

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROJETO DE GRADUAÇÃO**



**AMANDA SCHNEIDER BASSANI**

**USO DE UMA REDE NEURAL HÍBRIDA PARA  
AUXÍLIO NO DIAGNÓSTICO DE DEPRESSÃO A  
PARTIR DE SINAIS DE ELETROENCEFALOGRAMA**

VITÓRIA - ES

JULHO/2023

Amanda Schneider Bassani

**USO DE UMA REDE NEURAL HÍBRIDA PARA  
AUXÍLIO NO DIAGNÓSTICO DE DEPRESSÃO A  
PARTIR DE SINAIS DE ELETROENCEFALOGRAMA**

Parte manuscrita do Projeto de Graduação da aluna Amanda Schneider Bassani, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Vitória - ES

Julho/2023

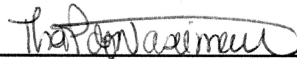
Amanda Schneider Bassani

## **USO DE UMA REDE NEURAL HÍBRIDA PARA AUXÍLIO NO DIAGNÓSTICO DE DEPRESSÃO A PARTIR DE SINAIS DE ELETROENCEFALOGRAMA**

Parte manuscrita do Projeto de Graduação da aluna Amanda Schneider Bassani, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenharia Eletricista.

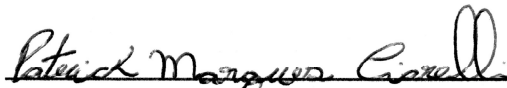
Aprovado em 20 de julho de 2023.

### **COMISSÃO EXAMINADORA:**

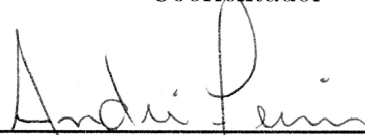


**Profa. Dra. Thaís Pedruzzi do  
Nascimento**

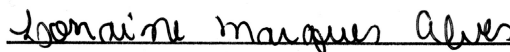
Universidade Federal do Espírito Santo  
Orientadora



**Prof. Dr. Patrick Marques Ciarelli**  
Universidade Federal do Espírito Santo  
Coorientador



**Prof. Dr. André Ferreira**  
Universidade Federal do Espírito Santo  
Examinador



**Dra. Lorraine Marques Alves**  
Examinadora

Vitória - ES

Julho/2023

## AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas especiais que participaram da minha vida enquanto eu trilhava essa jornada acadêmica e que permitiram que eu continuasse focada e dedicada aos meus estudos. A todos vocês, minha eterna gratidão.

A Deus, por permitir que várias oportunidades surgissem na minha vida e me guiar pelos caminhos que escolhi percorrer.

Aos meus pais, Carlos e Edna, pelo apoio e paciência. Sei que esses períodos finais não foram fáceis, principalmente por perdas de pessoas insubstituíveis nas nossas vidas.

À minha família que sempre acreditou em meu potencial, incentivando a minha evolução.

Ao meu namorado Gabriel por todo apoio emocional, tornando meus finais de semana tranquilos depois de semanas agitadas.

Aos meus amigos, Matheus e Igor, que não desistiram de mim e estiveram do meu lado ao longo do curso.

A todos os meus colegas de faculdade por tornarem a rotina universitária mais divertida.

À minha orientadora Thaís Pedruzzi e coorientador Patrick Ciarelli por toda disponibilidade, orientação e dedicação durante o desenvolvimento deste trabalho.

À banca examinadora pela aceitação do convite e pelo tempo investido para leitura e avaliação desse trabalho.

Agradeço à Universidade Federal do Espírito Santo pela minha formação.

## RESUMO

A depressão é um transtorno mental caracterizado pela dificuldade de concentração, baixa autoestima e perda de interesse. Atualmente, seu diagnóstico é feito de forma subjetiva, em que o paciente precisa dar respostas fiéis à realidade para um diagnóstico confiável. Este trabalho visa utilizar sinais de EEG, obtidos a partir de duas bases de dados públicas, como entrada para uma rede neural híbrida rasa composta por duas camadas convolucionais e uma recorrente. A rede construída tem como finalidade auxiliar no diagnóstico de depressão e garantir uma alta assertividade nos seus resultados. Como os sinais de EEG são extremamente sensíveis a ruídos, foram realizadas etapas de pré-processamento para facilitar a extração de características feitas pelas camadas convolucionais. O modelo obtido alcançou 98,3% de acurácia na primeira base de dados e 98,8% na segunda base de dados utilizada. A fim de validar o modelo, os resultados deste trabalho foram comparados com os resultados de trabalhos que construíram redes de mesma complexidade. Além disso, é feita uma comparação com outros modelos para expor a melhoria em relação ao tempo computacional do treinamento do modelo usado.

**Palavras-chave:** Depressão; Redes neurais convolucionais; Redes recorrentes; EEG.

## ABSTRACT

Depression is a mental illness related to lack of concentration, low self-esteem, and loss of interest. Currently, its diagnosis is subjective, in which the patient needs to adequately answer a number of questions for a reliable diagnosis. This work aims to use EEG signals, obtained from two public datasets, as input for a shallow hybrid neural network composed of two convolutional and one recurrent layers. This network has the purpose to assist in the diagnosis of depression and guarantee assertive results. As the signal is extremely sensitive to noise, a preprocessing step was required to facilitate the feature extraction made by convolutional layers. The obtained model reached 98.3% of accuracy in the first dataset and 98.8% in the second dataset. To validate the model, the results of this work were compared with the results of works that built networks of the same complexity. In addition, a comparison is made with other models to expose the improvement in relation to the computational training time of the used model.

**Keywords:** Depression; Convolutional neural network; Recurrent network; EEG.

## LISTA DE FIGURAS

Figura 1 – Estrutura de um neurônio . . . . .	17
Figura 2 – Vista superior das posições segundo o sistema 10-20 . . . . .	17
Figura 3 – Distâncias entre os eletrodos . . . . .	18
Figura 4 – Exemplo de uma mistura de sinais: um show musical com duas fontes sonoras, $s_1(t)$ e $s_2(t)$ , e dois microfones, $x_1(t)$ e $x_2(t)$ . Os áudios capturados pelos microfones são as misturas dos áudios emitidos pelas fontes sonoras. . . . .	19
Figura 5 – Artefato de movimento dos olhos . . . . .	21
Figura 6 – Artefato de ranger de mandíbula . . . . .	21
Figura 7 – Artefato de batimento cardíaco . . . . .	22
Figura 8 – Funções de janela . . . . .	23
Figura 9 – Estrutura de uma rede CNN-1D . . . . .	26
Figura 10 – Estrutura de uma célula GRU . . . . .	27
Figura 11 – Fluxograma da abordagem . . . . .	29
Figura 12 – Dados no domínio do tempo filtrados . . . . .	32
Figura 13 – Dados no domínio da frequência filtrados . . . . .	33
Figura 14 – Posição dos eletrodos Fp2, F8 e O1 . . . . .	34
Figura 15 – Exemplo de artefato da base de dados 1 . . . . .	35
Figura 16 – Exemplo de artefato da base de dados 2 . . . . .	35
Figura 17 – Exemplo de remoção de artefatos . . . . .	36
Figura 18 – Periodogramas de uma janela de 1 segundo de uma pessoa depressiva e de uma pessoa saudável . . . . .	38
Figura 19 – Estrutura de uma matriz de confusão . . . . .	40
Figura 20 – Arquitetura adotada . . . . .	42
Figura 21 – Resultados de treinamento para a base de dados 1 . . . . .	45
Figura 22 – Matriz de confusão obtida para a base de dados 1 . . . . .	46
Figura 23 – Comparativo entre modelos com respeito a base de dados 1 . . . . .	46
Figura 24 – Resultados de treinamento para a base de dados 2 . . . . .	47
Figura 25 – Matriz de confusão para a base de dados 2 . . . . .	48
Figura 26 – Comparativo entre modelos com respeito a base de dados 2 . . . . .	49

## LISTA DE TABELAS

Tabela 1 – Ajuste dos parâmetros de rede . . . . .	42
Tabela 2 – Ajuste dos parâmetros de treinamento . . . . .	43
Tabela 3 – Distribuição das amostras . . . . .	44
Tabela 4 – Métricas do modelo na base de dados 1 . . . . .	44
Tabela 5 – Métricas do modelo na base de dados 2 . . . . .	48



## LISTA DE QUADROS

Quadro 1 – Bibliotecas do Python utilizadas no projeto . . . . .	30
Quadro 2 – Especificações da máquina do Google Colab . . . . .	30

## LISTA DE ABREVIATURAS E SIGLAS

AU	<i>Arbitrary Unit</i>
CNN	<i>Convolutional Neural Network</i>
DBN	<i>Deep Belief Network</i>
DFT	<i>Discrete Fourier Transform</i>
DL	<i>Deep Learning</i>
EC	<i>Eyes Closed</i>
EEG	Eletroencefalograma
FastICA	<i>Fast Independent Component Analysis</i>
FC	<i>Fully Connected</i>
FIR	<i>Finite Impulse Response</i>
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
ICA	<i>Independent Component Analysis</i>
IIR	<i>Infinite Impulse Response</i>
LOO	<i>Leave-One-Out</i>
LSTM	<i>Long Short-Term Memory</i>
MDD	<i>Major Depressive Disorder</i>
ML	<i>Machine Learning</i>
MODMA	<i>Multi-modal Open Dataset for Mental-disorder Analysis</i>
PCA	<i>Principal Component Analysis</i>
PSD	<i>Power Spectral Density</i>
ReLU	<i>Rectified Linear Unit</i>
RNN	<i>Recurrent Neural Network</i>

SVM *Support Vector Machine*

UFES Universidade Federal do Espírito Santo

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos</b>	<b>14</b>
<b>1.2</b>	<b>Estrutura do Texto</b>	<b>15</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>16</b>
<b>2.1</b>	<b>Sinais de EEG</b>	<b>16</b>
<b>2.2</b>	<b>Filtros digitais</b>	<b>18</b>
<b>2.3</b>	<b>Análise de Componentes Independentes</b>	<b>19</b>
<b>2.4</b>	<b>Método de Welch</b>	<b>20</b>
<b>2.5</b>	<b>Redes Neurais</b>	<b>23</b>
2.5.1	Redes Convolucionais Unidimensionais	24
2.5.2	GRU	27
<b>3</b>	<b>METODOLOGIA</b>	<b>29</b>
<b>3.1</b>	<b>Bases de dados</b>	<b>30</b>
<b>3.2</b>	<b>Pré-processamento</b>	<b>31</b>
3.2.1	Filtragem	32
3.2.2	Remoção de Artefatos	34
3.2.3	Extração de Características	36
<b>3.3</b>	<b>Métricas</b>	<b>37</b>
<b>3.4</b>	<b>Criação do Modelo</b>	<b>40</b>
<b>4</b>	<b>RESULTADOS</b>	<b>44</b>
<b>4.1</b>	<b>Base de dados 1</b>	<b>44</b>
<b>4.2</b>	<b>Base de dados 2</b>	<b>47</b>
<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>50</b>
<b>5.1</b>	<b>Conclusão</b>	<b>50</b>
<b>5.2</b>	<b>Trabalhos futuros</b>	<b>50</b>
	<b>REFERÊNCIAS</b>	<b>52</b>

# 1 INTRODUÇÃO

A depressão é uma doença mental relacionada à falta de concentração, baixa autoestima e perda de interesse, podendo levar a pensamentos suicidas em casos mais graves. Estudos feitos pelo Instituto de Métricas e Avaliação em Saúde da Universidade de Washington, apontaram que cerca de 280 milhões de pessoas sofriam de depressão ao redor do mundo em 2019 (IHME, 2019). Conforme a Organização Mundial de Saúde, esse número aumentou 25% no primeiro ano de pandemia, sendo mulheres e jovens entre 20 e 24 anos os grupos mais afetados (WHO, 2022). O principal fator que contribuiu para esse aumento foi a necessidade de isolamento social, já que as pessoas se sentiam sozinhas e com medo de serem infectadas pelo vírus da Covid-19.

