros são conhecidos.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figura 20 - Matriz de confusão (Team, 2020)

Como é possível ver na Figura 20 (para o caso binário), existem duas possíveis classes: a classe positiva e a classe negativa e um total de 165 previsões. A classe positiva representa a classe de interesse e a negativa representa todo o resto que não faz parte da classe positiva (não tem interesse). Neste caso, a classe considerada como positiva é o "*YES"* e a negativa é o "*NO*". A previsão de duas classes é conhecida como classificação binária.

A previsão do modelo representada pela matriz de confusão (Figura 20), demonstra através da soma das colunas que houve 110 dados classificados como "*YES"* e 55 como "*NO*". Os dados reais são dados pela soma das linhas, 105 como "*YES"* e 60 como "*NO".* Através da leitura da matriz ainda é possível obter outos dados tais como:

- ***True positives* (*TP*):** Representa a quantidade de previsões de uma classe que foram classificadas como positivas, e são positivas.
- ***True negatives* (*TN*)**: Representa a quantidade de previsões de uma classe que foram classificadas como negativas, e são negativas.
- ***False positives* (*FP*):** Representa a quantidade de previsões de uma classe que foram classificadas como positivas, mas na realidade são negativas.
- ***False negatives* (*FN*):** Representa a quantidade de previsões de uma classe que foram classificadas como negativas, mas na realidade são positivas.

4.6.4 Métricas de avaliação

4.6.4.1 *Accuracy*

A *accuracy* é considerada a medida de desempenho mais intuitiva. Esta medida é obtida pela proporção entre as observações previstas corretamente e o total de observações. A fórmula da *accuracy* é a seguinte:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

4.6.4.2 *Precision*

A *precision* é uma proporção entre as observações de uma classe que são corretamente previstas por um modelo em relação ao total de observações positivas previstas por esse modelo. A fórmula da precisão é a seguinte:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

4.6.4.3 *Recall*

A *recall* é conhecida também como *sensitivity* e, corresponde à proporção de observações previstas corretamente pelo modelo sobre todas as observações classificadas incorretamente como outra classe.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4.6.4.4 *F1 score*

O *F1 score* é a média ponderada da *precision* e *recall*.

$$F1\ Score = 2 * \frac{(Recall * Precision)}{Recall + Precision} \quad (4)$$

4.6.4.5 *Overall*

A métrica *overall* é a média entre o *recall* e *precision*. Esta métrica foi utilizada para a comparação dos resultados no concurso *PhysioNet* 2016.

$$Overall = \frac{Recall + Precision}{2} \quad (5)$$

4.6.4.6 Curva *ROC* e *AUC*

A *Receiver Operator Characteristic* (*ROC*) é uma métrica de avaliação para problemas binários. Esta métrica consiste numa curva que representa a *True Positive Rate* (*TPR*) contra o *False Positive Rate* (*FPR*) para os vários *thresholds*. A *TPR* é também conhecida como *recall*. O *FPR* é calculado como:

$$FPR = \frac{FP}{FP + TN} \quad (6)$$

A *Area Under the Curve* (*AUC*) é a capacidade de um classificador de distinguir entre as classes e é usada com um resumo da curva *ROC*.

4.7 Modelos pré-treinados

As *CNN* têm alcançado bons resultados no campo da visão computacional nos últimos anos. No entanto, para ser possível atingir esses resultados, é necessário uma grande quantidade e variedade de dados para treinar os modelos, o que resulta em altos custos de computação (Koike *et al.*, 2020). Assim sendo, devido à falta de volume de dados, uma possível solução é a utilização de *transfer learning*. O *transfer learning* permite que modelos treinados num dado problema sejam usados num segundo problema relacionado. Desta forma, é possível utilizar modelos pré-treinados em grande volume de dados para casos específicos.

4.7.1 *AlexNet*

O *ImageNet Large Scale Visual Recognition Challenge* (*ILSVRC*) foi uma competição realizada entre 2011 e 2016, com o objetivo de estimular a inovação no campo da visão computacional (Russakovsky *et al.*, 2015). Neste evento, o modelo *AlexNet* criado por Alex Krizhevsky, demonstrou uma evolução nos modelos de *CNN* com a introdução de novas variáveis, *ReLU* e *dropout* na sua arquitetura. Estas variáveis foram amplamente utilizadas na altura e atualmente tornaram-se requisitos para a utilização de *CNN* na classificação de imagens. Na Figura 21 é apresentada a arquitetura do *AlexNet* (Brownlee, 2019).

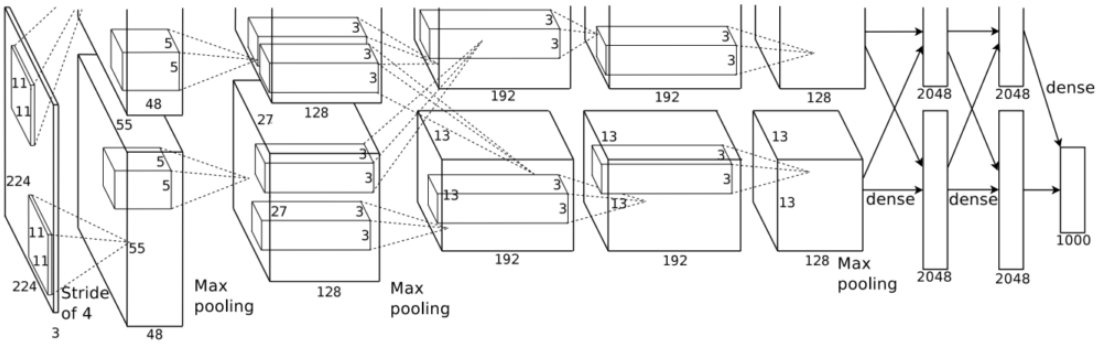


Figura 21 - Representação da arquitetura *AlexNet* (Brownlee, 2019)

A rede é composta por 5 camadas *convolution* para extração de recursos e três camadas *fully connected na* parte de classificadora (K.Raimi, 2019).

4.7.2 *VGG*

A arquitetura *Visual Geometry Group* (*VGG*) foi desenvolvida por Karen Simonyan e Andrew Zisserman da universidade de Oxford, em 2014. Estes desenvolveram duas versões amplamente conhecidas, a *VGG-16* e a *VGG-19.* A *VGG-16* conta com um total de 16 camadas, enquanto a *VGG-19* conta com um total de 19 camadas. A introdução de modelos mais profundos do que os conhecidos na altura, tinha com o objetivo introduzir um padrão de arquiteturas *CNN* mais profundas (Brownlee, 2019).

Esta arquitetura destaca-se pelo uso de um grande número de filtros, especialmente filtros 3x3. Usa também um grande número de filtros, sendo que, a primeira camada começa com 64 filtros e aumenta com a profundidade para 128, 256 e 512 filtros (Brownlee, 2019).

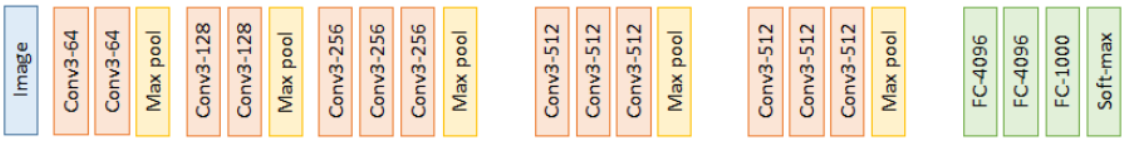


Figura 22 - Representação da arquitetura *VGG* (Basaveswara, 2019)

A arquitetura *VGG* é apresentada na Figura 22. Esta é formada por camadas *convolutions* com filtros 3x3, camadas de *max pooling* com filtros 2x2 e uma camada *fully connected* no final. O tamanho de imagem padrão neste modelo é de 224x224x3 (K.Raimi, 2019).

4.7.3 *MobileNet*

A *MobileNet* é uma rede neuronal, desenvolvida pela Google em 2017, com o objetivo de desenvolver uma arquitetura ideal para dispositivos moveis. Esta é caracterizada pelo seu pequeno modelo e pela sua alta capacidade de cálculos. A chave para isto ser possível é a

utilização de *depthwise separable convolutions* em vez das tradicionais operações *convolution* (Prabinnepal, 2020).

4.7.4 *ResNet*

O *RestNet* foi desenvolvido por Ioffe e Szegedy, em 2015, tratando-se de uma rede neuronal desenhada para permitir centenas ou milhares de camadas. Esta rede neuronal conta com um mecanismo que não permite a degradação do desempenho da rede devido ao elevado número de camadas (Keras, 2020).

4.7.5 *Baidu's Deep Speech*

O *Baidu' Deep Speech foi* desenvolvido em 2015, com o objetivo de converter ficheiros áudio de fala, em sequências de caracteres utilizando espectrogramas. A arquitetura da rede é composta por camadas *convolutional*, uma camada GRU e no momento de avaliação, utiliza uma heurística chamada *beam search,* que retorna uma lista com os caracteres mais prováveis (Wolfram, 2018). Este modelo é treinado com um total de 11,940 horas de Inglês e 9,400 horas de Mandarim (Amodei *et al.*, 2015).

4.8 *Frameworks de deep learning*

4.8.1 *Python*

O *Python* é uma linguagem de programação orientada a objetos de alto nível, criada por Guido Van Rossum, em 1991 (Python, 2021). Esta linguagem teve um grande crescimento nos últimos anos devido ao grande número de bibliotecas robustas e populares na área de IA. Segundo TIOBE, em dezembro de 2020, o *Python* é a terceira linguagem de programação mais popular do mundo, e segundo o PYPL (*PopularitY of Programming Language Index*), é classificada como sendo a linguagem que mais tem crescido nos últimos 5 anos (TIOBE, 2020; Pierre Carbonnelle, 2021).

4.8.2 *TensorFlow*

O *TensorFlow* é uma *framework* de *ML* desenvolvida pela Google *Brain Team*. Trata-se de uma biblioteca gratuita que utiliza a linguagem *Python* como API de *front-end* para a construção de modelos de alto desempenho em C++. Esta *framework* funciona nas principais plataformas de SO de 64 bits, como *Windows, macOS, Linux, JavaScript, iOS e Android*. É uma das bibliotecas mais populares sendo utilizada por empresas como Netflix, Intel e Uber (Kulkarni, 2021).

Com o recente lançamento do *TensorFlow* 2.0*, Keras* recebeu parte dos módulos do *TensorFlow* e construiu uma *API* simples e intuitiva para definir arquiteturas e treinar redes neuronais. Isto,

permite uma fácil e rápida construção de modelos, reduzir o número de *APIs* duplicadas e obsoletas e também permite a implantação robusta de modelos em qualquer plataforma (Keras, 2019).

4.8.3 *Keras*

O *Keras* é uma *API* de alto nível utilizada para o desenvolvimento modelos de *deep learning*. Esta fornece diversas *frameworks* de *backend* como é o exemplo da *API tf.keras* para o acesso à *framework Tensorflow*. As principais vantagens da utilização da *API Keras* são: a rapidez com que é possível criar modelos, a sua comunidade, qualidade de documentação e a execução de funções em paralelo.

5 Design

Neste capítulo é apresentado um resumo das ferramentas e algoritmos utilizados para o desenvolvimento deste projeto. É definida a linguagem de programação e *framework* de *deep learning* utilizadas, descrito o pré-processamento, extração de atributos, avaliação dos modelos, abordagens adotadas e implementação da solução.

5.1 Linguagem & *Deep Learning Framework*

A escolha da *framework* de *deep learning* é um processo importante para o desenvolvimento deste projeto. Por esse motivo, foi selecionado um conjunto de *frameworks* e de seguida utilizada a metodologia AHP descrita na Secção 3.1. O resultado da aplicação dessa metodologia foi *Tensorflow* com integração no *Keras*, sendo uma ferramenta bastante popular, rápida e no qual existe grande quantidade de informação disponível.

Para o desenvolvimento do trabalho descrito nesta dissertação foi utilizado um servidor *notebook* com o *TensorFlow* 2.4.1 alojado no *Paperspace* (Paperspace, 2021).

5.2 Pré-processamento

Os sons cardíacos gravados por dispositivos contêm, frequentemente, uma quantidade de ruído que dificulta a deteção dos batimentos cardíacos. Isso torna o pré-processamento do som do coração uma etapa essencial na análise automática das gravações dos batimentos cardíacos. O pré-processamento revela a estrutura fisiológica inerente do sinal do coração, detetando as anormalidades nas regiões significativas do sinal PCG (Latif *et al.*, 2018).

O pré-processamento definido neste trabalho, é baseado nas abordagens da Secção 4.3:

- O *resample* dos sons gravados. Este passo é importante dado que os dados são provenientes de dispositivos diferentes;
- Aplicar um filtro passa-banda com frequências de corte de 20 a 400 Hz (Potes *et al.*, 2016; Bourouhou *et al.*, 2020; Khan *et al.*, 2020);
- Dividir o som em segmentos do mesmo tamanho.

5.3 Extração de atributos

A extração de atributos é o passo responsável pela extração dos parâmetros dos sinais PCG. Nesta dissertação, são comparados três métodos mais utilizados, *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*.

O *MFCC* surge em muitos dos estudos para classificação automática de sons cardíacos e isto deve-se à capacidade de representar a amplitude do espetro do discurso numa forma compacta (Latif *et al.*, 2018; Boulares, Alafif and Barnawi, 2020), como foi já referido na Secção 4.4.3.

5.4 Avaliação dos modelos

Os conjuntos de dados (*dataset*) PASCAL e *PhysioNet* são utilizados para o treino dos modelos. O conjunto PASCAL contém um total de 200 sons do tipo normal e 122 do tipo anormal, enquanto, o conjunto de dados *PhysioNet* contém 2575 normal e 665 anormal. Devido ao desequilíbrio existente nos dados (desbalanceamento das classes) e à sua reduzida quantidade, durante o processo de treino e teste é garantida a estratificação dos dados utilizando a metodologia *Hold-out*.

A identificação da classe positiva e classe negativa é importante para determinar a principal métrica de avaliação a aplicar nos modelos desenvolvidos. Assim sendo, a classe positiva corresponde a "pacientes com DCV" e a classe negativa a "pacientes sem DCV". Nos conjuntos de dados, a classe positiva é representada pelos dados do tipo anormal e a classe negativa pelos dados do tipo normal. Através da definição das classes, foi então definida como a principal métrica para a avaliação dos modelos o *F1 score* e posteriormente o *overall,* descritos na Secção 4.6.

5.5 Abordagens

Dos trabalhos analisados para este projeto foram selecionados dois artigos e desenvolvidas duas abordagens diferentes. A primeira abordagem, consiste na aplicação *transfer learning* de modelos já treinados. Estes modelos, ao aplicar conhecimento já adquirido para um problema relacionado, são úteis para casos em que o conjunto de treino tem poucos exemplos, em comparação com os modelos de *deep learning* convencionais. A aplicação de *transfer learning* é adequada para o treino de sons de batimentos cardíacos, uma vez que existe uma quantidade muito pequena de sons publicados.