Atualmente, o diagnóstico de doenças mentais é feito de maneira subjetiva em entrevistas com psiquiatras e questionários respondidos pelo paciente (SHARMA; JOSHI, 2021). Considerando a relevância dos problemas causados pela depressão, a busca de informações mais objetivas para o resultado do diagnóstico tem se intensificado na biomedicina. O uso de aprendizado de máquina (ML, do inglês *Machine Learning*) a fim de obter resultados precisos tem sido cada vez mais visado, principalmente a partir de informações adquiridas de sinais de eletroencefalograma (EEG). Esse tipo de dado é frequentemente escolhido porque apresenta características específicas conforme a doença analisada. O uso de sinais de EEG é também bastante relevante por possuir uma riqueza de informações em períodos relativamente curtos de tempo. Além disso, sua aquisição é menos custosa, quando comparada à aquisição de imagens de ressonância magnética (CUKIC; LOPEZ; PAVON, 2020).

Nos algoritmos que utilizam técnicas clássicas de ML para esse diagnóstico, existem etapas de pré-processamento, que muitas vezes tornam-se tarefas complicadas. Assim, surge como alternativa o uso de aprendizado profundo (DL, do inglês *Deep Learning*) para reduzir a complexidade do estágio de pré-processamento e obter melhores resultados (DING et al., 2020). Estratégias que envolvem o uso de DL com sinais de EEG também são utilizadas para diagnóstico de outras doenças mentais, como o transtorno bipolar (LAKSSHMAN et al., 2017) e esquizofrenia (FU et al., 2021).

Um dos problemas enfrentados por autores que usam DL como ferramenta para o desenvolvimento de seus trabalhos é a implementação dos métodos criados. À medida que os autores se esforçam para aprimorar ao máximo a precisão de seus modelos, as arquiteturas se tornam cada vez mais complexas, o que acaba dificultando a aplicação prática desses métodos devido ao alto custo computacional envolvido. Dessa forma, há estudos atualmente focados na busca de arquiteturas mais simplificadas com alta assertividade

nos resultados para auxiliar no diagnóstico da doença (SAFAYARI; BOLHASANI, 2021).

Cai et al. (2016) encontraram seu melhor resultado combinando o uso de uma rede de crença profunda (DBN, do inglês *Deep Belief Network*) com a potência absoluta da banda beta, alcançando 78,24% de acurácia ao utilizar uma grande quantidade de características e 3 eletrodos para gravação dos sinais de EEG. Já Mao et al. (2018) alcançaram 77,20% de acurácia construindo uma rede neural convolucional (CNN, do inglês *Convolutional Neural Network*) e usando o método de projeção não-distancial. Os autores Acharya et al. (2018) e Ay et al. (2019) também adotaram o uso das CNNs, porém utilizaram apenas sinais vindos do hemisfério direito do cérebro. Enquanto no primeiro trabalho foi obtido 96% de acurácia utilizando uma arquitetura de CNNs, no segundo foi alcançada uma acurácia de 99,12% ao combinar essa arquitetura com uma rede de memória de curto prazo longa (LSTM, do inglês *Long Short-Term Memory*), porém a eliminação de artefatos foi feita manualmente<sup>1</sup>.

Wajid e Abdul (2019) propuseram a utilização de uma rede CNN unidimensional com método de extração *end-to-end* obtendo 98,32% de acurácia. Já Qayyum, Razzak e Mumtaz (2020) combinaram CNN com uma rede de unidade recorrente fechada (GRU, do inglês *Gated Recurrent Unit*), alcançando 96,87% de acurácia para sinais cerebrais captados de pessoas de olhos fechados, usando a função de ativação sigmoide na camada de classificação, e 98,87% de acurácia usando um classificador SVM (*Support Vector Machine*). Já Wang et al. (2022) utilizaram sinais cerebrais captados de pessoas com olhos fechados e obtiveram 96% de acurácia juntando uma CNN-1D com uma GRU, mas adicionando uma camada de *Attention*<sup>2</sup> aumentaram essa acurácia para 99,33%. Saeedi et al. (2021) fizeram uma comparação entre modelos híbridos: o primeiro modelo combinava uma LSTM com uma rede CNN unidimensional, enquanto o segundo combinou uma LSTM com uma rede CNN bidimensional. A maior acurácia foi obtida no primeiro modelo com o valor de 99,24%.

O presente trabalho visa combinar uma CNN unidimensional com uma GRU. A escolha de uma CNN se baseia em seu enorme poder de extração de características. A opção por uma CNN unidimensional é explicada pela redução de operações de convolução e *pooling* em comparação com arquiteturas CNN bidimensionais, além de retornarem bons resultados em dados sequenciais (KIRANYAZ et al., 2021). A combinação com uma rede recorrente é preferida devido à sua capacidade de considerar características temporais

<sup>1</sup> A eliminação manual de artefatos é feita por especialistas em neurologia (SAFAYARI; BOLHASANI, 2021).

<sup>2</sup> A camada de *Attention* tem a finalidade de atribuir diferentes pesos na rede para focar nas características com maiores pesos e valorizá-las como as características mais importantes do sinal (WANG et al., 2022).

dos dados. A justificativa para o uso de uma unidade de recorrência GRU é por possuir menos parâmetros em comparação com uma unidade LSTM. A GRU possui duas portas<sup>3</sup> enquanto a LSTM possui três, sendo uma das portas da GRU uma combinação de duas portas da LSTM. Espera-se, com este trabalho, obter um modelo com alta acurácia, pois essas arquiteturas foram as que mais contribuíram para melhorar os resultados ao analisar os trabalhos citados e suas comparações propostas. Além disso, o trabalho visa explicar detalhadamente o algoritmo utilizado para ser possível sua compreensão e replicação<sup>4</sup>.

## 1.1 Objetivos

### Objetivo Geral

- O objetivo deste trabalho é o desenvolvimento de um algoritmo usando técnicas de processamento de sinais e uma rede neural híbrida rasa que auxilie no diagnóstico de depressão e garanta uma alta assertividade dos resultados, mantendo baixo custo computacional.

### Objetivos Específicos

- Estudar os métodos de pré-processamento de sinais biológicos e a teoria relacionada à construção de redes neurais;
- Validar e testar diferentes modelos: o primeiro composto apenas com camadas CNN unidimensionais, o segundo composto apenas com GRU, o terceiro composto apenas com LSTM, o quarto combinando CNN unidimensional com LSTM e o quinto combinando CNN unidimensional com GRU;
- Comparar o modelo proposto e seus resultados com outros modelos similares da literatura.

<sup>3</sup> As portas, também chamadas de *gates*, são mecanismos que controlam o fluxo de informação na rede. Esses *gates* regulam a informação que entra, sai ou permanece na célula de memória, permitindo que esta guarde ou descarte informações ao longo do tempo.

<sup>4</sup> O projeto se encontra disponível em <<https://github.com/amandasbassani/DepDetect>>.

## 1.2 Estrutura do Texto

O presente trabalho está estruturado da seguinte maneira:

- **Introdução:** este capítulo inicial contextualiza o leitor em relação à motivação das escolhas do tema e dos métodos que serão utilizados;
- **Referencial Teórico:** neste capítulo é apresentada a base teórica utilizada para elaboração do projeto;
- **Metodologia:** neste capítulo são apresentadas as etapas de desenvolvimento do projeto;
- **Resultados:** neste capítulo são dispostos os resultados obtidos após o desenvolvimento do projeto;
- **Conclusão:** neste capítulo é feita a conclusão do trabalho e são sugeridos temas para trabalhos futuros.



## 2 REFERENCIAL TEÓRICO

Neste capítulo são explicados os detalhes teóricos dos elementos que compõem o trabalho. Inicia-se com os sinais de EEG, que constituem os dados a serem analisados. Na sequência, são explorados e explicados os conceitos referentes às etapas do algoritmo desenvolvido: o pré-processamento (filtros, técnica de remoção dos artefatos e método de extração das características) e a teoria de redes neurais artificiais.

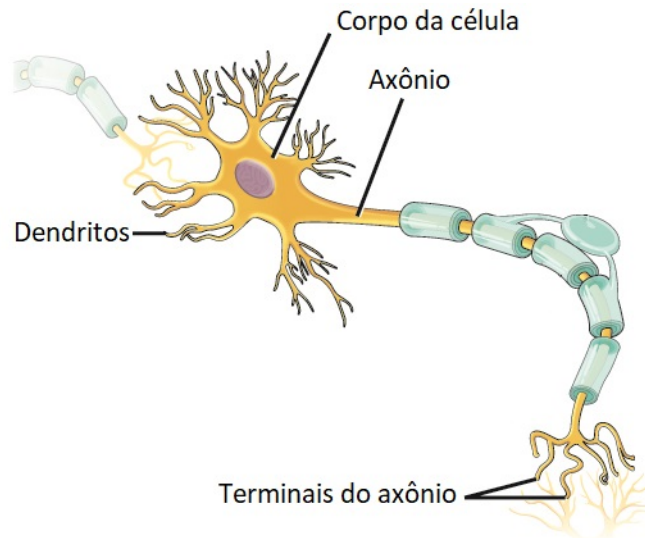
### 2.1 Sinais de EEG

O primeiro conceito de eletroencefalograma surgiu em 1875, quando o cientista britânico Richard Caton posicionou dois eletrodos de um galvanômetro sobre a cabeça de alguns animais e registrou as suas atividades cerebrais na forma de sinais elétricos. Mas foi em 1924 que o cientista Hans Berger registrou atividades cerebrais em humanos (ATTAR, 2022). Atualmente, a eletroencefalografia é um método usado como exame complementar de diagnósticos. Por consistir em um método não invasivo e permitir a medição de sinais sinápticos após sua ocorrência em resoluções milimétricas (CAI et al., 2020), é compreensível que os sinais de eletroencefalografia sejam o tipo de dado mais escolhido pelos pesquisadores em estudos relacionados a doenças mentais (SIULY; LI; ZHANG, 2016).