A segunda abordagem, consiste na seleção do estudo que melhor se adequa ao objetivo desta dissertação. O estudo escolhido foi o de Khan *et al.* (2020) que para além dos resultados relevantes que obteve, baseou-se no conjunto de dados PASCAL, conjunto que apresenta bastante ruído que o torna mais difícil de tratar comparativamente com o *PhysioNet*.

5.5.1 Primeira Abordagem

A primeira abordagem segue a técnica *CNN fine-tuning* aplicada no projeto Boulares, Alafif and Barnawi (2020). Nesta abordagem, são retreinadas as primeiras camadas dos modelos pré-treinados responsáveis pela extração dos atributos genéricos das imagens. Isto porque, as primeiras camadas detetam os padrões gerais das imagens e, quanto mais se avança na rede, mais específicos são os padrões detetados para o conjunto de dados. Depois dessa informação genérica ser retreinada, são adicionadas novas camadas para aprenderem a complexidade dos sinais dos batimentos cardíacos. A Figura 23 apresenta a arquitetura da abordagem.

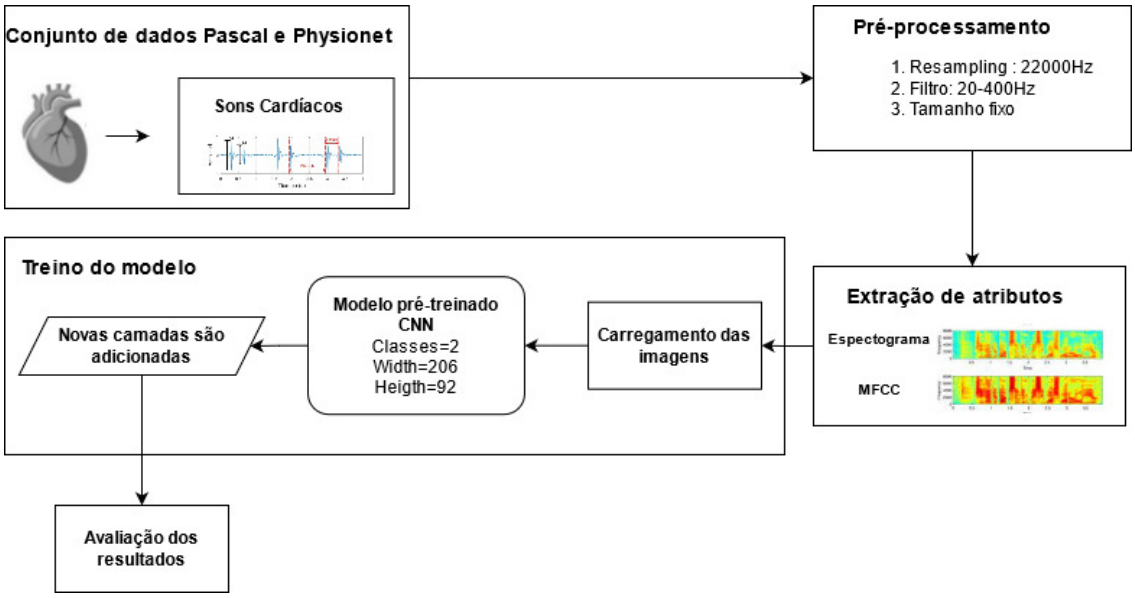


Figura 23 – Primeira Abordagem

Neste artigo não é realizado o pré-processamento dos dados, no entanto, esse é um passo importante, principalmente, para *resample* do som e remoção do ruído. Assim sendo, inicialmente é aplicado o pré-processamento descrito na Secção 5.2. De seguida, no processo de extração de atributos, os sinais sonoros são convertidos em espectrogramas utilizando os métodos descritos na Secção 5.3. Após a criação dos espectrogramas, o modelo pré-treinado é alterado (é removida a camada de classificação) e são adicionadas quatro novas camadas. Por último, os modelos são treinados, testados e avaliados de acordo com a Secção 5.4.

Na Tabela 3 são demonstradas as quatro camadas adicionadas às camadas dos modelos pré-treinados.

Tabela 3 - Camadas adicionadas aos modelos pré-treinados

(1) Flatten()
(2) Dense (1024, activation='relu')
(3) Dense (512, activation=' relu')
(4) Dense (1, activation='sigmoid')

A primeira camada é uma *Flatten* que recebe um tensor e o transforma num tensor unidimensional. A segunda e terceira camada são do tipo *dense*, que permitem aprender funções mais complexas e obter melhores resultados de classificação. A quarta camada representa a camada de classificação, tratando-se de uma camada *dense* com função de ativação S*igmoid.*

5.5.2 Segunda Abordagem

A segunda abordagem, segue a arquitetura de *CNN* da Figura 24 descrita no estudo Khan et al. (2020) .

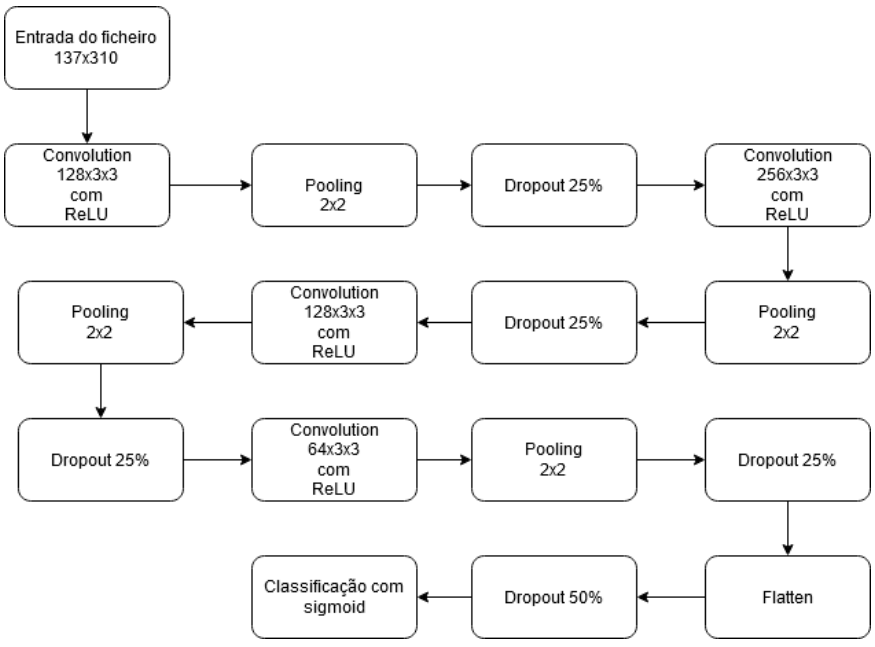


Figura 24 - Arquitetura *CNN* (Khan *et al.*, 2020)

A arquitetura é composta por quatro camadas *convolution* formadas por filtro de 3x3, uma função de ativação *ReLU* e com 128, 256, 128 e 64 filtros respetivamente. Cada camada *convolution* é seguida de uma camada *pooling* com um tamanho de 2x2 e uma camada *dropout*

(25%). Depois da última camada de *dropout*, a saída é convertida num único *vector* utilizando a camada *flatten*. O *vector,* de seguida, alimenta a camada de classificação com um *dropout* (50%). A camada de classificação utiliza a ativação *Sigmoid* por se tratar de uma classificação binária.

Esta arquitetura é utilizada na abordagem da Figura 25.

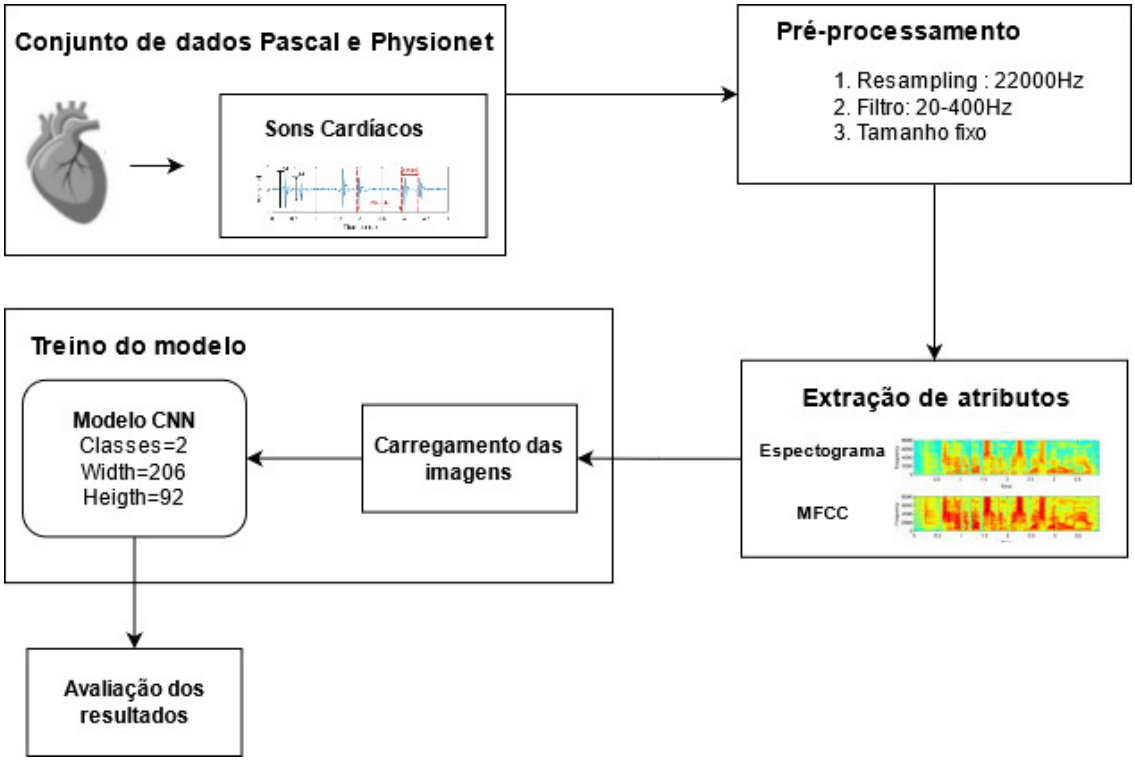


Figura 25 - Segunda Abordagem

A abordagem inicia com o pré-processamento dos sons de acordo com a Secção 5.2. De seguida, os sinais sonoros são convertidos em imagens utilizando os métodos propostos na Secção 5.3. Por último, a arquitetura *CNN* (Figura 24) é treinada, testada e avaliada de acordo com a Secção 5.4.

5.6 Implementação

Nesta secção descreve-se a implementação da solução proposta. Inicialmente apresenta-se o pré-processamento utilizado nos conjuntos de dados, seguido dos métodos de extração de atributos considerados e, por último, o carregamento das imagens e treino dos modelos.

5.6.1 Pré-processamento

O pré-processamento utilizado para o desenvolvimento da solução foi descrito na Secção 5.2. A Figura 26 apresenta os passos utilizados no pré-processamento.

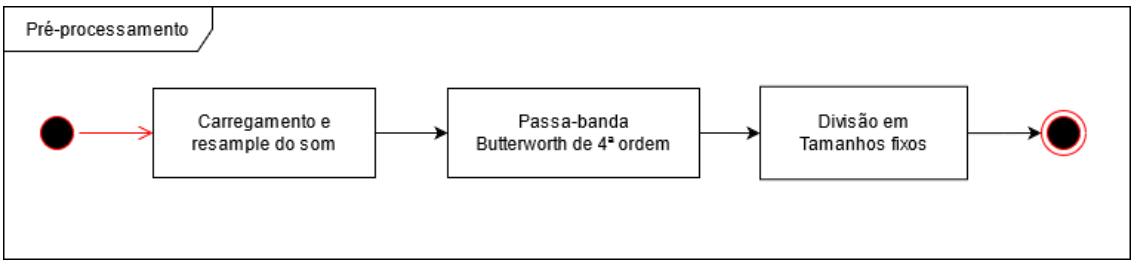


Figura 26 - Passos do pré-processamento

O primeiro passo consiste no carregamento e *resample* do som para um *sample rate* específico. De seguida, é aplicado um filtro passa-banda *Butterworth* de 4ª ordem de corte nas frequências 20Hz-400Hz e, por último, a divisão dos sinais de áudio em tamanhos iguais.

5.6.1.1 Carregamento e *resample* do som

O primeiro passo do pré-processamento é o carregamento dos dados. Os dados utilizados são provenientes de dois conjuntos de dados diferentes, que, por sua vez, necessitam de funções de carregamento distintas. O conjunto de dados PASCAL fornece três pastas distintas, em que uma contém os sons normais e duas os sons anormais. O conjunto *PhysioNet* é composto por seis pastas (A – F) em que cada uma dessas pastas contém um ficheiro que permite caracterizar cada um dos seus ficheiros entre o tipo normal ou anormal.

O carregamento do ficheiro de áudio e o seu *resample* é realizada por uma função comum aos conjuntos de dados. Essa função é apresentada no extrato de Código 1.

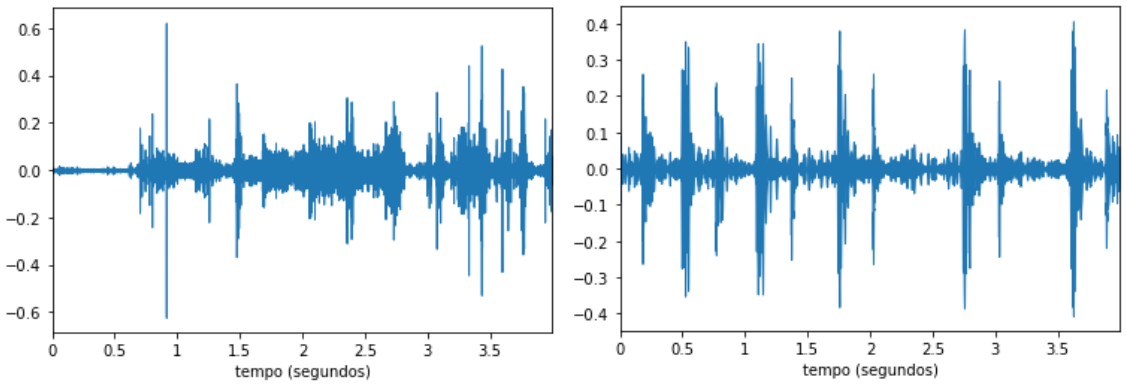
```
import librosa

     def load_wav_file(path,sample_rate=22000)

         wav,sr=librosa.load(path,sr=sample_rate)
         return wav,sr
```

Código 1 - Carregamento e *resample* de um ficheiro de áudio

A função recebe como parâmetro o ficheiro de som e o *sample rate* a que o som é *resampled* e devolve o sinal de áudio no *sample rate* definido.

A representação do sinal de áudio é conhecida como representação temporal. Nesta representação é demonstrada a amplitude do som ao longo do tempo em que a amplitude 0 representa o silêncio. Essa representação é demonstrada nas figuras 27 e 28.