Para entender como surgem os sinais de EEG, primeiro é necessário conhecer a estrutura básica responsável por sua geração: o neurônio. A Figura 1 mostra a estrutura do neurônio composta por 3 partes principais. Basicamente, o corpo da célula guarda o núcleo e as organelas, os dendritos recebem os impulsos elétricos e o axônio garante a condução do impulso elétrico. A comunicação entre os neurônios permite a medição dos sinais de EEG, os quais são da ordem de  $\mu V$  quando obtidos no couro cabeludo (SANEI; CHAMBERS, 2013).

Ao analisar o sinal de EEG, é possível identificar 5 principais ondas cerebrais, as quais são diferenciadas através da sua banda de frequência. Essas bandas são chamadas de delta ( $<4\text{Hz}$ ), theta (4-8Hz), alpha (8-12Hz), beta (12-30Hz) e gamma (30-80Hz) (JIA; KOHN, 2011). Além disso, existe um método reconhecido internacionalmente que descreve as localizações em que devem ser colocados os eletrodos para a captação de sinais de EEG, que é mostrado na Figura 2. Esse sistema é chamado sistema 10-20, em que os números 10 e 20 se referem a distâncias específicas ao longo da linha média sagital do escalpo,

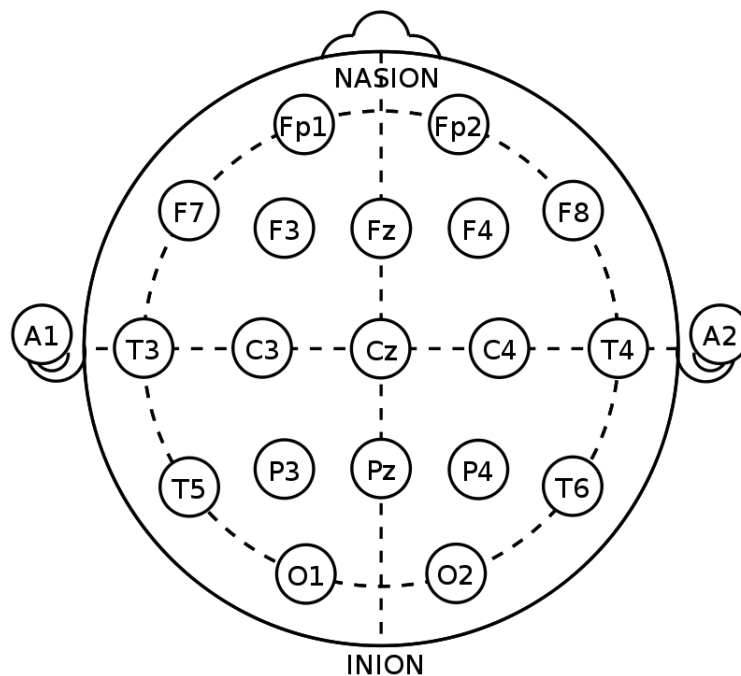
Figura 1 – Estrutura de um neurônio



Fonte: Traduzido e adaptado de Connie et al. (2016).

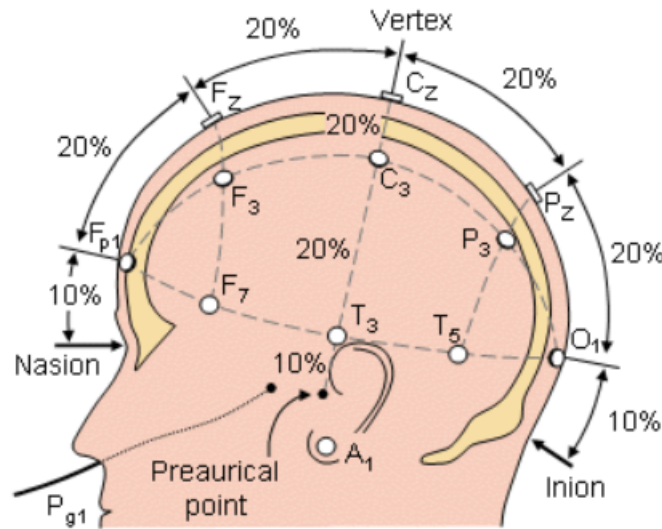
começando no ponto naso-ínion (a distância entre o nariz e o ponto mais proeminente na parte posterior da cabeça). Essas distâncias são divididas em intervalos iguais e os eletrodos são colocados nesses pontos, como mostra a Figura 3.

Figura 2 – Vista superior das posições segundo o sistema 10-20



Fonte: McCallan et al. (2021).

Figura 3 – Distâncias entre os eletrodos



Fonte: Malmivuo e Plonsey (1995).

## 2.2 Filtros digitais

Os filtros digitais são dispositivos eletrônicos projetados para processar sinais digitais, como áudio, imagens ou sinais elétricos, visando remover ou modificar determinadas características desses sinais. Esses filtros podem ser classificados em dois tipos principais: FIR (do inglês *Finite Impulse Response*) e IIR (do inglês *Infinite Impulse Response*).

Um filtro FIR é caracterizado por uma resposta ao impulso de duração finita, o que significa que sua saída depende apenas de uma soma ponderada dos valores de entrada passados. Já um filtro IIR possui uma resposta ao impulso de duração infinita, de forma que sua saída depende tanto dos valores de entrada passados quanto dos valores de saída passados. Os filtros IIR podem ter uma resposta de fase não linear, o que pode introduzir distorções de fase no sinal de saída, ao contrário dos filtros FIR. No entanto, eles geralmente requerem menos coeficientes do que os filtros FIR para obter a mesma característica de filtragem, resultando em menor consumo de recursos computacionais (FERREIRA, 2008).

Dessa forma, o filtro IIR foi escolhido, já que as distorções de fase introduzidas por este tipo de filtro não são críticas para a abordagem utilizada neste trabalho. A função de transferência do filtro digital IIR é apresentada na Equação 1.

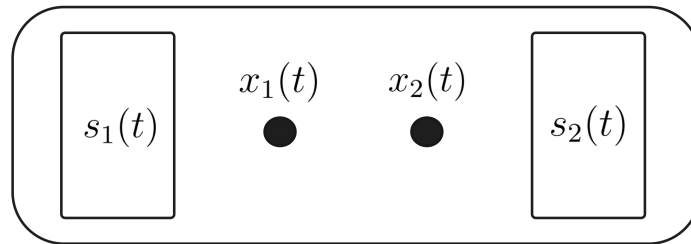
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{l=0}^M b_l z^{-l}}{1 + \sum_{k=1}^N a_k z^{-k}}, \quad (1)$$

em que  $Y(z)$  representa a transformada  $z$  da saída do filtro,  $X(z)$  representa a transformada  $z$  da entrada do filtro,  $b_l$  e  $a_k$  são os coeficientes de recursão,  $M$  é a ordem do polinômio dos zeros (número de coeficientes no numerador) e  $N$  é a ordem do polinômio dos polos (número de coeficientes no denominador).

### 2.3 Análise de Componentes Independentes

A Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*) é uma técnica utilizada para separar e identificar diferentes fontes de um sinal misturado. Um sinal misturado pode ocorrer, por exemplo, na situação em que duas pessoas estão em um festival de música onde ocorrem simultaneamente dois shows diferentes. Cada pessoa tem um microfone que recebe as gravações  $x_1(t)$  e  $x_2(t)$  e os shows são as fontes dos sons  $s_1(t)$  e  $s_2(t)$ , como mostra a Figura 4.

Figura 4 – Exemplo de uma mistura de sinais: um show musical com duas fontes sonoras,  $s_1(t)$  e  $s_2(t)$ , e dois microfones,  $x_1(t)$  e  $x_2(t)$ . Os áudios capturados pelos microfones são as misturas dos áudios emitidos pelas fontes sonoras.



Fonte: Produção da própria autora.

Assim, cada microfone receberá uma soma ponderada dos sinais emitidos pelas fontes, como mostram as Equações 2 e 3 de forma simplificada. Os parâmetros  $a_{11}$ ,  $a_{12}$ ,  $a_{21}$  e  $a_{22}$  dependem da distância dos microfones em relação às fontes. Quando os shows estão acontecendo, cada microfone captura uma mistura dos sons provenientes dos dois shows. A ICA funciona encontrando combinações lineares das amostras de áudio capturadas pelos microfones que sejam estatisticamente independentes entre si (HYVÄRINEN; OJA, 2000). Ao aplicar essa análise aos sinais misturados, é possível reconstruir as fontes originais, isolando e permitindo que cada pessoa escute isoladamente o seu show de interesse.

$$x_1(t) = a_{11}s_1 + a_{12}s_2 \quad (2)$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2 \quad (3)$$

Antes de aplicar a técnica de ICA, é importante realizar um pré-processamento dos dados. Primeiro, é realizado um pré-branqueamento dos dados. Uma das estratégias para essa etapa é a divisão dos dados de cada canal pelo desvio padrão ( $\sigma$ ) entre as amostras de todos os canais. O cálculo do desvio padrão é mostrado na Equação 4, onde  $x_i$  é o valor de cada amostra,  $n$  é o total de amostras e  $\mu$  é a média dos valores das amostras.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}} \quad (4)$$

Após o pré-branqueamento, é aplicado o método de Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*), o qual procura combinações lineares das variáveis originais para representar os dados baseando-se na correlação (WOLD; ESBENSEN; GELADI, 1987), de forma a descorrelacionar os dados na nova representação. Assim, são passados para o algoritmo de ICA apenas os primeiros componentes principais, ordenados por sua relevância, encontrados no PCA<sup>1</sup>. A ICA consegue encontrar independências estatísticas presentes nos componentes principais e cria componentes independentes.