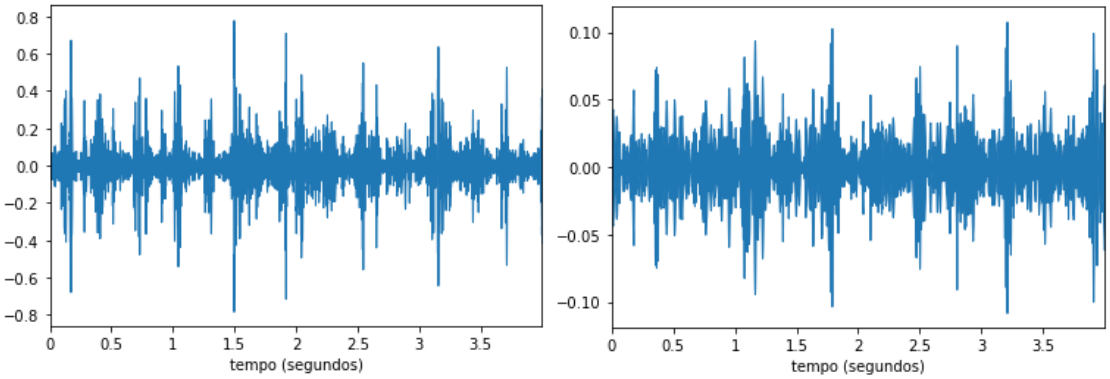
A Figura 27(a) apresenta um sinal do tipo normal proveniente do conjunto de dados PASCAL e na Figura 27(b) um som do tipo normal proveniente do conjunto de dados *PhysioNet*.



(a) Sinal de áudio 101_1305030823364_B (b) Sinal de áudio a0007

Figura 27 Representação do som do tipo normal do conjunto PASCAL e *PhysioNet*

A Figura 28(a) apresenta um exemplo de um som do tipo anormal do conjunto de dados PASCAL e na Figura 28(b) um som do tipo anormal do conjunto de dados *PhysioNet*.



(a) Sinal de áudio 135_1306428972976_C (b) Sinal de áudio a0008

Figura 28 - Representação do som do tipo anormal do conjunto de dados PASCAL e *PhysioNet*

Ao analisar as figuras 27 e 28 é possível observar que o conjunto PASCAL apresenta mais ruído do que o conjunto de dados *PhysioNet*.

5.6.1.2 Filtro do som

Depois do carregamento dos sinais de áudio, é necessário proceder à remoção do ruído existente no áudio que dificulta a identificação dos batimentos cardíacos. Para a remoção desse ruído é aplicado o filtro passa-banda *Butterworth* de 4ª ordem com frequências de corte de 20 Hz a 400 Hz.

A Figura 29 e 30 apresentam o sinal com ruído (a azul) e o sinal após a aplicação do filtro (a laranja).

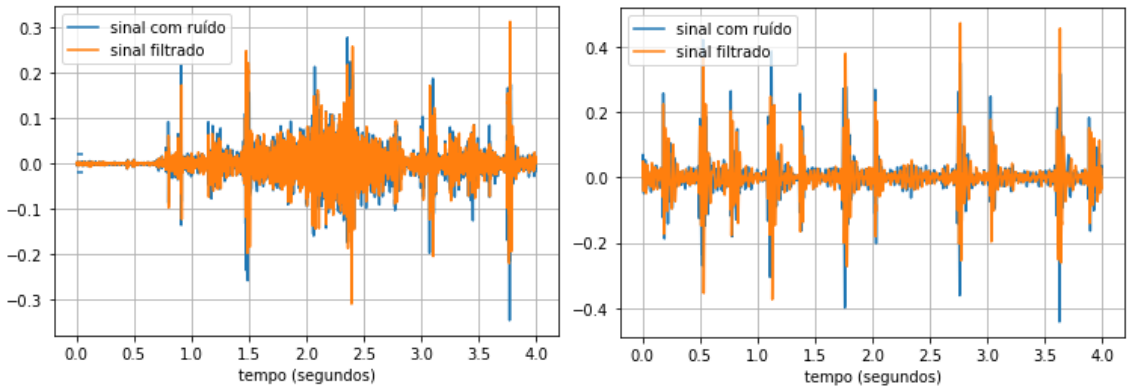


Figura 29 - Exemplo de aplicação do filtro no som normal
(ID: 101_1305030823364_B e ID: a0007)

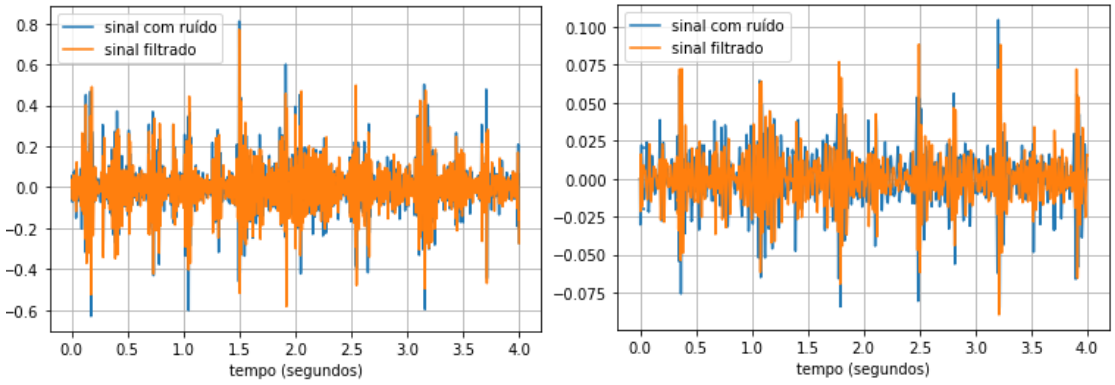


Figura 30 - Exemplo de aplicação do filtro no som anormal
(ID: 135_1306428972976_C e ID: a0008)

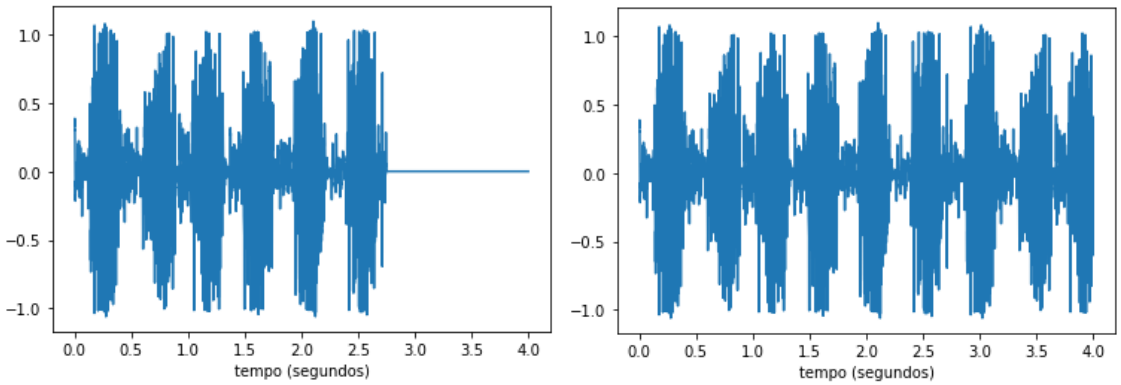
Analisando as figuras 29 e 30 pode-se verificar que a amplitude do sinal do som é bastante distinta comparativamente ao som original. De forma a validar o filtro aplicado, é possível converter novamente o sinal sonoro no formato "*.wav*" e verificar que os sons dos batimentos cardíacos são mais audíveis contendo menos ruído.

5.6.1.3 Tamanho fixo

Depois da aplicação do filtro nos sons, esses são divididos em amostras do mesmo tamanho, o que permite aumentar consideravelmente o conjunto de dados existente. Para a realização

deste processo, todos os sons inferiores ao tamanho da amostra definida são aumentados e todos os sons com tamanho superior são divididos em múltiplas amostras.

A Figura 31 apresenta dois processos diferentes que permite aumentar o tamanho do som de acordo com o tamanho de amostra definido.



(a) *Padding* do som (b) Replicação do som

Figura 31 - Processo de *padding* e repetição do som (ID: 193_1308078104592_C1)

Na Figura 31(a) é possível verificar que o som tem uma duração de aproximadamente 2.7 segundos e que ao resto foi aplicado um *padding* até aos 4 segundos. Na Figura 27(b) é apresentado o mesmo som no qual é aplicado uma replicação do mesmo até obter um som de 4 segundos.

5.6.2 Extração de atributos

Com o pré-processamento do som realizado, segue-se a extração de atributos (Figura 32). Para a realização deste processo foram utilizados espectogramas do tipo *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*. Para a conversão dos sinais de áudio em espectrograma foi utilizado a biblioteca Librosa (McFee *et al.*, 2015).

De seguida, os atributos extraídos são normalizados entre 0-255 e posteriormente gravados no formato "*PNG",* utilizando a biblioteca *Matplotlib* (Hunter, 2021). O formato "*PNG*", foi escolhido visto que mantém a qualidade da imagem no processo de compressão, ao contrário do "*JPEG"* que perde qualidade de imagem na sua gravação.

Na Figura 32 são apresentados os passos da extração de atributos.

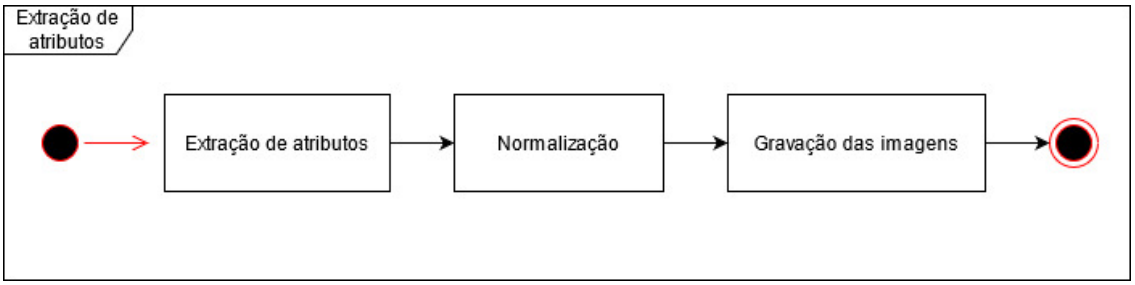


Figura 32 – Representação dos passos da extração de atributos

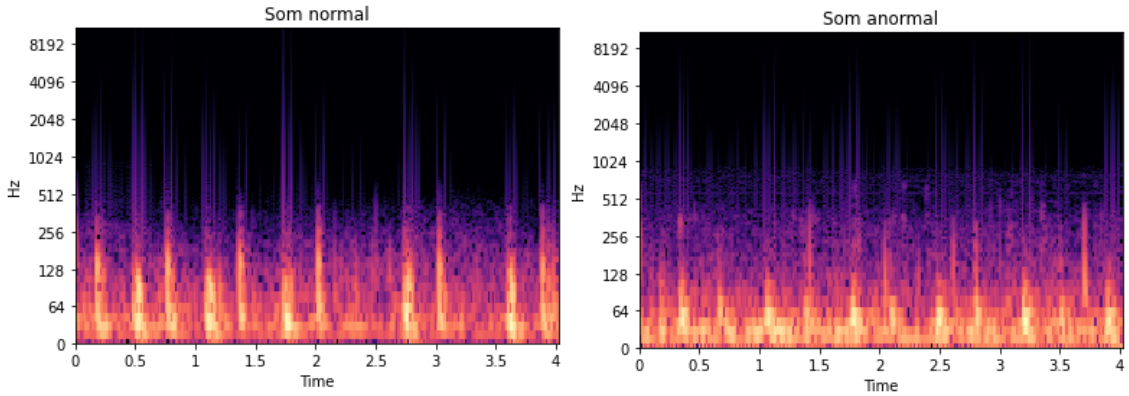
5.6.2.1 *STFT Spectrogram*

O extrato de Código 2 apresenta um exemplo da codificação do *STFT Spectrogram* de Khan *et al.* (2020).

```
def stftSpectrogram(wav, sr):

    n_fft=int(np.round(0.064*sample_rate)) #fft de 64ms
    win_length =  int(np.round(0.064*sample_rate))
    hop_length =  win_length/4
    window = 'hamming'

    stft = np.abs(librosa.core.stft(wav, n_fft=n_fft, hop_length=hop_length,
    win_length=win_length, window=window))

    spec = librosa.amplitude_to_db(stft, ref=np.max)
    spec=scale_minmax(spec, 0, 255).astype(np.uint8) #Normalização

    return spec
```

Código 2 - Criação do *STFT Spectrogram*

Na Figura 33 são apresentados exemplos da extração de atributos utilizando *STFT Spectrogram* no conjunto de dados *Physionet.* A Figura 33(a) apresenta a extração do som cujo nome é "a0007" e na Figura 33(b) com o nome "a0008".



(a) *STFT Spectrogram* do tipo normal (ID: a0007)

(b) *STFT Spectrogram* do tipo anormal (ID: a0008)

Figura 33 - Exemplo do *STFT Spectrogram*

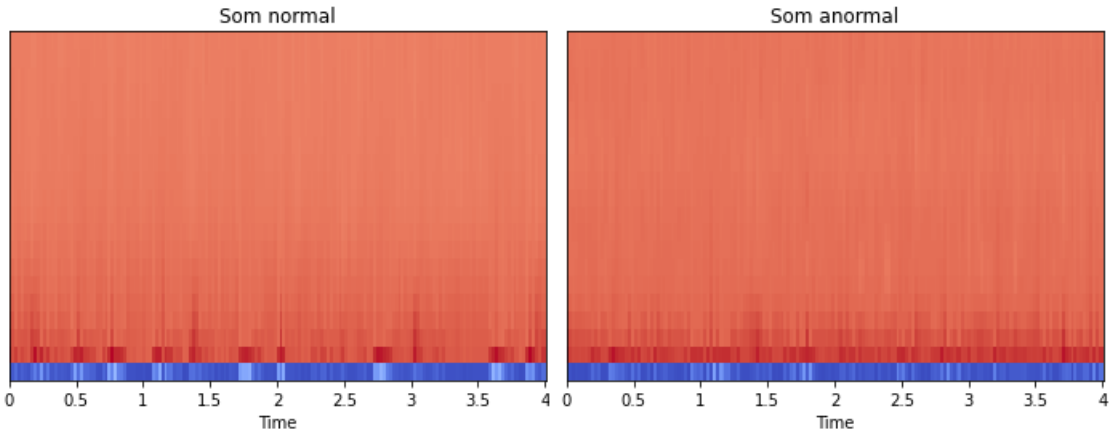
5.6.2.2 *MFCC*

O extrato de Código 3 apresenta um exemplo da codificação do *MFCC* de Boulares, Alafif and Barnawi (2020).

```
def mfcc(wav, sr):

    window = 'hamming'
    n_fft =1024 #tamanho fixo

    mfcc=librosa.feature.mfcc(y=wav,sr=sr,window=window, n_fft =
    n_fft, hop_length=hop_length)

    mfcc =scale_minmax(mfcc, 0, 255).astype(np.uint8) #Normalização

    return mfcc
```

Código 3 - Criação do *MFCC*

Utilizando os mesmos sons da Figura 33, na Figura 34 são apresentadas as extrações de atributos utilizando o *MFCC*.



(a) *MFCC* do tipo normal (ID: a0007)

(b) *MFCC* do tipo anormal (ID a0008)

Figura 34 - Exemplo do *MFCC*

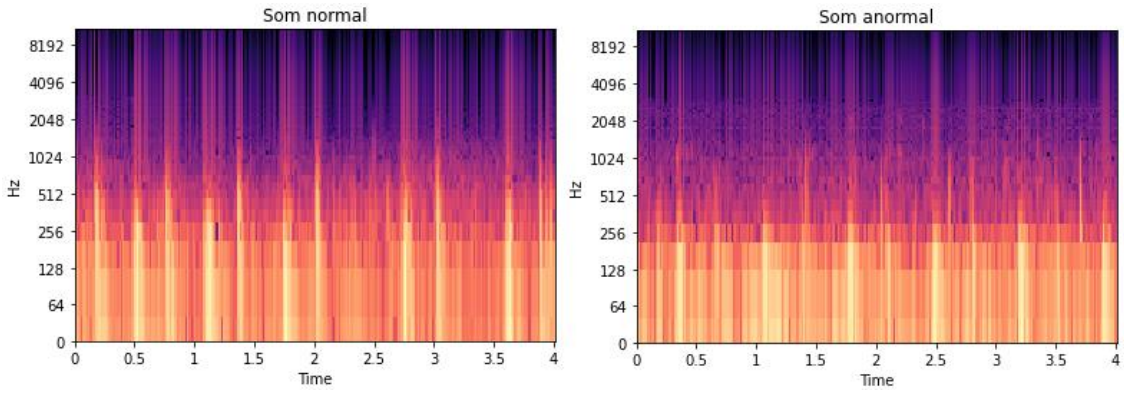
5.6.2.3 *Mel Spectrogram*

O extrato de Código 4 apresenta um exemplo de codificação de um *Mel Spectrogram* de Chowdhury, Poudel and Hu (2020).

```
def melSpec(wav, sr):

    window = 'hamming' #type
    n_fft = int(np.round(0.030*sample_rate))
    hop_length = int(np.round(0.015*sample_rate))

    ms =librosa.feature.melspectrogram(y=wave, sr=sr,hop_length =
    hop_length, n_fft=n_fft, window=window,n_mels=n_mels)

    ms = np.log(ms + 1e-9)
    ms =scale_minmax(ms, 0, 255).astype(np.uint8) #Normalização

    return ms
```

Código 4 - Criação do *Mel Spectrogram*

Utilizando os sons da Figura 33 e 34, na Figura 35 é apresentado a extração de atributos utilizando o *Mel Spectrogram.*



(a) *Mel Spectrogram* do tipo normal (ID: a0007)

(b) *Mel Spectrogram* do tipo anormal (ID: a0008)

Figura 35 - Exemplo do *Mel Spectrogram*

As configurações dos três tipos de espectrogramas são idênticas. O *n_fft* permite a divisão dos sinais em pequenos *frames (windows*) e o *hop_length* indica a distância entre as *windows,* possibilitando a existência de sobreposição das *windows* (ver Código 5). Uma *window* com um tamanho pequeno permite com que não seja perdida informação relevante do som e o tamanho da sobreposição garante que não sejam perdidas frequências do som. Por exemplo, no caso do *Mel Spectrogram* existe uma divisão em pequenas *windows* do tamanho de 30ms com uma sobreposição de 15ms, ou seja, uma sobreposição de 50%.

5.6.3 Carregamento dos dados e configuração dos modelos

Depois dos espectrogramas serem extraídos para imagem, estes ficam prontos para serem carregados novamente e serem utilizados para treinar o modelo *CNN*. O processo de pré-processamento e extração de atributos são processos comuns a todos os modelos, isto significa que depois de definido o melhor processo, este pode ser executado uma única vez e assim ser reutilizado para todos os modelos.

Depois das imagens serem carregadas, estas são representadas por vetores de valores inteiros e variam entre 0 e 255. As redes neuronais processam entradas usando pequenos valores de peso e entradas com grandes valores inteiros podem interromper ou retardar o processo de aprendizagem. Como é possível processar imagens com valores de *pixel* entre 0 e 1, é considerada uma boa prática dividir todos os valores (pixéis) pelo maior valor (255) e assim normalizar os dados.

Depois de os dados serem devidamente carregados e normalizados, é possível ter dois conjuntos de treinos distintos, um do tipo desbalanceado e um balanceado (ver Figura 36).

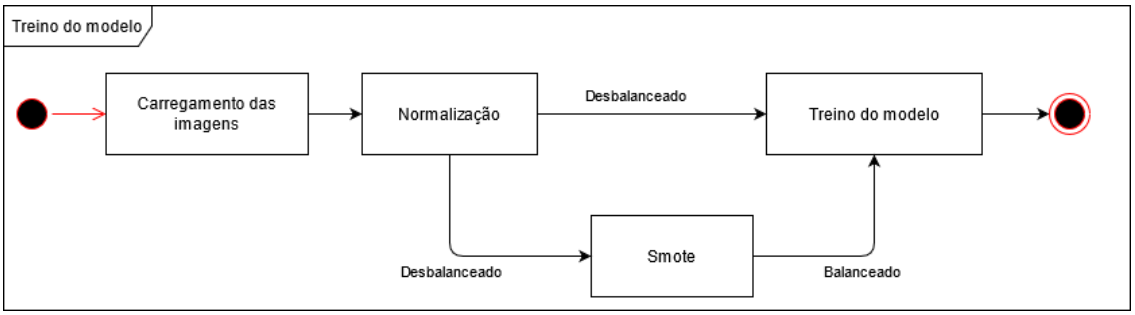


Figura 36 - Passos do carregamento das imagens e treino do modelo

5.6.3.1 Treino do modelo

O processo de treino do modelo é o passo no qual os modelos de *deep learning* aprendem mapeamentos dos valores de entradas que possibilitam a geração dos valores de saída de acordo com o problema em questão. Isto é possível através da atualização dos pesos da rede em resposta aos erros do modelo nos dados do conjunto de treino. As atualizações são feitas para reduzir continuamente o erro até que um modelo seja bom o suficiente ou até ao momento do treino em que o modelo deixe de melhorar. Neste processo é comum ser usado a função *fit* (Keras, 2021) fornecida pela *API Keras.*

Na realização do processo de treino foi necessário garantir que para cada etapa, o conjunto de teste estava a ser avaliado por paciente e não por sons. Como os sons dos pacientes apresentam tamanhos distintos que resultam em número de amostras diferentes, não foi possível a utilização da função *fit*. Por este motivo, foi necessário criar um processo de treino que permitisse em cada iteração treinar e avaliar o modelo utilizando um número de sons variável. Para tal, foi feita a atualização manual dos pesos da rede em cada iteração utilizando o *gradiente tape* (Tensorflow, 2021).

A primeira abordagem definida para a atualização manual dos pesos através do *gradiente tape* consiste no processo de treino e teste por paciente. Isto significa que para cada iteração do treino é necessário realizar uma previsão para todos os sons do paciente e calcular o respetivo erro para posteriormente os pesos da rede serem atualizados. O número de iterações da etapa corresponde ao número de pacientes definidos para treino. Contudo, durante a fase de avaliação, os resultados obtidos não foram satisfatórios. Isto deve-se ao facto de o modelo se ajustar demasiado à classe maioritária e aos seus pacientes, perdendo a capacidade de generalizar (*overfit*). Para resolver este problema, foi necessário definir um tamanho fixo de valores aleatórios por iteração, permitindo assim que o modelo atualizasse os pesos em cada iteração com sons de diversos pacientes.

O extrato de Código 5 apresenta como é possível atualizar os pesos manualmente.

```
def stepTrain(X,Y):
     X=tf.Variable(X)
      with tf.GradientTape() as tape:

         pred = model(X,training=True)
        loss = tf.keras.losses.binary_crossentropy(y,pred,from_logits=True)

      total_loss=tf.reduce_mean(loss)

      grads = tape.gradient(loss, model.trainable_variables)
      opt.apply_gradients(zip(grads, model.trainable_variables))

      ...

       return total_loss,total_acc
```

Código 5 - Exemplo da atualização manual dos pesos

A função recebe como entrada os sons do paciente e a sua classe correspondente. O modelo faz a predição e é calculado o erro correspondente.

No final de cada etapa de treino, é importante garantir a capacidade do modelo avaliar o estado do paciente e não o som. Por este motivo, o conjunto de teste é agrupado por paciente permitindo assim fazer predições com o modelo unicamente a sons do mesmo paciente. Para determinar o estado do paciente, foi considerado que o estado do paciente era do tipo anormal se um dos sons do paciente fosse considerado positivo. As métricas finais do modelo são geradas após a previsão de todos os pacientes.

6 Avaliação

6.1 Métricas de avaliação

A metodologia de avaliação utilizada para determinar o melhor modelo para ambas as abordagens definidas foi o *Hold-out* em que 80% dos dados são utilizados para o treino do modelo e os restantes 20% utilizados para a sua validação. As métricas de avaliação consideradas para a sua avaliação é a *accuracy*, *F1 score* e *loss*. No entanto, a métrica *F1 score* será a principal medida de avaliação por fornecer uma média harmónica entre o *recall* e *precision*.

Na experiência final, os modelos que obtiveram melhores resultados são comparados com os resultados do artigo de Nogueira *et al.* (2019). Para isso, foi utilizado metodologia de avaliação *K-fold stratified cross validation* com um *K*=10. A principal métrica utilizada para a comparação é o *overall*, descrito na secção 4.6.4. Para melhor entender a diferença entre os modelos utilizados foram apresentadas e comparadas as curvas *ROC* e a respetiva *AUC* para cada um dos modelos.

No processo de calculo das métricas de ambas a metodologia é necessário ter em consideração que o objetivo do trabalho consiste na classificação do paciente. Por este motivo, todos os sons do mesmo paciente são avaliados em conjunto dando origem a um resultado do tipo normal ou anormal. Foi considerado que um paciente tem DCV se em alguns dos sons do paciente o modelo considerar com som anormal.

6.2 Conjuntos de teste

Durante o processo de criação do conjunto de testes são considerados unicamente sons do conjunto de dados *PhysioNet*. Foi considerado que cada som deste conjunto corresponde a um paciente. Isto significa que num total de 3240 sons, estes correspondem a 3240 pacientes, em que aproximadamente 80% dos sons são do tipo normal e 20% do tipo anormal.

No processo de treino foram utilizados os conjuntos de dados PASCAL e *PhysioNet* e considerados dois conjuntos de treino, um do tipo balanceado e um outro do tipo desbalanceado.

Os conjuntos balanceados são criados igualando a classe minoritária (sons anormais) à classe maioritária (sons normais). A abordagem mais simples para essa tarefa envolve a duplicação de exemplos da classe minoritária até que tenha a mesma proporção da classe maioritária. Esta abordagem não é eficiente na medida em que os novos exemplos não adicionam informação ao modelo e consequentemente podem causar *overfitting*. Por esse motivo, foi utilizada uma outra técnica denominada de *Synthetic Minority Oversampling Technique* (*SMOTE*) para a criação de dados sintéticos. Esta técnica começa por escolher os dados da classe minoritária de

forma aleatória e, de seguida, são definidos *K*-vizinhos mais próximos dos dados. Os dados sintéticos seriam então criados entre os dados aleatórios e o vizinho mais próximo *k* selecionado aleatoriamente (Brownlee, 2020c).

6.3 Hipótese

A hipótese nula afirma que a *AUC* das duas abordagens apresenta resultados semelhantes, a rejeição da hipótese nula significa que existe diferença entre a *AUC* de pelo menos um dos pares de modelos.

Para a comparação entre a *AUC* do modelo foi utilizado o teste não paramétrico DeLong e de seguida aplicado a correção de Bonferroni (DeLong, DeLong and Clarke-Pearson, 1988; Adam Hayes, 2020). O teste de DeLong permite a comparação entre dois modelos e identificar se existe diferença estatística entre as áreas da *AUC*. Como se pretende testar um total de quatro modelos (várias hipóteses), existe a possibilidade de cometer um erro que pode resultar na rejeição da hipótese nula incorretamente. Assim, foi aplicada a correção de Bonferroni que ajusta o nível de significância considerado (α) para α /*m*, sendo *m* é o número de hipóteses testadas.

6.4 Experiências

Nesta secção são apresentadas as principais experiências realizadas à solução desenvolvida. As experiências iniciam-se com a comparação dos tamanhos das amostras, o que permite definir o tamanho a ser utilizado nas experiências seguintes. De seguida, são extraídos e comparados os espectrogramas: *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*, com base nos artigos descritos na Secção 4.3. Após determinar o tamanho fixo e o método de extração de atributos, os modelos pré-treinados (segunda abordagem) são comparados com o objetivo de determinar o melhor modelo. A experiência final consiste na comparação das abordagens definidas (5.5.1 e 5.5.2) com o artigo de Nogueira *et al.* (2019). Nesta comparação é utilizado o melhor tamanho fixo, espectrograma e modelo pré-trainado encontrado nos passos anteriores.

Para a realização das experiências foi seguida a metodologia de avaliação *Hold-out* com um máximo de *epochs* de 50. Este número (máximo) de *epochs* foi definido após a realização de testes aos modelos onde foi possível verificar que estes quase não melhoravam após as 50 *epochs* e os valores de *loss* aumentavam o que indica que a existência de *overfitting.*

Para cada uma da experiência do modelo são criados três conjuntos de treino e de teste diferentes. Estes três conjuntos de treino e teste são partilhados por todos os modelos o que permite assim garantir que os modelos são avaliados e treinados usando os mesmos conjuntos de dados. A métrica final corresponde à média dos três resultados obtidos pelos modelos treinados.

No processo da criação dos conjuntos de treino e teste foi garantido que os sons de cada paciente se encontravam apenas num dos conjuntos (treino e teste). Este passo foi importante para garantir que o modelo não tinha conhecimento do paciente e assim não influenciar a avaliação do modelo.

6.4.1 Tamanho fixo das amostras

Como referido anteriormente, as experiências do tamanho fixo das amostras permite definir o tamanho no qual os dados são divididos e o melhor resultado é obtido. Os tamanhos fixos utilizados para os testes são amostras com o tamanho de 4, 5 ,6 e 7 segundos. Para a realização destes testes foi utilizada a segunda abordagem descrita na Secção 5.2, o *STFT Spectrogram* para a extração de atributos descrito na Secção 5.6.2 e o conjunto de dados PASCAL e *PhysioNet*. As métricas consideradas foram *accuracy*, *F1 score* e *loss*.

Na tabela 4 é possível ver a distribuição do conjunto de dados para cada um dos diferentes tamanhos da amostra.

Tabela 4 - Total de sons por tamanho das amostras

Tamanho Fixo	Total Normal	Total Anormal	Sons totais
4 (s)	12642 (74%)	4340 (26%)	16 982
5 (s)	10 023 (74%)	3 589 (26%)	13 612
6 (s)	8236 (74%)	2867 (26%)	11 103
7 (s)	6985 (73%)	2588 (27%)	9 573

A primeira coluna da tabela 4 representa o tamanho fixo definido para as amostras, a segunda e terceira coluna representam o total de amostras do tipo normal e anormal e a quarta o total de amostras.