De forma análoga, a ICA pode ser aplicada no processamento de sinais biológicos. O sinal de EEG sem tratamento é sensível a vários ruídos, principalmente por ser um sinal de baixa potência. Esses ruídos, também chamados de artefatos, podem ser causados tanto por fontes externas quanto por fontes do próprio corpo humano. A ICA permite que esses artefatos sejam isolados em componentes independentes, sendo possível eliminá-los do sinal original. Alguns dos principais artefatos que afetam a integridade do sinal de EEG são movimento dos olhos, ranger de mandíbula e batimento cardíaco (FERREIRA, 2008). A topografia e as componentes independentes desses artefatos estão apresentadas nas Figuras 5, 6 e 7.

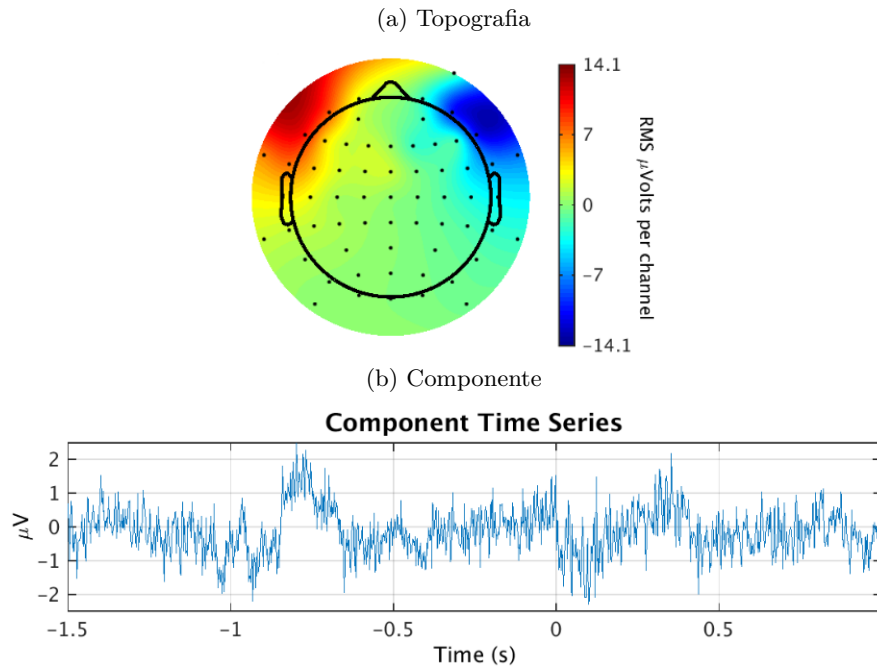
## 2.4 Método de Welch

O método de Welch foi elaborado por Welch (1967) e permite obter uma estimativa confiável das densidades espectrais (PSDs, do inglês *Power Spectral Densities*), que informam como a potência média está distribuída com relação à frequência de um dado sinal. O interessante desse método é que ele permite reduzir a variância<sup>2</sup> das PSDs estimadas. Em vez

<sup>1</sup> O número de componentes principais que serão usados como entrada para a ICA depende da aplicação e suas respectivas limitações.

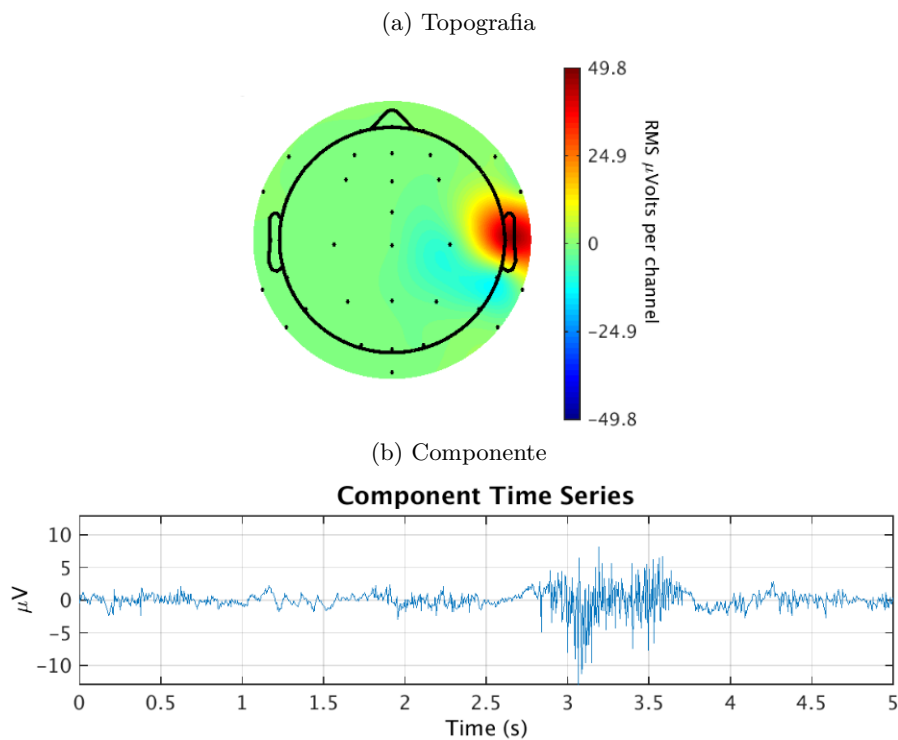
<sup>2</sup> A variância é o quadrado do desvio padrão, que por sua vez é calculado segundo a Equação 4.

Figura 5 – Artefato de movimento dos olhos



Fonte: Adaptado de Pion-Tonachini, Kreutz-Delgado e Makeig (2019).

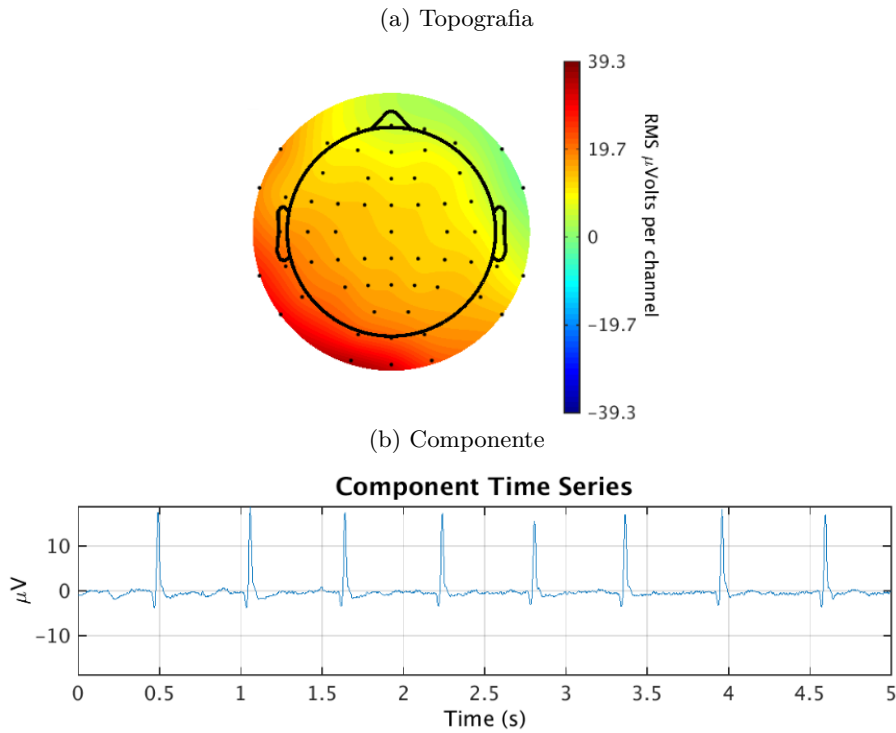
Figura 6 – Artefato de ranger de mandíbula



Fonte: Adaptado de Pion-Tonachini, Kreutz-Delgado e Makeig (2019).

de calcular o periodograma do sinal inteiro, o método apresenta etapas precedentes, descritas nos parágrafos subsequentes. O resultado é chamado de periodograma modificado, pois o gráfico resultante possui linhas mais suaves que um periodograma tradicional.

Figura 7 – Artefato de batimento cardíaco



Fonte: Adaptado de Pion-Tonachini, Kreutz-Delgado e Makeig (2019).

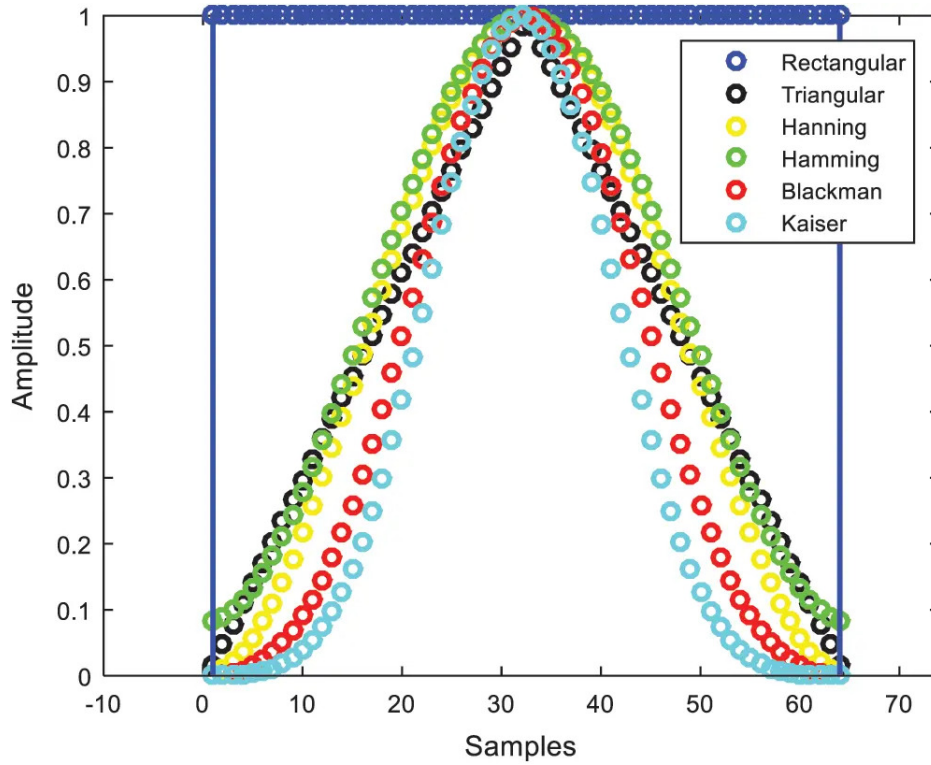
Considere que uma amostra  $X(n)$ ,  $n = 0, \dots, N - 1$ , pertence a um sinal amostrado no tempo e esse sinal é dividido em  $K$  segmentos,  $X_1(n), \dots, X_i(n), \dots, X_K(n)$ ,  $n = 0, \dots, L - 1$ . Pode-se escolher se o segmento  $X_i$ ,  $i = 2, \dots, K$  é obtido ao se sobrepor em uma parcela do segmento  $X_{i-1}$  ou não. Em seguida, é aplicada uma janela  $W(n)$ ,  $n = 0, \dots, L - 1$  a cada segmento para minimizar a distorção introduzida nas bordas do segmento durante o processo de segmentação. Algumas funções de janela são apresentadas na Figura 8.