Para a divisão do som foram consideradas a alternativa *padding* do som ou a repetição do som definidos na Secção 5.6.1. De forma a definir qual a melhor abordagem a seguir, foram testadas ambas as abordagens para cada um dos tamanhos fixos definidos na tabela 4.

A tabela 5 apresenta os resultados da comparação entre os tamanhos fixos aplicando o método *padding* e replicação do som. Os resultados apresentados correspondem à média das três experiências para cada um dos tamanhos fixos definidos.

Tabela 5 - Comparação entre *padding* e repetição

		Conjunto de treino			Conjunto de teste		
Tipo	**Tamanho Fixo**	***F1 score***	***Accuracy***	***Loss***	***F1 score***	***Accuracy***	***Loss***
Padding	4 (s)	94.47%	95.83 %	0.098	86.36%	90.22 %	0.234
	5 (s)	92.31%	93.42%	0.131	85.05%	89.24%	0.231
	6 (s)	94.35%	95.76%	0.102	**87.82%**	91.46%	0.219
	7 (s)	94.69%	95.89%	0.097	86.27%	90.48%	0.238
Replicação	4 (s)	94.34%	95.89%	0.095	85.56%	89.50%	0.236
	5 (s)	94.28%	95.77%	0.394	84.78%	88.78%	0.231
	6 (s)	95.06%	96.28%	0.089	87.18%	91.10%	0.224
	7 (s)	95.55%	96.54%	0.086	86.71%	90.84%	0.233

Como é possível verificar na tabela 5, o melhor resultado obtido para as medidas consideradas foi o tamanho fixo de 6 segundos, seguido do tamanho fixo de 7 segundos e 4 segundos. Os resultados obtidos pelo *padding* e replicação de 6 segundos são muito idênticos, sendo que o *padding* apresenta mais 0.64% do que a replicação.

As matrizes de confusão das três experiências utilizando *padding* de 6 segundos são apresentadas na Figura 37.

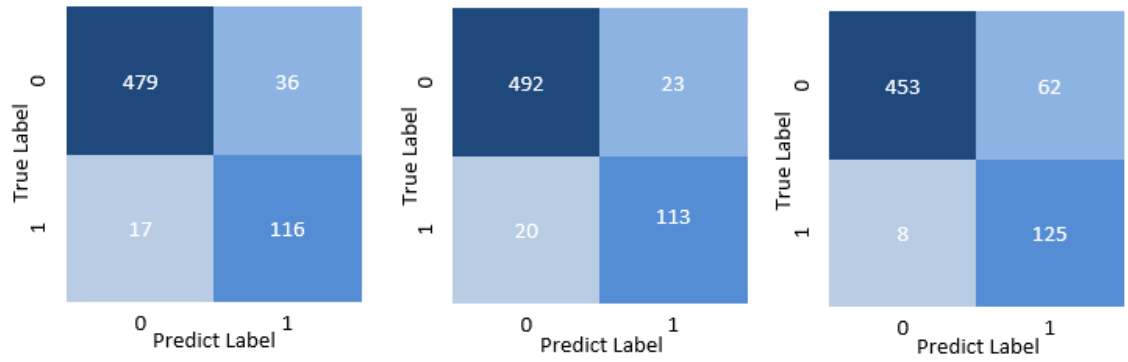


Figura 37 - Resultados das três experiências do melhor modelo

Cada matriz de confusão da Figura 37 apresenta o resultado de uma experiência do modelo com *padding* de 6 segundos numa amostra estratificada de 648 pacientes (conjunto de treino).

A classe 0 da matriz de confusão corresponde aos sons do tipo normal e a classe 1 aos do tipo anormal.

Na Figura 38 pode observar-se os resultados obtidos para as medidas *accuracy*, *loss* e *F1 score* da melhor experiência (a azul é apresentado o treino do modelo e a cor de laranja a validação).

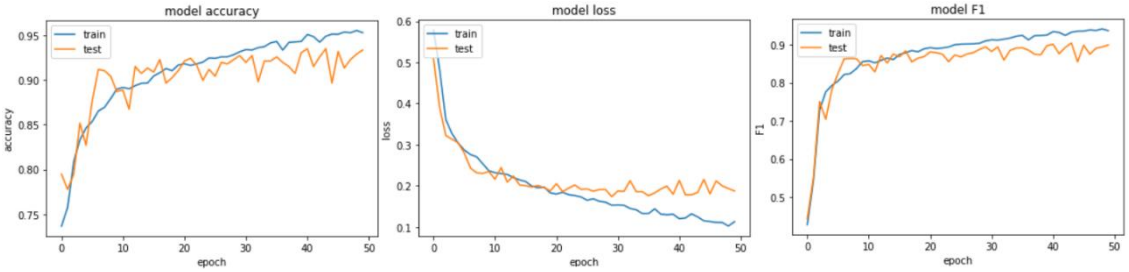


Figura 38 - Representação da *accuracy*, *loss* e *F1 score*

Como é visível na Figura 38, a linha de treino e teste ambas seguem uma direção similar em ambos os gráficos até ao momento em que a linha fica constante, deixando de melhorar o resultado.

6.4.2 Comparação entre métodos de extração de atributos no som

O objetivo desta experiência consiste na comparação entre o *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*, descritos na Secção 5.6.2.

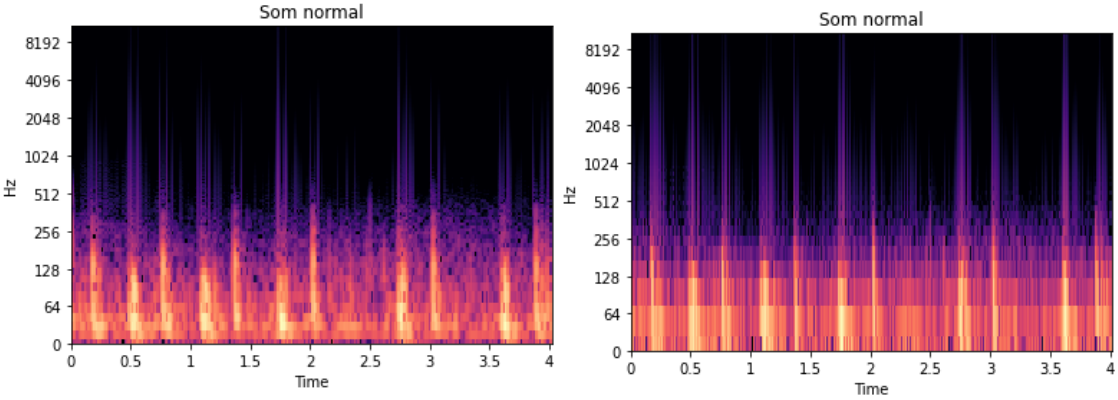
As experiências foram realizadas utilizando a segunda abordagem com o conjunto de dados PASCAL e *PhysioNet* com uma amostra de tamanho fixo de 6 segundos determinada na secção anterior. Para cada um dos métodos são apresentadas duas alternativas baseadas na literatura.

A tabela 6 apresenta os parâmetros para a extração do *STFT Spectrogram* para um *sample rate* de 22000.

Tabela 6 - Parâmetros para a extração do *STFT Spectrogram*

Autor	**Alternativa**	**Parâmetros**	
		n_fft	***hop_length***
(Khan *et al.*, 2020)	1	1408	352
(Oliveira and Praca, 2021)	2	440	220

Na Figura 39 são apresentadas as extrações do *STFT Spectrogram* utilizando as alternativas da tabela 6.



(a) Alternativa 1 (b) Alternativa 2

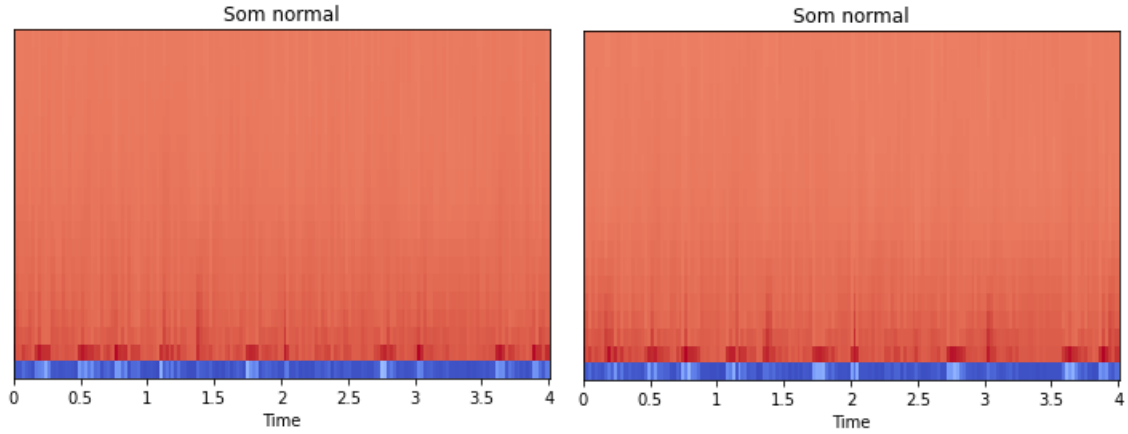
Figura 39 - Exemplo da extração do *STFT Spectrogram* (ID: a0007)

A tabela 7 apresenta os parâmetros para a extração do *MFCC* para um *sample rate* de 22000.

Tabela 7 - Parâmetros para a extração do *MFCC*

Autor	**Alternativa**	**Parâmetros**	
		n_fft	***hop_length***
(Nogueira *et al.*, 2019)	1	550	220
(Boulares, Alafif and Barnawi, 2020)	2	1024	256

A Figura 40 são apresentadas as extrações do *MFCC* utilizando as alternativas da tabela 7.



(a) Alternativa 1 (b) Alternativa 2

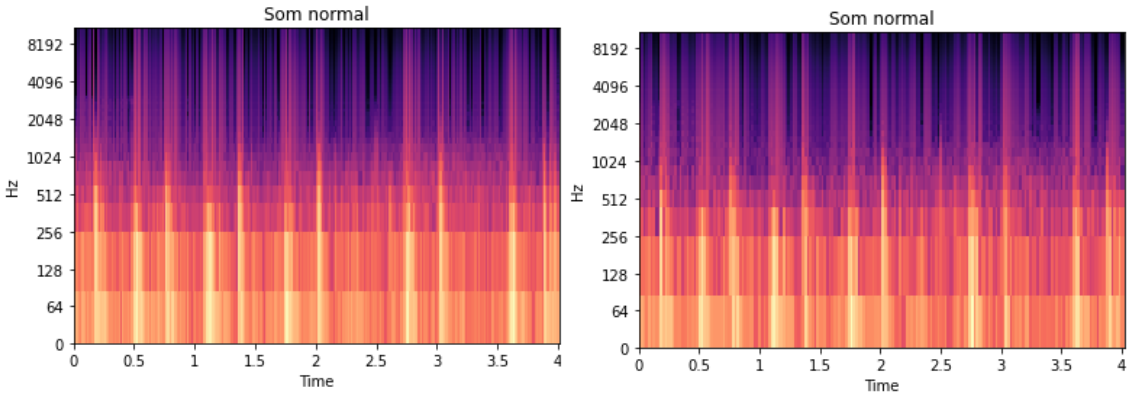
Figura 40 - Exemplo da extração do *MFCC* (ID: a0007)

A tabela 8 apresenta os parâmetros para a extração do *Mel Spectrogram* para um *sample rate* de 22000.

Tabela 8 - Parâmetros para a extração do *Mel Spectrogram*

Autor	**Alternativa**	**Parâmetros**	
		n_fft	***hop_length***
(Chowdhury, Poudel and Hu, 2020)	1	660	330
(Koike *et al.*, 2020)	2	440	440

A Figura 41 são apresentadas as extrações do *Mel Spectrogram* utilizando as alternativas da tabela 8.



(a) Alternativa 1 (b) Alternativa 2

Figura 41 - Exemplo da extração do *Mel Spectrogram* (ID: a0007)

Para obter resultados mais confiáveis, os três conjuntos de treino e teste definidos inicialmente são executados duas vezes e é utilizada a sua média para a comparação entre os espectrogramas.

Na tabela 9 apresenta-se a comparação entre os *STFT Spectrogram*, *MFCC* e *Mel Spectrogram.*

Tabela 9 - Comparação entre *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*

	Conjunto de treino			Conjunto de teste		
Alt.	***F1 score***	***Accuracy***	***Loss***	***F1 score***	***Accuracy***	***Loss***
***STFT* (1)**	94.98%	96.22	0.091	87.15%	90.97%	0.229
***STFT* (2)**	94.34%	95.72%	0.104	85.55%	89.56%	0.234
***MFCC* (1)**	92.89%	94.65%	0.131	76.98%	81.74%	0.449
***MFCC* (2)**	92.26%	91.59%	0.149	78.82%	83.59%	0.377
***Mel* (1)**	92.48%	94.32%	0.190	87.37%	91.23%	0.209
***Mel* (2)**	91.86%	93.86%	0.142	**87.45%**	91.36%	0.193

Como é possível observar na tabela 9, a alternativa 2 do *Mel Spectrogram* conseguiu um *F1 score* médio de 87.45% sendo o melhor resultado obtido comparativamente com as restantes alternativas. É possível verificar que ambas as alternativas do *Mel Spectrogram* em conjunto com a alternativa 1 do *STFT Spectrogram* conseguiram resultados muito aproximados. Por fim, os resultados mais baixos foram alcançados pelo *MFCC* em que o melhor resultado foi de 78.82%.

A Figura 42 apresenta a *accuracy*, *loss* e *F1 score* da melhor iteração da alternativa 2 do *Mel Spectrogram* em que a azul é apresentado o treino do modelo e a cor de laranja a validação.

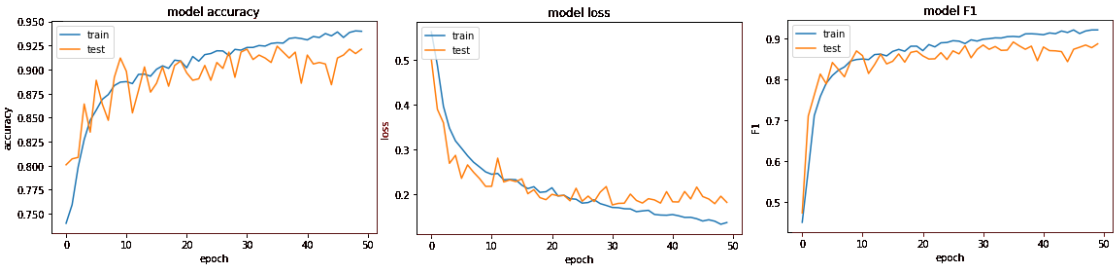


Figura 42 - Representação da *accuracy*, *loss* e F1

Como é visível na Figura 42, a linha de teste e treino seguem a mesma direção até perto do *epoch* 30, mantendo-se contante nos *epochs* seguintes.

6.4.3 Modelos pré-treinados

Depois de ser determinado o tamanho fixo das amostras e o melhor espectrograma com as respetivas configurações, esses são utilizados para a realização dos testes da abordagem 1 definida na Secção 5.5.1.

Tabela 10 - Resultados dos testes realizado aos modelos pré-treinados (abordagem 1)

	Conjunto de treino			Conjunto de teste		
Alt.	***F1 score***	***Accuracy***	***Loss***	***F1 score***	***Accuracy***	***Loss***
MobileNetV2	99.01%	99.25%	0.022	82.17%	86.37%	0.607
Xception	99.15%	99.36%	0.015	86.14%	90.07%	0.8019
ResNet50	99.49%	99.62%	0.011	82.24%	86.47%	0.570
VGG-16	92.47%	94.29%	0.127	85.83%	89.92%	0.242
VGG-19	91.23%	93.42%	0.141	**86.20%**	90.33%	0.237
Baidu	87.95%	90.78%	0.214	75.30%	80.09%	0.393

De acordo com os resultados da tabela 10, o melhor resultado foi alcançado pelo *VGG-19* com uma diferença de apenas 0.06% do *Xception* e de 0.37% do *VGG-16*.

A Figura 43, 44 e 45 apresentam uma comparação entre a *accuracy*, *loss* e *F1 score* do conjunto de teste dos três melhores modelos.

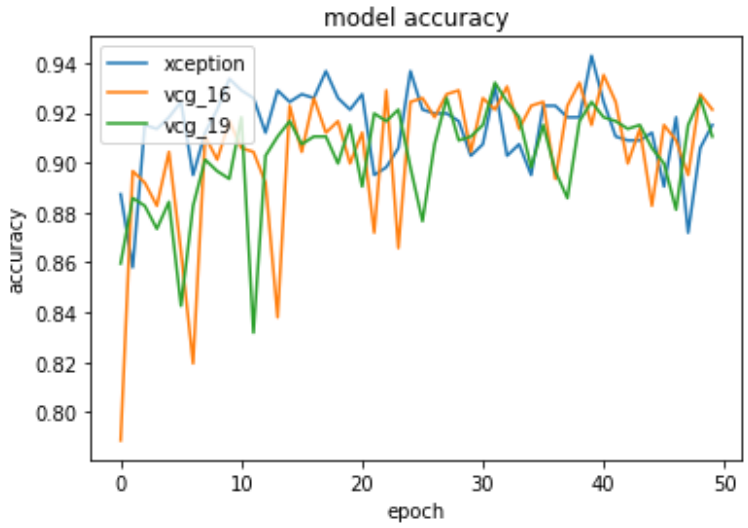


Figura 43- Resultado da *accuracy* dos modelos *Xception*, *VGG-16* e *VGG-19*

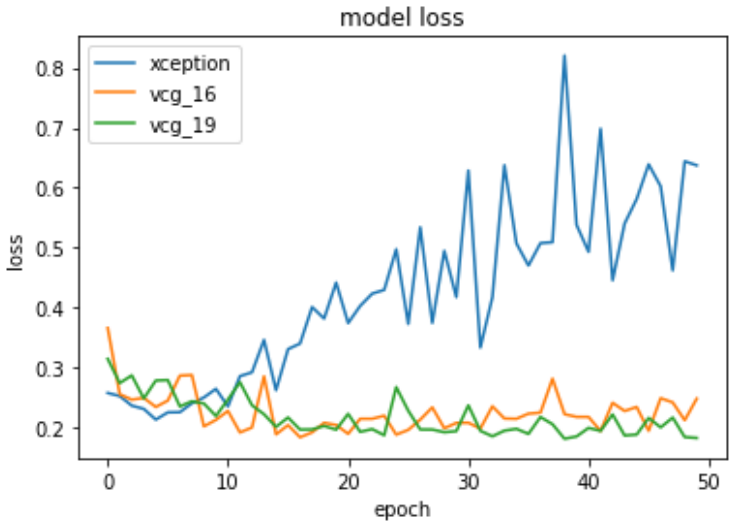


Figura 44 - Resultado da *loss* dos modelos *Xception*, *VGG-16* e *VGG-19*

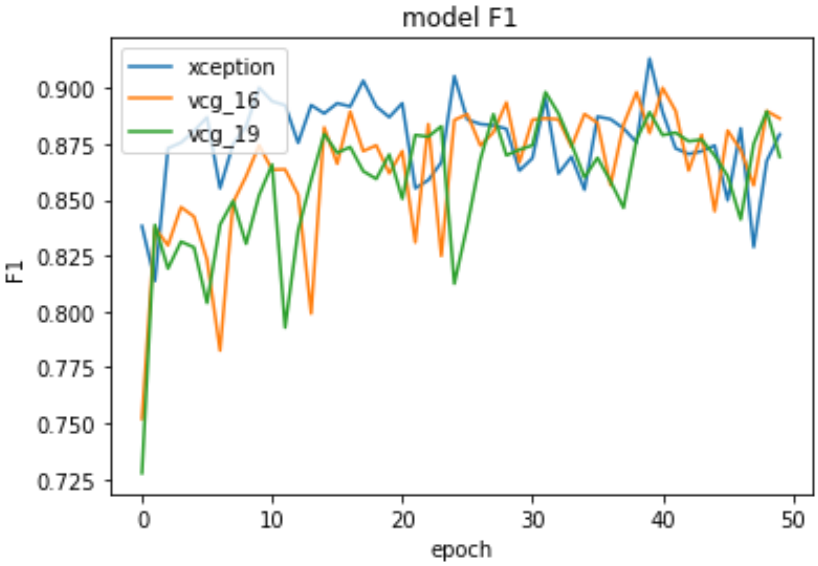


Figura 45 - Resultado do *F1 score* dos modelos *Xception*, *VGG-16* e *VGG-19*

Como se observa na Figura 43 e 45, o pior resultado (*accuracy* e *F1 score*) foi alcançado pelo modelo *VGG-19* que corresponde ao modelo, que na média dos três testes, obteve o melhor resultado (ver tabela 10). Ao analisar a Figura 44, que apresenta a comparação da *loss* entre os três modelos, é possível verificar que a linha do modelo *VGG-19* e *VGG-16* encontram-se a decrescer ou constante. O mesmo não acontece com a linha do modelo *Xception*, que ao fim de poucos *epochs* vai na direção crescente. Este comportamento tem a ver com a arquitetura visto que com a alteração de *learning rate* a *loss* tem um comportamento semelhante.

Através dos resultados obtidos e pela análise dos gráficos, foi considerado como o melhor modelo pré-treinado o *VGG-19*.

6.4.4 Experiência final

Nesta secção é realizada a comparação com o artigo de Nogueira *et al.*(2019) que aplica uma metodologia de testes que permite a comparação dos resultados com o concurso de *PhysioNet*. O melhor resultado obtido neste artigo, consiste num *recall* de 87.37%, *precision* de 79.07% e o *overall* de 83.22% posicionando-o no Top 10 melhores resultados do concurso *PhysioNet* (Latif *et al.*, 2018). A métrica *overall* apresentada na Secção 4.6.4, é a métrica utilizada no concurso *PhysioNet* para a comparação dos resultados. Esta também será a métrica utilizada para a comparação dos resultados de Nogueira *et al.* (2019) com os resultados obtidos no trabalho que aqui se apresenta.

A metodologia utilizada por Nogueira *et al.* (2019) consiste na utilização do *K-fold stratified cross validation* descrita na Secção 4.6.1 com um *K*=10. Na aplicação desta metodologia é importante garantir que os sons do mesmo paciente não constem no conjunto de treino e de teste, mas apenas num. Esta metodologia garante que todos os sons do conjunto de dados *PhysioNet* sejam só testados uma única vez e que tenham estado no conjunto de treino.

Os dois modelos selecionados para a aplicação da metodologia descrita consistem na arquitetura *CNN* definida na secção 5.5.2 e no modelo pré-treinado *VGG-19*. Para cada um dos modelos, será realizada uma experiência em que os dados de treino serão balanceados utilizando a técnica *SMOTE*.

Nesta experiência, tendo em conta as Figuras 38, 42 e 44, foi considerado um máximo de 30 *epoch* dado que os modelos deixavam de melhorar ao fim de 30 iterações de treino.

Na Tabela 11 é apresentada a comparação dos resultados alcançados pelos modelos desenvolvidos nesta dissertação e o melhor resultado alcançado por Nogueira *et al.*(2019).

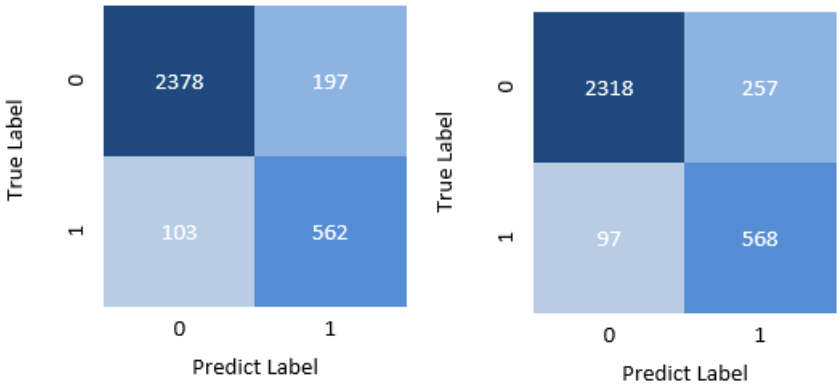
Tabela 11 - Comparação dos resultados

Alt.	***Recall***	***Precision***	***Overall***	***Accuracy***	***Loss***
(Nogueira *et al.*, 2019)	87.37%	79.07%	83.22%		
CNN	87.72%	82.42%	85.07%	89.07%	0.2140
***CNN* (*SMOTE*)**	88.43%	84.95%	**86.69%**	90.74%	0.2173
VGG-19	87.24%	80.32%	83.79%	87.43%	0.2393
***VGG-19* (*SMOTE*)**	86.38%	80.67%	83.52%	87.75%	0.2698

Os resultados obtidos pela *CNN*, *CNN* (*SMOTE*), *VGG-19* e *VGG-19* (*SMOTE*) conseguiram superar os resultados obtidos por Nogueira *et al.*(2019). É possível verificar que a utilização do

SMOTE permitiu melhorar significativamente os resultados do modelo *CNN*, o que o mesmo não aconteceu para o modelo *VGG-19*.

Na Figura 46 é apresentado a matriz de confusão correspondente ao modelo da *CNN* (*SMOTE*) e *CNN*.

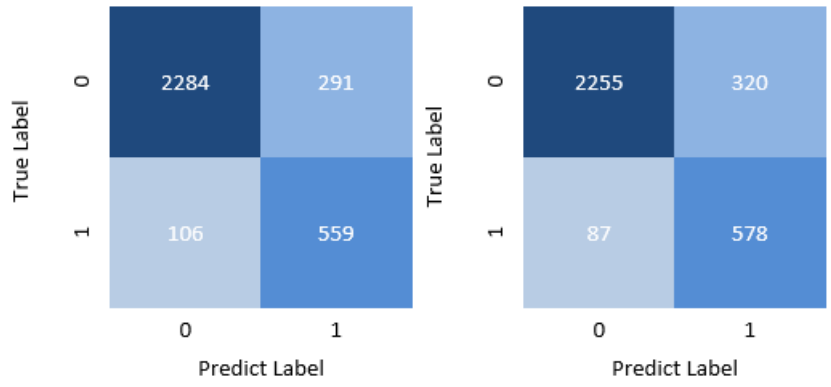


(a) *CNN* (*SMOTE*) (b) *CNN*

Figura 46 - Matriz de confusão do modelo *CNN* (*SMOTE*) e *CNN*

O modelo apresentado na Figura 46(a) conseguiu uma classificação de mais 60 pacientes do tipo normal do que o modelo da Figura 46(b). No entanto, o mesmo modelo errou na classificação mais 6 pacientes do tipo anormal. Na Tabela 11 é possível verificar que esta diferença no número de pacientes previstos corretamente permitiu ao modelo da Figura 46(a) ter um melhor *overall* que o modelo da Figura 46(b).

A Figura 47 apresenta a matriz de confusão do modelo da *VGG-19* (*SMOTE*) e *VGG-19*.



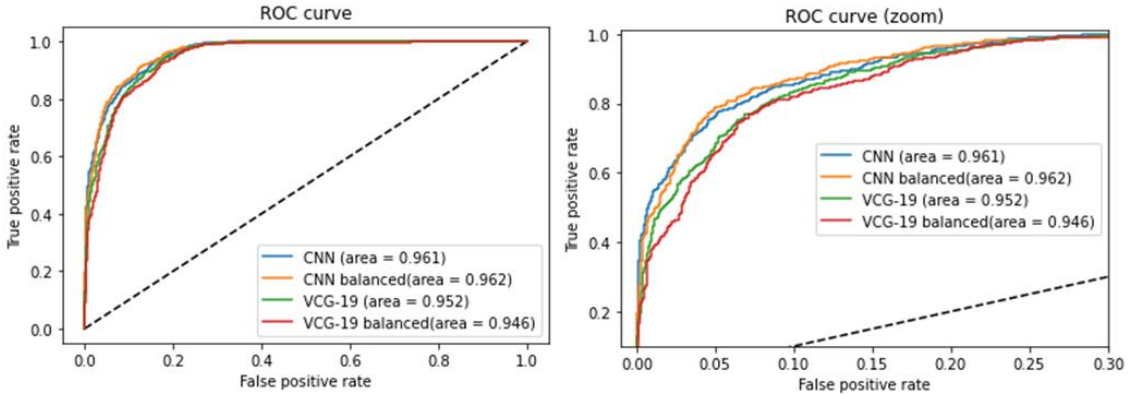
(a) *VGG-19* (*SMOTE*) (b) *VGG -19*

Figura 47 - Matriz de confusão do modelo *VGG-19* (*SMOTE*) e *VGG-19*

O modelo da Figura 47(a) conseguiu uma classificação de mais 29 pacientes do tipo normal e menos 19 pacientes do tipo anormal, comparativamente com o modelo da Figura 47(b). Tendo

em conta a Tabela 11, é possível verificar que os pesos dos pacientes do tipo anormal nas métricas finais fazem com que o modelo da Figura 47(b) conseguisse obter um *overall* superior ao do modelo da Figura 47(a).

Para melhor entender o desempenho e a diferença entre os quatro modelos é apresentado a curva *ROC* para cada um dos modelos. A curva *ROC* é uma métrica muito utilizada para avaliar o desempenho de modelos binários. Esta permite verificar se o modelo consegue discriminar as classes tendo em conta o desbalanceamento das mesmas. Na Figura 48(a) é apresentado a curva *ROC* para cada um dos modelos e na Figura 48(b) uma ampliação da zona significante das curvas dos modelos.



(a) Curvas *ROC* dos modelos

(b) Ampliação das curvas *ROC*

Figura 48 - Comparação da curva *ROC* entre os modelos

Como se observa na Figura 48, todas as curvas *ROC* dos modelos encontram-se acima da linha a tracejado o que significa que os modelos não estão a fazer previsões aleatoriamente. O modelo que melhor *AUC* (a área abaixo da curva) consiste na *CNN* balanceada com uma diferença da *AUC* de 0.1% da *CNN*, 1% da *VGG-19* e de 1.6% da *VGG* balanceada.

Para comparar e determinar se existe diferença estatística entre a *AUC* dos quatro modelos foi aplicado o teste de DeLong (DeLong, DeLong and Clarke-Pearson, 1988).