A Transformada Discreta de Fourier (DFT, do inglês *Discrete Fourier Transform*), expressa na Equação 5, é aplicada, onde  $n$  representa o índice no domínio do tempo,  $k$  representa o índice no domínio da frequência,  $j$  é a unidade imaginária  $\sqrt{-1}$  e  $L$  é o tamanho de cada segmento.

$$A_i(k) = \sum_{n=0}^{L-1} X_i(n)W(n)e^{-j2\pi nk/L} \quad (5)$$

Finalmente, é calculado o periodograma modificado de cada segmento janelado, conforme a Equação 6, onde  $f_k$  é a frequência normalizada em relação à taxa de amostragem pela Equação 7 e  $U$  é uma constante dada pela Equação 8.

Figura 8 – Funções de janela



Fonte: Jwo, Chang e Wu (2021).

$$I_i(f_k) = \frac{1}{U} |A_i(k)|^2 \quad i = 1, \dots, K \quad (6)$$

$$f_k = k/L \quad k = 0, \dots, L/2 \quad (7)$$

$$U = \sum_{n=0}^{L-1} W^2(n) \quad (8)$$

As PSDs estimadas são as médias dos periodogramas, como mostra a Equação 9.

$$\hat{P}(f_k) = \frac{1}{K} \sum_{i=1}^K I_i(f_k) \quad (9)$$

## 2.5 Redes Neurais

O método de classificação desenvolvido é baseado em redes neurais, particularmente nas redes convolucionais e nas redes recorrentes. Nesta seção são apresentados conceitos teóricos de tais arquiteturas.



### 2.5.1 Redes Convolucionais Unidimensionais

A Rede Neural Convolutiva (CNN, do inglês *Convolutional Neural Network*) é usada em diversos campos da área de aprendizado de máquina, sendo uma das arquiteturas mais representativas no segmento de aprendizado profundo (WANG et al., 2022). Esses tipos de redes, como o próprio nome já diz, usam a operação de convolução entre duas funções, descrita na sua forma contínua na Equação 10. A função  $x$  é chamada de entrada e a função  $w$  é tipicamente chamada de *kernel*.

$$s(t) = \int x(a)w(t-a)da = (x * w)(t) \quad (10)$$

A convolução possui uma forma discreta, mostrada na Equação 11. Como os dados utilizados são provenientes de um sinal amostrado, essa será a operação de convolução utilizada nos algoritmos desenvolvidos.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (11)$$

As CNN são formadas por camadas do tipo convolucionais e do tipo *pooling*. Na camada convolutiva existem filtros que recebem o sinal de entrada e extraem características. As características que serão obtidas ao final dessa camada dependem dos pesos dos *kernels* de cada filtro, os quais percorrem todos os dados da entrada realizando sucessivas operações de convolução. A saída da camada convolutiva é alterada por uma função de ativação, geralmente não linear. Dentre as funções de ativação mais conhecidas na literatura estão a sigmoide, a tangente hiperbólica e a Unidade Linear Retificada (ReLU, do inglês *Rectified Linear Unit*), as quais estão descritas nas Equações 12, 13 e 14, respectivamente.

Já na camada de *pooling*, os dados são divididos em grupos e apenas um valor de cada grupo é extraído, podendo ser a média, a soma ou o valor máximo do grupo. A camada de *pooling* permite a compressão das características, resultando na redução dos dados de treinamento (WANG et al., 2022).

Ao final de uma CNN, são adicionadas camadas totalmente conectadas (FC, do inglês *Fully Connected*). Enquanto as camadas convolucionais são responsáveis pela extração de características, as camadas totalmente conectadas são responsáveis pela classificação dessas características extraídas. Uma camada totalmente conectada é composta por um conjunto de neurônios artificiais que estão conectados a todas as saídas da camada ante-

rior. Para os dados serem passados corretamente de uma camada de convolução para uma camada FC, é necessária uma camada de *flatten*, que realiza o achatamento dos dados das saídas dos filtros em um único vetor unidimensional. Por fim, a saída da camada FC passa por uma função de ativação que atuará como classificador. Além da já citada função sigmoide, que é usada para tarefas de classificação binária, existe a função softmax que generaliza a função sigmoide para  $n$  classes, mostrada na Equação 15 (GOODFELLOW; BENGIO; COURVILLE, 2016b). As informações construtivas de uma rede, como o número de *kernels*, tamanho do *kernel*, tamanho do grupo usado na camada de *pooling* e função de ativação, são chamadas de hiperparâmetros de rede.

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (12)$$

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1. \quad (13)$$

$$\text{ReLU}(x) = \begin{cases} x, & \text{se } x > 0 \\ 0, & \text{se } x \leq 0 \end{cases} \quad (14)$$

$$\text{softmax}(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad j = 1, \dots, K \quad (15)$$

Para treinar uma rede CNN, o algoritmo de *backpropagation* é usualmente empregado. Esse algoritmo possui duas fases: a propagação e a retro propagação. A princípio, os pesos da rede devem ser iniciados, geralmente com valores aleatórios. Na fase de propagação, os dados da entrada passam pela rede, sendo convoluídos com os *kernels* e reduzida sua dimensão pela operação de *pooling*. Assim, obtêm-se os primeiros valores preditos ao final da rede e estes são comparados com os valores verdadeiros dos rótulos usando uma função de perda. No contexto de classificação binária, pode-se utilizar a entropia cruzada binária<sup>3</sup>, mostrada na Equação 16, em que  $N$  representa o número de amostras,  $y_i$  representa o valor de rótulo verdadeiro e  $\hat{y}_i$  representa o valor predito.

$$\text{BCE} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (16)$$

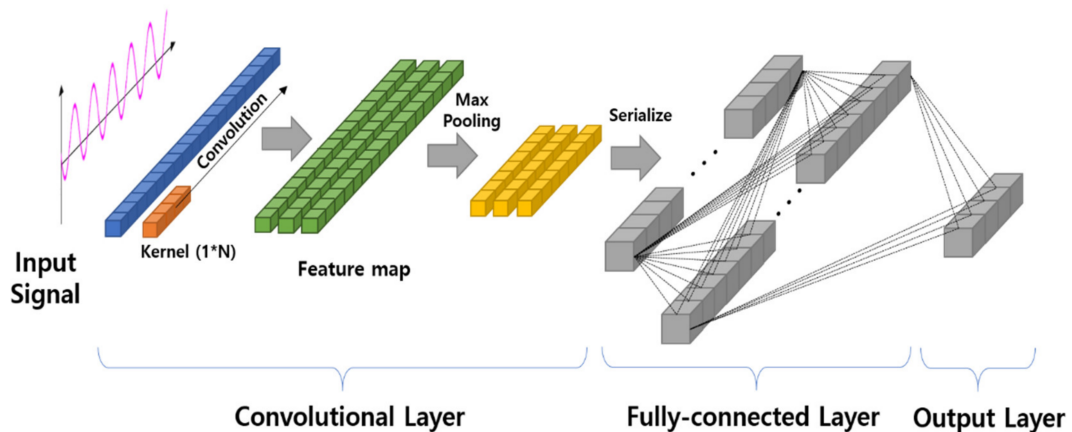
<sup>3</sup> A entropia cruzada binária é um caso particular da entropia cruzada, também chamada de *log loss*, sendo frequentemente usada como função de perda em redes neurais (Pedregosa et al., 2011a).

Quando a função de perda é igual a zero, significa que todos os valores preditos são iguais aos rótulos. Dessa forma, busca-se minimizá-la o quanto for possível. Assim, inicia-se a fase de retro propagação, onde as derivadas parciais das funções de perda são calculadas camada a camada em relação aos pesos até a camada de entrada. Ao final, tem-se uma série de gradientes que indicam a contribuição de cada peso da rede para a função de perda. Nesse momento, é usado um algoritmo de otimização, comumente chamado de otimizador, para que ele percorra em direção oposta ao gradiente com um passo chamado de taxa de aprendizagem e estabeleça novos valores para os pesos, finalizando a primeira iteração do *backpropagation* (GOODFELLOW; BENGIO; COURVILLE, 2016a).

Essa iteração não precisa necessariamente usar o conjunto de dados completo. Em vez disso, os dados podem ser divididos em lotes menores, chamados de *batches*. A divisão em *batches* contribui com o aumento da frequência em que os pesos são ajustados, o que pode ajudar a obter melhores resultados. Além disso, permite uma redução na demanda de memória, tornando o algoritmo mais eficiente computacionalmente. Quando todos os *batches* são apresentados ao *backpropagation* significa que o treinamento realizou uma época. O tamanho dos *batches* e a quantidade total de épocas do treinamento são hiperparâmetros de treinamento.

Apesar da versão bidimensional das CNN ser frequentemente usada, principalmente para o uso de imagens como dados de entrada, a sua versão unidimensional é preferível no processamento de sinais de séries temporais (GOODFELLOW; BENGIO; COURVILLE, 2016b). Existem dois tipos de camadas em uma rede convolucional unidimensional. As operações de convolução unidimensional, *pooling* e ativação são realizadas nas camadas convolucionais (*convolutional layer*). Já a camada totalmente conectada (*fully-connected layer*) é densa e permite concentrar os dados das camadas anteriores no final da rede. A Figura 9 mostra um exemplo de arquitetura CNN-1D.

Figura 9 – Estrutura de uma rede CNN-1D



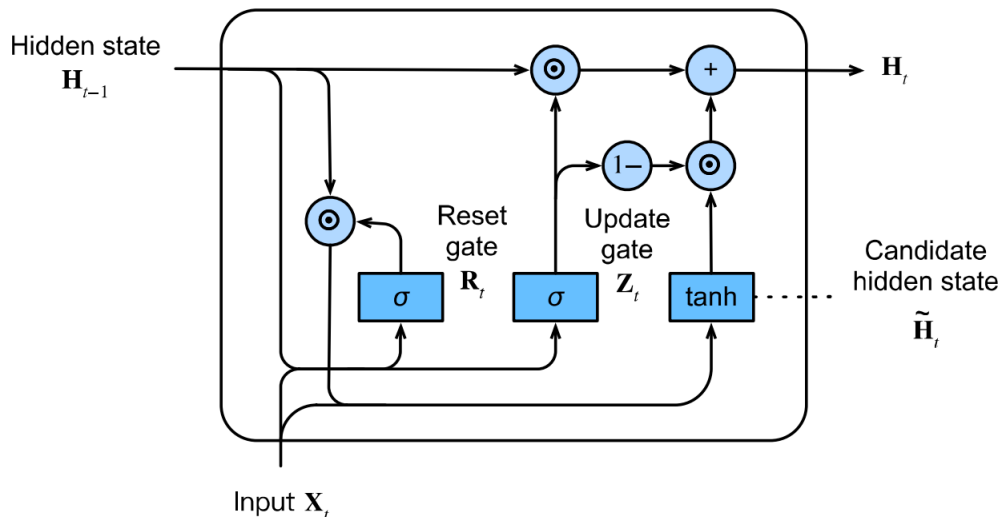
Fonte: Kim, Geem e Han (2020).