A tabela 12 apresenta a diferença estatística entre o modelo A e modelo B tendo em conta um nível de significância de 5%.

Tabela 12 - Resultados da aplicação o teste de DeLong na primeira e segunda abordagem

Modelo A	Modelo B	Variância de A	Variância de B	Covariância de A\B	Z-score	P-value
CNN	***CNN* (*SMOTE*)**	9.28e-06	9.12e-06	6.69 e-06	-0.481	0.6303
CNN	***VGG-19***	9.28e-06	1.20e-05	6.49e-06	3.251	0.0011
CNN	***VGG-19* (*SMOTE*)**	9.28e-06	1.44e-05	7.11 e-06	4.867	1.12e-06
VGG-19	***VGG-19* (*SMOTE*)**	1.20e-05	1.44e-05	8.92e-06	1.898	0.0576
VGG-19	***CNN* (*SMOTE*)**	1.20e-05	9.12e-06	6.25 e-06	-3.556	0.0003
***VGG-19* (*SMOTE*)**	***CNN* (*SMOTE*)**	1.44e-05	9.12e-06	6.79 e-06	-5.087	3.63e-07

Tendo em conta o nível de significância de 5%, todos os valores de *p-value* >0.05 não foram estatisticamente significativos. Neste caso, como estamos a comparar várias hipóteses é necessário aplicar a correção de Bonferroni para compensar o erro, como é descrito na Secção 6.3. Por este motivo é considerado que todos os valores de *p-value*>0.008 não são estatisticamente significativos. Analisando os resultados da tabela 12, observa-se que o resultado obtido pela comparação entre a *CNN* e *CNN* (*SMOTE*) não rejeita a hipótese nula, ou seja, não é possível concluir que existe uma diferença significativa para um nível de significância de 5%. O mesmo acontece com os modelos *VGG-19* e *VGG* (*SMOTE*). Isto significa que os resultados obtidos com a utilização do *SMOTE* por cada uma das abordagens não permitiu rejeitar a hipótese nula. No que diz respeito às restantes comparações podemos rejeitar a hipótese nula e concluir que existe uma diferença significativa da *AUC* entre os modelos da primeira abordagem (*VGG-19* e *VGG-19* (*SMOTE*)) e os da segunda abordagem (*CNN* e *CNN* (*SMOTE*)). Por este motivo, é possível concluir que existe uma diferença estatisticamente significativa entre as duas abordagens (para 5%).

7 Conclusão

Esta dissertação tem como objetivo o desenvolvimento de um sistema que permita a classificação de sons cardíacos utilizando técnicas de *deep learning*. Para o desenvolvimento deste sistema foram utilizados os conjuntos de dados PASCAL e *Physionet*, provenientes de dois concursos.

Para o desenvolvimento do trabalho que se descreve nesta dissertação, realizou-se a análise dos diversos artigos científicos existentes na literatura. A análise destes artigos permitiu a definição da estratégia de pré-processamento, extração de atributos e das arquiteturas que originaram as duas abordagens descritas neste projeto. A primeira abordagem consiste na aplicação de *fine-tunning* de modelos pré-treinados em imagens ou sons e a segunda abordagem na utilização de uma arquitetura *CNN*, abordadas na Secção 5.5. As metodologias de avaliação utilizadas foram *Hold-out* e *K-fold cross-validation*. Na aplicação destas metodologias foi necessário garantir que os sons pertencentes ao mesmo paciente constavam no conjunto de treino ou teste no momento de avaliação (nunca em ambos). Nestas metodologias, um paciente é classificado como anormal quando uma das amostras do mesmo paciente for considerada positiva, caso contrário é considerado normal.

De forma a ser possível avaliar o conjunto de testes de acordo com a metodologia definida anteriormente, foi necessário proceder à customização da função de treino dos modelos. Esta customização permitiu que todas as métricas obtidas nos conjuntos de testes em cada iteração de treino do modelo fossem por paciente e não por som.

As experiências realizadas nesta dissertação têm como objetivo identificar os melhores modelos a serem utilizados nas abordagens e assim criar a melhor abordagem tendo em conta a análise realizada. A primeira experiência consistiu na escolha do tamanho fixo da amostra a ser utilizada na etapa de pré-processamento. Esta experiência comparou as amostras de tamanho 4 a 7 utilizando a técnica de *padding* ou repetição do som.

A segunda experiência consistiu na escolha do melhor método de extração de atributos usando um tamanho de amostra fixo (6 segundos) definido na experiência anterior. Os métodos utilizados para esta comparação foram: *STFT Spectrogram*, *MFCC* e *Mel Spectrogram*. O *MFCC* sendo o método preferencial na classificação de sons cardíacos foi o método que alcançou piores resultados. O melhor resultado obtido foi pelo *Mel Spectrogram* com um F1 de 87.45% e uma *accuracy* de 91.36%.

Depois de selecionado o tamanho de amostra e o método de extração de atributos, a terceira experiência consistiu na comparação de modelos pré-treinados (segunda abordagem). O melhor resultado obtido foi pelo *VGG-19* com um *F1 score* de 86.20% e uma *accuracy* de 90.33%.

A experiência final consistiu na comparação das duas abordagens definidas inicialmente com o melhor resultado obtido por Nogueira *et al.* (2019). Nesta experiência, para cada uma das

abordagens foi aplicada a técnica de *oversampling SMOTE*. Esta técnica permitiu balancear o conjunto de treino das abordagens e assim criar uma alternativa extra para cada. Desta forma, para a comparação final foram utilizadas a *VGG-19* e *VGG-19* (*SMOTE*) da primeira abordagem e *CNN* e *CNN* (*SMOTE*) da segunda abordagem.

Para a realização da experiência final foi garantida a utilização da mesma metodologia de avaliação que Nogueira *et al.* (2019). O melhor *overall* obtido nesta experiência foi de 86.69% através do modelo *CNN* (*SMOTE*), superando assim, o resultado de 83.22% de Nogueira *et al.* (2019). A experiência é descrita na Secção 6.4.4.

Futuramente, para dar continuidade ao trabalho desenvolvido nesta dissertação seria um passo importante incluir novos conjuntos de sons de DCV que permitiriam a criação de modelos com uma maior diversidade de dados. No processo de extração de atributos poderia ser incluído a extração de *TF,* que possibilitaria a extração de mais informações sobre os sons ajudando na sua classificação (normal ou anormal). Por fim, poderiam ser aplicados métodos *ensemble* nos modelos criados de modo a tornara-los mais robustos, mais estáveis e assim provavelmente melhorar os resultados obtidos.

Referências

Adam Hayes (2020) *Bonferroni Test*. Available at: https://www.investopedia.com/terms/b/bonferroni-test.asp (Accessed: 24 June 2021).

Ahlstrom, C. *et al.* (2006) 'Feature Extraction for Systolic Heart Murmur Classification', *Annals of Biomedical Engineering*, 34(11), pp. 1666–1677. doi: 10.1007/s10439-006-9187-4.

Amodei, D. *et al.* (2015) 'Deep Speech 2: End-to-End Speech Recognition in English and Mandarin', *33rd International Conference on Machine Learning, ICML 2016*, 1, pp. 312–321. Available at: http://arxiv.org/abs/1512.02595.

Antiques (2019) *ANTIQUE STETHOSCOPES*. Available at: https://www.antiquemed.com/ (Accessed: 11 December 2020).

Azuaje, F. (2006) 'Witten IH, Frank E: Data Mining: Practical Machine Learning Tools and Techniques 2nd edition', *BioMedical Engineering OnLine*, 5(1), p. 51. doi: 10.1186/1475-925X-5-51.

B2b International (2014) *What is the Value Proposition Canvas?* Available at: https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/ (Accessed: 28 January 2021).

Baeldung (2020) *How ReLU and Dropout Layers*. Available at: https://www.baeldung.com/cs/ml-relu-dropout-layers (Accessed: 20 February 2021).

Banerjee, M. and Majhi, S. (2020) 'Multi-class heart sounds classification using 2D-convolutional neural network', in *Proceedings of the 2020 International Conference on Computing, Communication and Security, ICCCS 2020*. IEEE, pp. 1–6. doi: 10.1109/ICCCS49678.2020.9277204.

Basaveswara, S. K. (2019) *CNN Architectures, a Deep-dive*. Available at: https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049 (Accessed: 15 February 2021).

Boulares, M., Alafif, T. and Barnawi, A. (2020) 'Transfer Learning Benchmark for Cardiovascular Disease Recognition', *IEEE Access*, 8, pp. 109475–109491. doi: 10.1109/ACCESS.2020.3002151.

Bourouhou, A. *et al.* (2020) 'Heart Sound Signals Segmentation and Multiclass Classification', *International Journal of Online and Biomedical Engineering (iJOE)*, 16(15), p. 64. doi: 10.3991/ijoe.v16i15.16817.

Brownlee, J. (2019) *Convolutional Neural Network Model Innovations for Image Classification*. Available at: https://machinelearningmastery.com/review-of-architectural-innovations-for-convolutional-neural-networks-for-image-classification/ (Accessed: 28 January 2021).

Brownlee, J. (2020a) *How to Evaluate the Skill of Deep Learning Models*. Available at: https://machinelearningmastery.com/evaluate-skill-deep-learning-models/ (Accessed: 28 January 2021).

Brownlee, J. (2020b) *How to Fix k-Fold Cross-Validation for Imbalanced Classification*. Available at: https://machinelearningmastery.com/cross-validation-for-imbalanced-classification/ (Accessed: 28 January 2021).

Brownlee, J. (2020c) *SMOTE for Imbalanced Classification with Python*. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ (Accessed: 5 March 2021).

Casey Newman (2018) *What is a Value Proposition?* Available at: https://www.kunocreative.com/blog/good-value-proposition-examples (Accessed: 28 January 2021).

Chengyu Liu (2016) *Classification of Heart Sound Recordings - The PhysioNet Computing in Cardiology Challenge 2016*. Available at: https://physionet.org/content/challenge-2016/1.0.0/ (Accessed: 29 December 2020).

Chowdhury, T. H., Poudel, K. N. and Hu, Y. (2020) 'Time-Frequency Analysis, Denoising, Compression, Segmentation, and Classification of PCG Signals', *IEEE Access*, 8, pp. 160882–160890. doi: 10.1109/ACCESS.2020.3020806.

Cristian, E. *et al.* (1995) 'o estetoscópio', 4(3), pp. 112–118. Available at: https://seer.sis.puc-campinas.edu.br/seer/index.php/cienciasmedicas/article/download/1406/1380.

DeGroff, C. G. *et al.* (2001) 'Artificial Neural Network–Based Method of Screening Heart Murmurs in Children', *Circulation*, 103(22), pp. 2711–2716. doi: 10.1161/01.CIR.103.22.2711.

DeLong, E. R., DeLong, D. M. and Clarke-Pearson, D. L. (1988) 'Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach', *Biometrics*, 44(3), p. 837. doi: 10.2307/2531595.

Direção Geral de Saude (2016) *Doenças Cérebro-Cardiovasculares em Números*. Available at: https://www.dgs.pt/estatisticas-de-saude/estatisticas-de-saude/publicacoes/portugal-doencas-cerebro-cardiovasculares-em-numeros-2015-pdf.aspx (Accessed: 1 February 2021).

Eda Kavlakoglu (2020) *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* Available at: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks (Accessed: 2 January 2021).

Flamm, K. L. *et al.* (2020) 'Evidence-Based Assessment of the Heart and Circulatory System', in *Evidence-Based Physical Examination*. New York, NY: Springer Publishing Company. doi: 10.1891/9780826164544.0006.

Gavrovska, A. *et al.* (2013) 'Classification of prolapsed mitral valve versus healthy heart from phonocardiograms by multifractal analysis', *Computational and Mathematical Methods in Medicine*, 2013. doi: 10.1155/2013/376152.

Getz, P. B. G. N. M. C. S. M. R. (2012) *Classifying Heart Sounds Challenge*. Available at: http://www.peterjbentley.com/heartchallenge/.

Gomes, E. F. *et al.* (2013) 'Classifying heart sounds: Approaches to the PASCAL challenge', *HEALTHINF 2013 - Proceedings of the International Conference on Health Informatics*, pp. 337–340. doi: 10.5220/0004234403370340.

Gomes, E. F., Jorge, A. M. and Azevedo, P. J. (2013) 'Classifying heart sounds using multiresolution time series motifs: An exploratory study', *ACM International Conference Proceeding Series*, pp. 23–30. doi: 10.1145/2494444.2494458.

Gomes, E. F., Jorge, A. M. and Azevedo, P. J. (2014) 'Classifying heart sounds using SAX motifs, random forests and text mining techniques', *ACM International Conference Proceeding Series*, pp. 334–337. doi: 10.1145/2628194.2628240.

Gomes, E. F. and Pereira, E. (2012) 'Classifying heart sounds using peak location for segmentation and feature construction', *Aistats*, (1), pp. 1–5.

Goyal, K. (2021) *Top 10 Deep Learning Frameworks in 2021*. Available at: https://www.upgrad.com/blog/top-deep-learning-frameworks/ (Accessed: 2 January 2021).

Gupta, C. N. *et al.* (2007) 'Neural network classification of homomorphic segmented heart sounds', *Applied Soft Computing*, 7(1), pp. 286–297. doi: 10.1016/j.asoc.2005.06.006.

HUGHES, G. (1996) 'Turning new product development into a continuous learning process', *Journal of Product Innovation Management*, 13(2), pp. 89–104. doi: 10.1016/0737-6782(95)00112-3.

Hunter, J. (2021) *Matplotlib*. Available at: https://matplotlib.org/ (Accessed: 24 May 2021).

IBM Cloud Education (2020a) *Artificial Intelligence (AI)*. Available at: https://www.ibm.com/cloud/learn/what-is-artificial-intelligence (Accessed: 2 January 2021).

IBM Cloud Education (2020b) *Deep Learning*. Available at: https://www.ibm.com/cloud/learn/deep-learning (Accessed: 2 January 2021).

Jameslyons (2013) *Mel Frequency Cepstral Coefficient (MFCC)*. Available at: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/] (Accessed: 2 January 2021).

Jason Brownlee (2019) *What is Deep Learning?* Available at: https://machinelearningmastery.com/what-is-deep-learning/ (Accessed: 2 January 2021).

K.Raimi (2019) *10 CNN Architectures*. Available at: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#e4b1 (Accessed: 15 February 2021).

Keras (2019) *What's coming in TensorFlow 2.0*. Available at: https://blog.tensorflow.org/2019/01/whats-coming-in-tensorflow-2-0.html (Accessed: 28 January 2021).

Keras (2020) *Keras ResNet: Building, Training & Scaling Residual Nets on Keras*. Available at: https://missinglink.ai/guides/keras/keras-resnet-building-training-scaling-residual-nets-keras/ (Accessed: 28 January 2021).

Keras (2021) *Model training APIs*. Available at: https://keras.io/api/models/model_training_apis/ (Accessed: 24 May 2021).

Khan, K. N. *et al.* (2020) 'Deep Learning Based Classification of Unsegmented Phonocardiogram Spectrograms Leveraging Transfer Learning', *arXiv*. Available at: http://arxiv.org/abs/2012.08406.

Kishan Maladkar (2018) *6 Types of Artificial Neural Networks Currently Being Used in Machine Learning*. Available at: https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/ (Accessed: 2 January 2021).

Koen, P. A., Bertels, H. M. J. and Kleinschmidt, E. (2014) 'Managing the front end of innovation-part I: Results from a three-year study', *Research Technology Management*, 57(2), pp. 34–43. doi: 10.5437/08956308X5702145.

Koen, P. A., Bertels, H. M. J. and Kleinschmidt, E. J. (2014) 'Managing the front end of innovation-part II: Results from a three-year study', *Research Technology Management*, 57(3), pp. 25–35. doi: 10.5437/08956308X5703199.

Koike, T. *et al.* (2020) 'Audio for Audio is Better? An Investigation on Transfer Learning Models for Heart Sound Classification', in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, pp. 74–77. doi: 10.1109/EMBC44109.2020.9175450.

Kulkarni, K. A. (2021) *Tutorial — How to Install TensorFlow 2.0 on Ubuntu 20.04?* Available at: https://medium.com/swlh/tutorial-how-to-install-tensorflow-2-0-on-ubuntu-20-04-10cf6fe28de1 (Accessed: 28 January 2021).

Latif, S. *et al.* (2018) 'Phonocardiographic Sensing using Deep Learning for Abnormal Heartbeat Detection', *arXiv*, 18(22), pp. 9393–9400. Available at: http://arxiv.org/abs/1801.08322.

Liu, C. *et al.* (2016) 'An open access database for the evaluation of heart sound algorithms', *Physiological Measurement*, 37(12), pp. 2181–2213. doi: 10.1088/0967-3334/37/12/2181.

McFee, B. *et al.* (2015) 'librosa: Audio and Music Signal Analysis in Python', in *Proceedings of the 14th Python in Science Conference*, pp. 18–24. doi: 10.25080/Majora-7b98e3ed-003.

Mdsave (2021) *Electrocardiogram (EKG/ECG)*. Available at: https://www.mdsave.com/procedures/electrocardiogram-ekg-ecg/d182ff (Accessed: 28 January 2021).

Michael Copeland (2016) *What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning?* Available at: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/ (Accessed: 2 January 2021).

MissingLink (2020) *7 Types of Neural Network Activation Functions: How to Choose?* Available at: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/ (Accessed: 19 February 2021).

Mltut (2020) *K Fold Cross-Validation in Machine Learning? How does K Fold Work?* Available at: https://www.mltut.com/k-fold-cross-validation-in-machine-learning-how-does-k-fold-work/ (Accessed: 28 January 2021).

Mustafa, M. *et al.* (2020) 'Detection of heartbeat sounds arrhythmia using automatic spectral methods and cardiac auscultatory', *The Journal of Supercomputing*, 76(8), pp. 5899–5922. doi: 10.1007/s11227-019-03062-7.

Nogueira, D. M. *et al.* (2019) 'Classifying Heart Sounds Using Images of Motifs, MFCC and Temporal Features', *Journal of Medical Systems*, 43(6), p. 168. doi: 10.1007/s10916-019-1286-5.

Nogueira, D. M., Ferreira, C. A. and Jorge, A. M. (2017) 'Classifying Heart Sounds Using Images of MFCC and Temporal Features', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 186–203. doi: 10.1007/978-3-319-65340-2_16.

Noponen, A. L. *et al.* (2007) 'Phono-spectrographic analysis of heart murmur in children', *BMC Pediatrics*, 7, pp. 1–10. doi: 10.1186/1471-2431-7-23.

ODSC (2020) *5 Deep Learning Frameworks to Consider for 2020*. Available at: https://medium.com/@ODSC/5-deep-learning-frameworks-to-consider-for-2020-183e6c12cba9 (Accessed: 23 January 2021).

Oh, S. L. *et al.* (2020) 'Classification of heart sound signals using a novel deep WaveNet model', *Computer Methods and Programs in Biomedicine*, 196, p. 105604. doi: 10.1016/j.cmpb.2020.105604.

Oliveira, J. and Praca, I. (2021) 'On the Usage of Pre-Trained Speech Recognition Deep Layers to Detect

Emotions', *IEEE Access*, 9, pp. 9699–9705. doi: 10.1109/ACCESS.2021.3051083.

Opala, M. (2019) *Deep Learning Frameworks Tensorflow Comparison*. Available at: https://www.netguru.com/blog/deep-learning-frameworks-comparison (Accessed: 21 January 2021).

Papadaniil, C. D. and Hadjileontiadis, L. J. (2014) 'Efficient heart sound segmentation and extraction using ensemble empirical mode decomposition and kurtosis features', *IEEE Journal of Biomedical and Health Informatics*, 18(4), pp. 1138–1152. doi: 10.1109/JBHI.2013.2294399.

Paperspace (2021) *Paperspace*. Available at: www.paperspace.com (Accessed: 1 April 2021).

Pathmind (2020) *Comparison of AI Frameworks*. Available at: https://wiki.pathmind.com/comparison-frameworks-dl4j-tensorflow-pytorch#torch%0A (Accessed: 18 January 2021).

Pierre Carbonnelle (2021) *PopularitY of Programming Language*. Available at: https://pypl.github.io/PYPL.html (Accessed: 5 January 2021).

Potes, C. *et al.* (2016) 'Ensemble of Feature:based and Deep learning:based Classifiers for Detection of Abnormal Heart Sounds', in *Computing in Cardiology*. CCAL, pp. 621–624. doi: 10.22489/CinC.2016.182-399.

Prabhu (2018) *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. Available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (Accessed: 2 January 2021).

Prabinnepal (2020) *MobileNet Architecture Explained*. Available at: https://prabinnepal.com/mobilenet-architecture-explained/ (Accessed: 28 January 2021).

Prodoctor (2019) *Estetoscópio eletrônico: tudo o que você precisa saber*. Available at: https://prodoctor.net/blog/estetoscopio-eletronico/ (Accessed: 29 December 2020).

Python (2021) *Python*. Available at: https://www.python.org/download/releases/2.6.8/license/ (Accessed: 5 March 2021).

Rains, J. A. (2002) 'What are the functions of function analysis', pp. 225–232.

Roberts, L. (2020) *Understanding the Mel Spectrogram*. Available at: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (Accessed: 11 January 2021).

Russakovsky, O. *et al.* (2015) 'ImageNet Large Scale Visual Recognition Challenge', *International Journal of Computer Vision*, 115(3), pp. 211–252. doi: 10.1007/s11263-015-0816-y.

Saaty, R. W. (1987) 'The analytic hierarchy process—what it is and how it is used', *Mathematical Modelling*, 9(3–5), pp. 161–176. doi: 10.1016/0270-0255(87)90473-8.

Seeley, R. R., Stephens, T. D. and Tate, P. (2011) 'Aparelho circulatório-capítulo 20', in *Anatomia e Fisiologia 8º edição*. Loures-Portugal: Lusociência, pp. 697–739.

Shivanandhan, M. (2020) *Deep Learning Frameworks Compared: MxNet vs TensorFlow vs DL4j vs PyTorch*. Available at: https://www.freecodecamp.org/news/deep-learning-frameworks-compared-mxnet-vs-tensorflow-vs-dl4j-vs-pytorch/ (Accessed: 28 January 2021).

Silverman, B. and Balk, M. (2019) 'Digital Stethoscope—Improved Auscultation at the Bedside', *American Journal of Cardiology*, 123(6), pp. 984–985. doi: 10.1016/j.amjcard.2018.12.022.

Singh, M. and Cheema, A. (2013) 'Heart Sounds Classification using Feature Extraction of Phonocardiography Signal', *International Journal of Computer Applications*, 77(4), pp. 13–17. doi: 10.5120/13381-1001.

SINGH, S. A. and MAJUMDER, S. (2020) 'Short unsegmented PCG classification based on ensemble classifier', *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, 28(2), pp. 875–889. doi: 10.3906/elk-1905-165.

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

Team, D. S. (2020) *Guia simples para a terminologia da matriz de confusão*. Available at: https://datascience.eu/pt/matematica-e-estatistica/guia-simples-para-a-terminologia-da-matriz-de-confusao/ (Accessed: 28 January 2021).

Tensorflow (2021) *GradientTape*. Available at: https://www.tensorflow.org/api_docs/python/tf/GradientTape (Accessed: 24 May 2021).

TIOBE (2020) *TIOBE Index for December 2020*. Available at: https://www.tiobe.com/tiobe-index/ (Accessed: 5 January 2021).

Varghees, V. N. and Ramachandran, K. I. (2014) 'A novel heart sound activity detection framework for automated heart sound analysis', *Biomedical Signal Processing and Control*, 13, pp. 174–188. doi: 10.1016/j.bspc.2014.05.002.

Venkatachalam, M. (2019) *Recurrent Neural Networks*. Available at: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce (Accessed: 10 January 2021).

Wolfram (2018) *Deep Speech 2 Trained on Baidu English Data*. Available at: https://resources.wolframcloud.com/NeuralNetRepository/resources/Deep-Speech-2-Trained-on-Baidu-English-Data (Accessed: 24 May 2021).

World health organization (2021) *Cardiovascular diseases (CVDs)*, *Cardiovascular diseases (CVDs)*. Available at: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (Accessed: 30 November 2020).

Anexo A – *Analytic Hierarchy Process*: cálculos

De acordo com os critérios definidos na secção 3.1.4, foi desenvolvido a Tabela A.1 onde é definida uma matriz de comparação entre as mesmas.

Tabela A.1 - Matriz de comparação

	Aprendizagem	**Performance**	**Conhecimento**	**Popularidade**
Aprendizagem	1.00	0.33	0.14	0.33
Performance	3	1.00	0.33	2
Conhecimento	7	3	1.00	5
Popularidade	3	0.5	0.2	1.00
Soma	14	4.83	1.67	8.33

De seguida, é calculada a importância de cada atributo através do cálculo dos seus pesos. Para isso, inicialmente, é realizada a normalização dos dados, em que, cada um dos valores é dividido pela soma da sua respetiva coluna. De seguida, o peso é obtido através da média aritmética dos valores de cada linha. A Tabela A.2 demonstra o cálculo final do peso de cada atributo.

Tabela A.2 - Cálculo do peso

	Aprendizagem	**Performance**	**Conhecimento**	**Popularidade**	**Peso**
Aprendizagem	0.07	0.07	0.08	0.03	0.06
Performance	0.2	0.21	0.2	0.24	0.21
Conhecimento	0.5	0.62	0.6	0.6	0.58
Popularidade	0.2	0.1	0.12	0.12	0.14

A partir dos pesos obtidos, o conhecimento aparece em primeiro lugar com o valor de 0.58, seguido da performance com 0.21, a popularidade com 0.14 e, por fim, o conhecimento com o valor de 0.06. A próxima fase consiste no cálculo do teste de consistência (Tabela A.3), em que cada um dos critérios definidos inicialmente são multiplicados pelo peso, e de seguida, cada uma das linhas são somadas novamente dando origem à Soma Ponderada (SP). De seguida, o ***λmax*** é calculado através da média da soma ponderada a dividir pelo peso de cada linha (Tabela A.4.).

Tabela A.3 - Teste de consistência

	Aprendizagem	**Performance**	**Conhecimento**	**Popularidade**	**Soma ponderada**
Aprendizagem	0.06	0.07	0.08	0.05	0.26
Performance	0.18	0.21	0.19	0.28	0.86
Conhecimento	0.42	0.63	0.58	0.7	2.33
Popularidade	0.18	0.11	0.12	0.14	0.55

Tabela A.4 - Cálculo da soma ponderada a dividir pelo peso

	Soma ponderada	**Peso**	**SP/Peso**
Aprendizagem	0.26	0.06	4.3
Performance	0.86	0.21	4.1
Conhecimento	2.33	0.58	4.02
Popularidade	0.55	0.14	3.93

λmax = $(4.3 + 4.1 + 4.02 + 3.93) \div 4 = 4.09$

De seguida, o índice de consistência (IC) é calculado da seguinte forma:

IC= $(\lambda max\ - \mathrm{n}) \div (\mathrm{n} - 1) =\ (4.09 - 4)\ \div\ (4 - 1) =\ 0.03$

De acordo com a tabela ponderação de critérios, 4 critério corresponde a 0.90. Então a razão de consistência (RC) é calculada da seguinte forma.

RC=$IC \div 0.90 = 0.03/0.90 = 0.03$

Como o RC<0.10 então podemos assumir que os valores são consistentes.

Por fim, são construídas as tabelas de comparação para cada critério comparando cada uma das alternativas.

Tabela A.5 - Comparação da aprendizagem entre as *frameworks*

Aprendizagem	***Keras/Tensorflow***	***Pytorch***	***Caffe***	***CNTK***	**Peso**
Keras/Tensorflow	1	4	0.33	3	0.27
Pytorch	0.2	1	0.25	0.5	0.07
Caffe	3	5	1	4	0.53
CNTK	0.25	2	0.33	1	0.12
Soma	4.45	12	1.91	8.5	

Tabela A.6 -Comparação da *performance* entre as *frameworks*

Perforamance	***Keras/Tensorflow***	***Pytorch***	***Caffe***	***CNTK***	**Peso**
Keras/Tensorflow	1	4	5	3	0.53
Pytorch	0.2	1	3	2	0.22
Caffe	0.25	0.33	1	0.33	0.09
CNTK	0.33	0.5	3	1	0.16
Soma	1.78	5.83	12	6.33	

Tabela A.7 - Comparação do conhecimento entre as *frameworks*

Conhecimento	***Keras/Tensorflow***	***Pytorch***	***Caffe***	***CNTK***	**Peso**
Keras/Tensorflow	1	2	6	5	0.51
Pytorch	0.5	1	4	3	0.29
Caffe	0.16	0.25	1	0.5	0.08
CNTK	0.2	0.33	2	1	0.12
Soma	1.86	3.58	13	9.5	

Tabela A.8 - Comparação da popularidade entre as *frameworks*

Popularidade	***Keras/Tensorflow***	***Pytorch***	***Caffe***	***CNTK***	**Peso**
Keras/Tensorflow	1	2	4	6	0.48
Pytorch	0.5	1	4	3	0.28
Caffe	0.25	0.25	1	4	0.15
CNTK	0.16	0.33	0.25	1	0.08
Soma	1.91	3.58	9.5	14	

Por fim, os pesos das tecnologias são multiplicados pelos pesos dos critérios.

$Keras/Tensorflow = (0.27 \times 0.06) + (0.53 \times 0.21) + (0.51 \times 0.58) + (0.48 \times 0.14) = 0.49$

$Pytorch = (0.07 \times 0.06) + (0.22 \times 0.21) + (0.29 \times 0.58) + (0.28 \times 0.14) = 0.26$

$Caffe = (0.53 \times 0.06) + (0.09 \times 0.21) + (0.08 \times 0.58) + (0.15 \times 0.14) = 0.12$

$CNTK = (0.12 \times 0.06) + (0.16 \times 0.21) + (0.12 \times 0.58) + (0.08 \times 0.14) = 0.12$