## 2.5.2 GRU

A GRU (*Gated Recurrent Unit*) é uma rede recorrente introduzida por Cho et al. (2014). Redes recorrentes têm como especialidade lidar com sinais sequenciais no tempo, enquanto as redes convolucionais lidam melhor com entradas de dados que se organizam espacialmente (GOODFELLOW; BENGIO; COURVILLE, 2016b). Porém, esse tipo de rede possui problemas de propagação dos gradientes por muitos estágios. Os pesos, dependendo de suas magnitudes, começam a tender para zero ou até mesmo para infinito, em razão da característica não-linear das funções de ativação utilizadas.

Diante desse problema, foram criadas as *gated RNN* (*Recurrent Neural Network*). Esse tipo de arquitetura tornou os modelos sequenciais mais efetivos para sequências temporais por possuir dependências de longo alcance em razão da introdução de recorrências internas. Essa ideia foi introduzida na arquitetura LSTM (HOCHREITER; SCHMIDHUBER, 1997), que posteriormente serviu de base para o desenvolvimento da GRU. As GRU quando comparadas às redes LSTM apresentam um *gate* a menos, tendo apenas o *reset gate* e o *update gate* e, como consequência, possuem menos parâmetros a serem ajustados. O *reset gate* controla o quanto de informação do estado anterior deverá ser lembrado e o *update gate* controla o quanto de informação do novo estado será copiado do estado anterior. A Figura 10 esboça o fluxo de informações na célula GRU, o qual será descrito nos parágrafos subsequentes.

Figura 10 – Estrutura de uma célula GRU



Fonte: Zhang et al. (2021).

Ao descrever o funcionamento de uma célula GRU para um determinado instante de tempo  $t$ , Zhang et al. (2021) consideram uma entrada  $\mathbf{X}_t$  de tamanho  $n \times d$ , onde  $n$  é o número de amostras e  $d$  é o número de canais. Além disso, consideram o estado anterior

$\mathbf{H}_{t-1}$  de tamanho  $n \times h$ , onde  $h$  é o número de unidades ocultas. As Equações 17 e 18 descrevem a saída do *reset gate* e do *update gate*, respectivamente. As matrizes  $\mathbf{W}_{xr}$ ,  $\mathbf{W}_{xz}$  de tamanho  $d \times h$  e  $\mathbf{W}_{hr}$ ,  $\mathbf{W}_{hz}$  de tamanho  $h \times h$  são matrizes de pesos e  $\mathbf{b}_r$ ,  $\mathbf{b}_z$  de tamanho  $1 \times h$  são os valores de *bias*. A função sigmoide ( $\sigma$ ) é usada para delimitar os valores no intervalo de 0 a 1.

$$\mathbf{R}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xr} + \mathbf{H}_{t-1} \mathbf{W}_{hr} + \mathbf{b}_r). \quad (17)$$

$$\mathbf{Z}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xz} + \mathbf{H}_{t-1} \mathbf{W}_{hz} + \mathbf{b}_z). \quad (18)$$

Antes de encontrar o novo estado  $\mathbf{H}_t$ , existe a etapa de encontrar um candidato a novo estado  $\tilde{\mathbf{H}}_t$ , descrito na Equação 19. As matrizes  $\mathbf{W}_{xh}$  com tamanho  $d \times h$  e  $\mathbf{W}_{hh}$  com tamanho  $h \times h$  são matrizes de pesos e  $\mathbf{b}_h$  de tamanho  $1 \times h$  são os valores de *bias*. A função tangente hiperbólica (*tanh*) é usada como função de ativação e  $\odot$  representa a operação matricial elemento-a-elemento. O candidato é avaliado por meio do *update gate*  $\mathbf{Z}_t$  e a Equação 20 representa o novo estado  $\mathbf{H}_t$ .

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{hh} + \mathbf{b}_h). \quad (19)$$

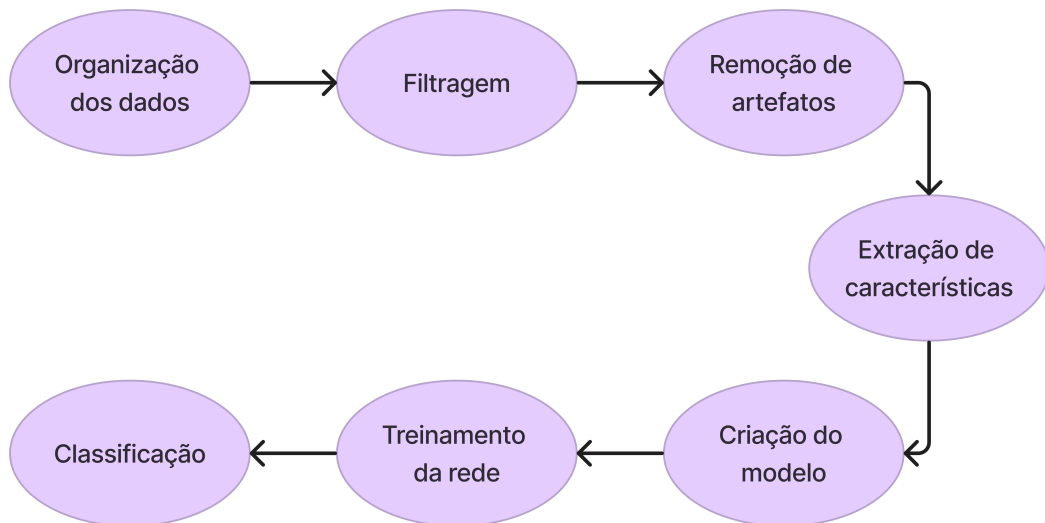
$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t. \quad (20)$$

### 3 METODOLOGIA

Adotou-se a metodologia proposta por Wang et al. (2022), porque o método desenvolvido pelos autores apresenta uma arquitetura de poucas camadas que consegue gerar resultados competitivos com o estado-da-arte. Entretanto, uma vez que os autores não explicitaram algumas etapas, foi necessário adaptá-las. Ao longo deste capítulo, tais etapas são explicadas.

O fluxograma da abordagem usada neste trabalho é mostrado na Figura 11. A organização dos dados será explicada em detalhes na Seção 3.1. A filtragem, a remoção de artefatos e a extração de características serão explicadas na Seção 3.2. A criação do modelo é descrita na Seção 3.4.

Figura 11 – Fluxograma da abordagem



Fonte: Produção da própria autora.

Foram usados dois conjuntos de dados públicos a fim de validar a abordagem. A linguagem utilizada para o desenvolvimento de todos os códigos foi Python por ser uma linguagem de programação de código aberto com bibliotecas especializadas em processamento de sinais. O Quadro 1 mostra as bibliotecas utilizadas no projeto e suas respectivas versões. O Google Colab<sup>1</sup> foi utilizado para realizar a etapa de treinamento e as especificações da máquina estão presentes no Quadro 2.

<sup>1</sup> O Colab permite que qualquer pessoa escreva e execute código Python por meio do navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação (GOOGLE, 2022).

Quadro 1 – Bibliotecas do Python utilizadas no projeto

Biblioteca	Versão
matplotlib	3.7.1
mne	1.4.0
numpy	1.23.5
scikit_learn	1.2.2
scipy	1.10.1
tensorflow	2.12.0

Fonte: Produção da própria autora.

Quadro 2 – Especificações da máquina do Google Colab

Componente	Descrição
Sistema operacional	Linux
Versão do sistema operacional	5.15.107+
Distribuição	Ubuntu
Versão da distribuição	20.04
Processador	Intel(R) Xeon(R) CPU @ 2.20GHz
Número de núcleos físicos	6
RAM do sistema	83.5 GB
Disco	166.8 GB
Placa de vídeo	NVIDIA A100-SXM-40GB

Fonte: Produção da própria autora.

### 3.1 Bases de dados

As bases de dados escolhidas possuem duas classes: uma para pacientes diagnosticados com Transtorno Depressivo Maior (MDD, do inglês *Major Depressive Disorder*)<sup>2</sup> e uma para pacientes considerados saudáveis.

**Base de dados 1.** Na coleção de dados MODMA (*Multi-modal Open Dataset for Mental-disorder Analysis*) (CAI et al., 2020) existem bases de dados relativas a 4 experimentos no total, sendo 3 deles gravações de EEG e 1 gravação de áudio. A base de dados escolhida entre as disponíveis foi a *EEG\_128channels\_resting\_lanzhou\_2015*, que contém o experimento realizado com os participantes em *resting state*<sup>3</sup>. Os dados foram coletados pelo Laboratório Provincial de Computação Vestível de Gansu na Universidade de Lanzhou na China. O número total de participantes do experimento foi 53, sendo 24 participantes diagnosticados com depressão por métodos tradicionais (13 homens e 11 mulheres, média

<sup>2</sup> MDD é um dos diagnósticos clínicos específicos mais comuns entre os tipos de depressão existentes (American Psychiatric Association, 2023).

<sup>3</sup> Os participantes foram orientados a permanecerem acordados de olhos fechados e evitarem qualquer movimentação desnecessária.

das idades igual a 30,88) e 29 considerados saudáveis (20 homens e 9 mulheres, média das idades igual a 31,45). O tempo de gravação foi de 5 minutos, a frequência de amostragem foi de 250 Hz e a impedância de cada eletrodo foi mantida abaixo de 50 k $\Omega$ . As gravações foram feitas usando 128 canais usando a referência de vértex comum (Cz).

**Base de dados 2.** A coleção de dados *MDD Patients and Healthy Controls EEG Data* (MUMTAZ, 2016) possui bases de dados relativas a 3 experimentos de gravações de EEG. Os dados foram coletados no ambulatório do Hospital da Universidade de Ciências da Malásia. A base de dados escolhida também foi a de *resting state*, a qual era diferenciada das demais bases através da sigla EC (*eyes closed*) incluída nos nomes de seus arquivos de gravação. Essa base possui 63 gravações no total, 33 diagnosticados com MDD (18 mulheres e 15 homens, média das idades igual a 40,33) e 30 saudáveis (9 mulheres e 21 homens, média das idades igual a 38,23) (MUMTAZ et al., 2017). As gravações foram feitas usando 19 canais usando a referência de orelhas conectadas. Os dados já se encontravam filtrados em 50 Hz (frequência da rede elétrica do local) utilizando um filtro *notch*, em 0,5 utilizando um filtro passa alta e em 80 Hz utilizando um filtro passa baixa. O tempo de gravação foi de aproximadamente 5 minutos e a frequência de amostragem foi de 256 Hz.

As bases incluem diferentes canais e referências. Portanto, foram selecionados apenas 16 canais de interesse (Fp1/2, F3/4, C3/4, P3/4, O1/2, F7/8, T3/4, T5/6) já usados em estudos de diagnóstico de depressão (Akdemir Akar et al., 2015) e a segunda base de dados foi re-referenciada para o canal Cz. Além disso, foram selecionadas apenas as gravações que possuem 5 minutos completos. Assim, o número de sujeitos na segunda base de dados passou a ser 45, sendo 19 com depressão e 26 saudáveis.

Os dados foram armazenados em *arrays 3-D*, sendo a primeira dimensão o número de indivíduos, a segunda dimensão o número de canais e a terceira dimensão o número de observações no tempo. Para a primeira base de dados esse *array 3-D* possui o formato de  $53 \times 16 \times 75000$ , enquanto para a segunda base de dados esse *array 3-D* possui o formato de  $45 \times 16 \times 76800$ .

### 3.2 Pré-processamento

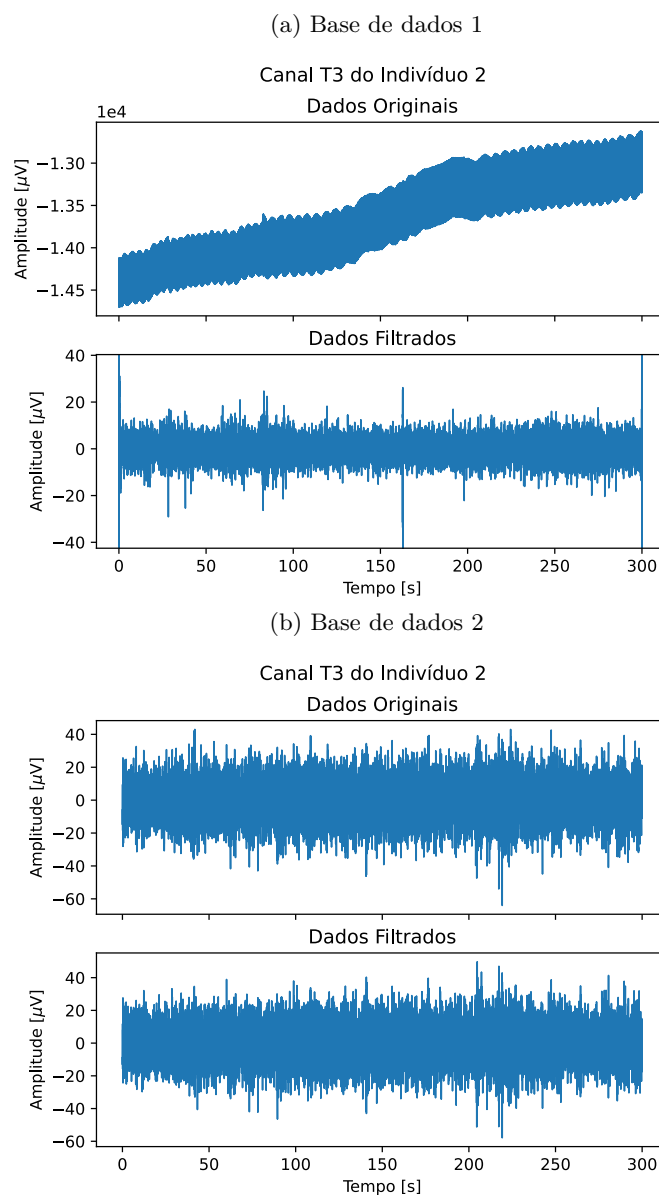
Esta seção apresenta os estágios de pré-processamento que foram necessários para garantir a extração de informações do modelo construído: filtragem, remoção de artefatos e extração de características.



### 3.2.1 Filtragem

Para a primeira base de dados foi usado um filtro notch em 50 Hz, já que essa é a frequência de rede de onde foram feitas as gravações. Um filtro Butterworth de 4ª ordem foi utilizado nas duas bases de dados para manter a banda entre as frequências 0,5 Hz e 50 Hz. A biblioteca utilizada para essa etapa foi a *Scipy* (VIRTANEN et al., 2020). As Figuras 12a e 12b mostram o antes e depois dos dados no domínio do tempo das bases de dados 1 e 2, respectivamente.

Figura 12 – Dados no domínio do tempo filtrados



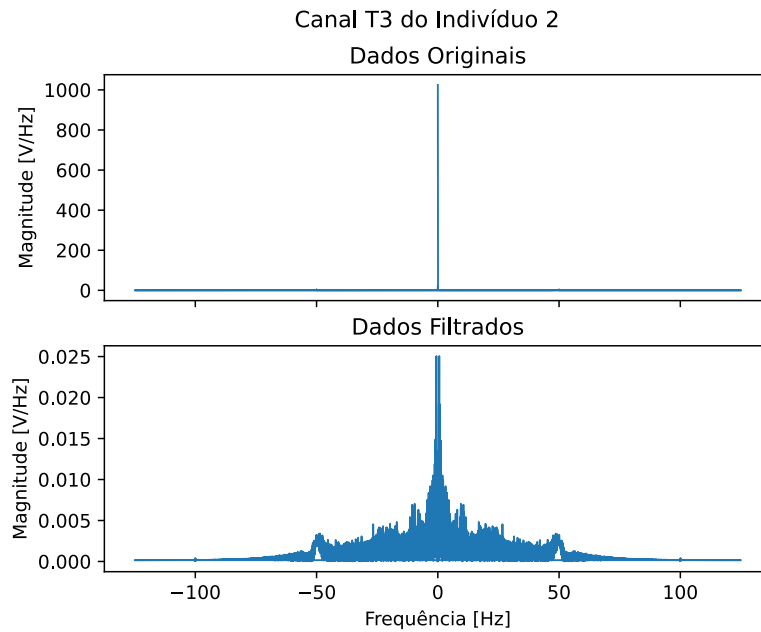
Fonte: Produção da própria autora.

Para verificar se as frequências foram apropriadamente filtradas, faz-se necessário uma análise no domínio da frequência, por meio de uma DFT (*Discrete Fourier Transform*).

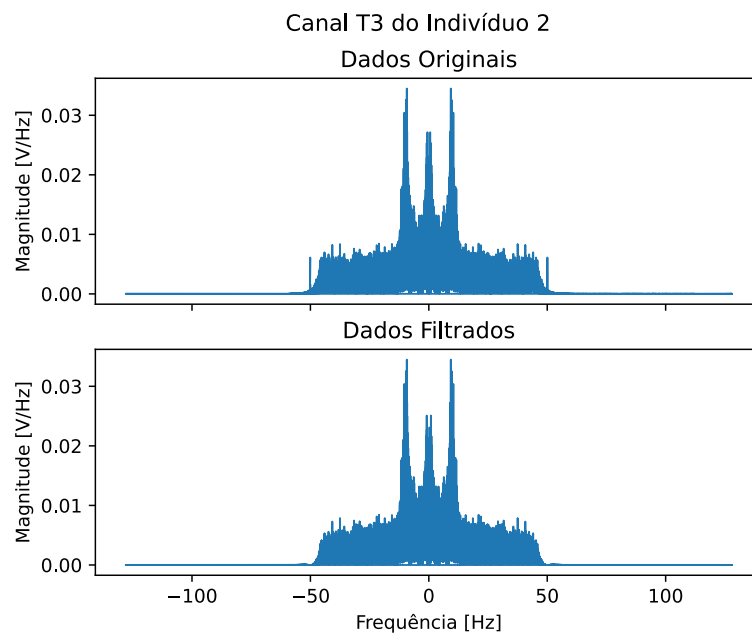
As Figuras 13a e 13b mostram o antes e depois dos dados no domínio da frequência das bases de dados 1 e 2, respectivamente. É possível observar o quão poluída está a base de dados 1, principalmente pela alta magnitude da componente contínua (0 Hz).

Figura 13 – Dados no domínio da frequência filtrados

(a) Base de dados 1



(b) Base de dados 2

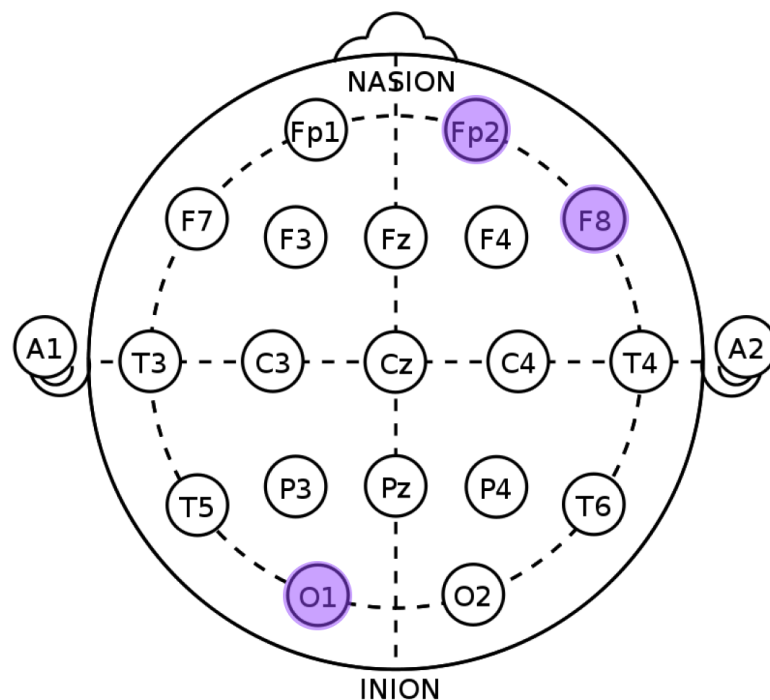


Fonte: Produção da própria autora.

### 3.2.2 Remoção de Artefatos

Na etapa de remoção de artefatos foi usada a biblioteca *MNE-Python* (Gramfort et al., 2013) para aplicar a técnica de pré-processamento ICA. As componentes independentes de cada indivíduo foram salvas utilizando a *FastICA*<sup>4</sup>. Foi usada a função *find\_bads\_eog*, também da biblioteca *MNE-Python* (Gramfort et al., 2013) para detectar automaticamente artefatos nos sinais já filtrados. Essa função possui um parâmetro *ch\_name*, no qual é informado o nome do canal usado para a detecção de artefatos relacionados ao movimento ocular. Nesse trabalho, foram removidos apenas os artefatos relacionados ao movimento ocular. Por meio de uma inspeção visual dos sinais filtrados, foi verificado que 3 canais estavam fortemente influenciados por artefatos: Fp2, F8 e O1, destacados na Figura 14. Os canais Fp2 e F8, por estarem na região frontal perto dos olhos, são bastante afetados por artefatos de movimento ocular. Já o canal O1 está na região occipital e é responsável pela atividade dos olhos. Pequenos movimentos oculares podem ocorrer involuntariamente mesmo com os olhos fechados, resultando em artefatos (SINGH; WAGATSUMA, 2017).

Figura 14 – Posição dos eletrodos Fp2, F8 e O1



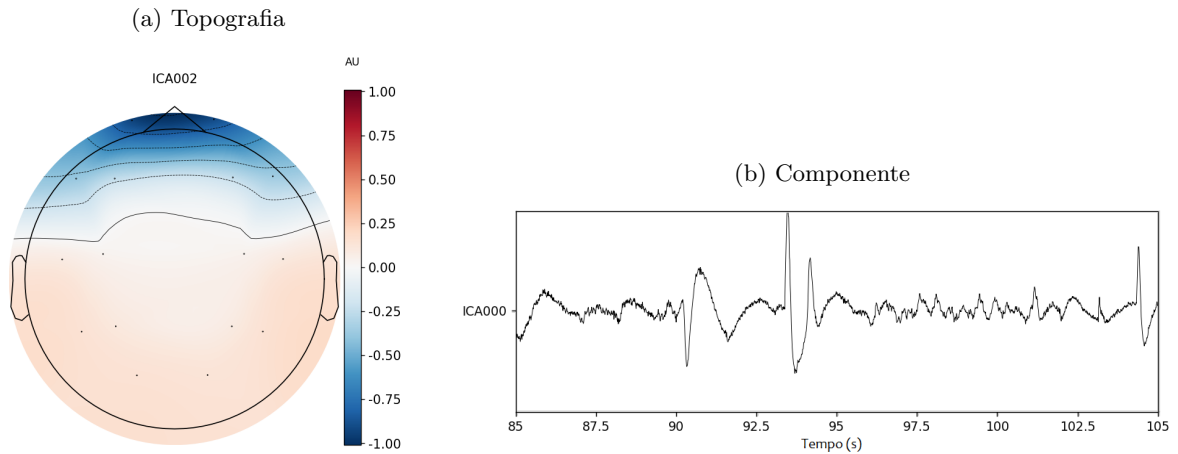
Fonte: Adaptado de McCallan et al. (2021).

As Figuras 15 e 16 apresentam alguns artefatos detectados automaticamente, sendo o primeiro detectado na base de dados 1 e o segundo detectado na base de dados 2. A

<sup>4</sup> *FastICA* (*Fast Independent Component Analysis*) é um algoritmo amplamente utilizado para a separação de sinais independentes em um conjunto de dados, sendo baseado no conceito de ICA (OJA; YUAN, 2006).

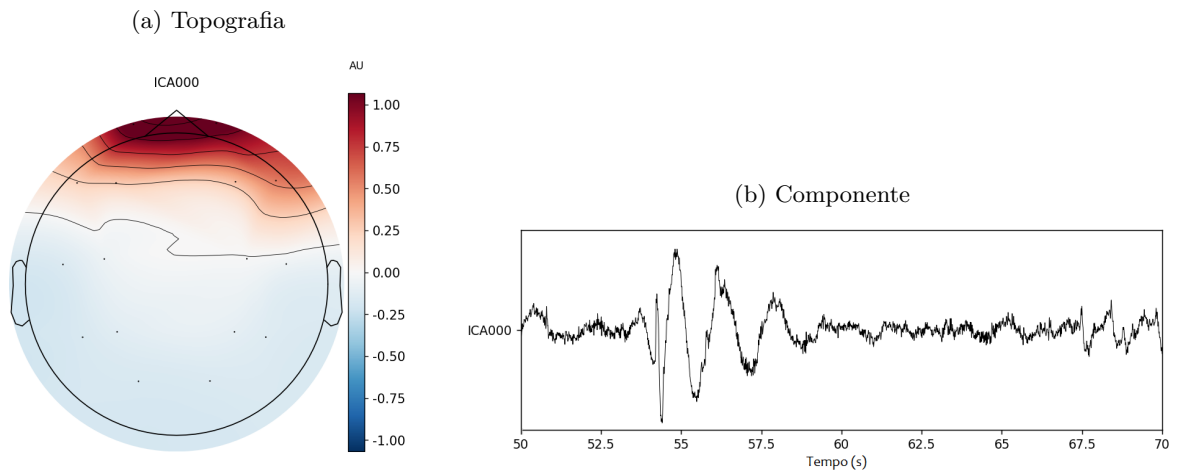
barra de cor das topografias apresentadas está em unidade arbitrária (AU). A partir da observação das topografias é possível identificar as regiões que possuem de forma mais intensa sua respectiva componente, representando a provável origem do artefato no espaço.

Figura 15 – Exemplo de artefato da base de dados 1



Fonte: Produção da própria autora.

Figura 16 – Exemplo de artefato da base de dados 2

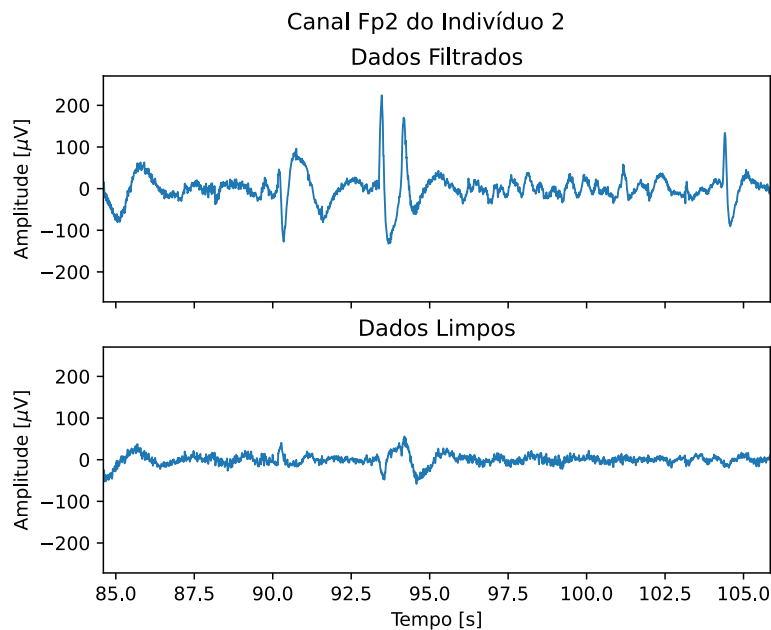


Fonte: Produção da própria autora.

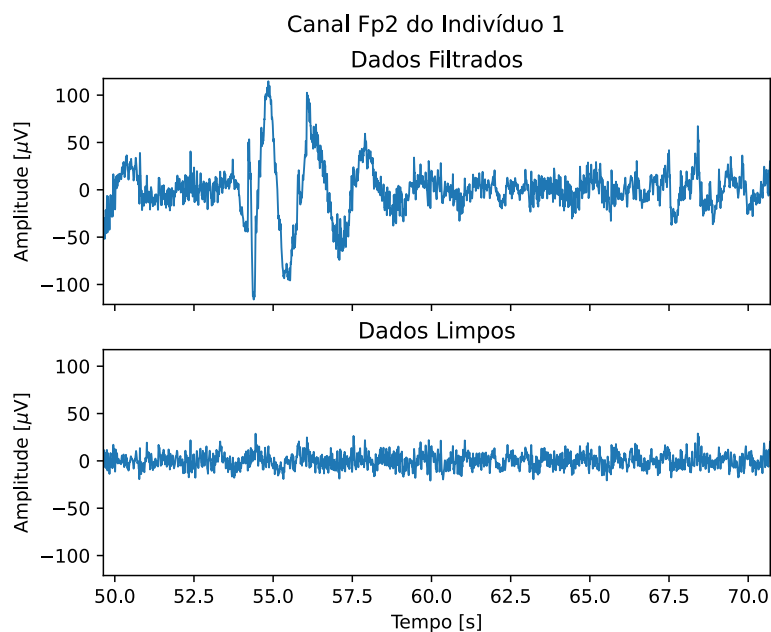
A Figura 17 mostra o antes e depois dos dados limpos. Apesar do algoritmo de ICA mostrar uma boa remoção dos sinais mostrados, nem todos os artefatos das bases foram removidos, já que foi usada a detecção automática de artefatos de movimento ocular. Porém, as métricas de avaliação do modelo obtidas ao final deste trabalho foram elevadas, indicando que as redes desenvolvidas apresentam uma robustez a dados ruidosos.

Figura 17 – Exemplo de remoção de artefatos

(a) Remoção do artefato da base de dados 1



(b) Remoção do artefato da base de dados 2



Fonte: Produção da própria autora.

### 3.2.3 Extração de Características

A etapa de extração de características foi feita com o Método de Welch para obter as densidades espectrais (PSDs, do inglês *Power Spectral Densities*), utilizando a função *welch* do pacote *Scipy* (VIRTANEN et al., 2020). Para aumentar o número de amostras de treinamento, divide-se os 5 minutos em intervalos de 1 segundo. Ou seja, cada intervalo

correspondente a um indivíduo originou 300 intervalos. Dessa forma, são classificados os intervalos e não os indivíduos, assim como foi feito por Wang et al. (2022). Por isso, o número de amostras no final dessa etapa será igual a 300 multiplicado pelo número de indivíduos da base de dados. Além disso, apesar de terem sido atenuadas as frequências acima de 50 Hz na etapa de filtragem, optou-se por não fazer o *downsampling*<sup>5</sup>, conforme visto no artigo de Qayyum, Razzak e Mumtaz (2020).

Os hiperparâmetros da função *welch* são a frequência de amostragem (para retornar um vetor com frequências desnormalizadas), a janela de Hamming, o tamanho do segmento de 100 amostras no tempo e a sobreposição de 50% na separação dos segmentos. Wang et al. (2022) não explicita o valor da sobreposição utilizado, de forma que após simular o treinamento de uma rede e, feitos alguns testes, 50% foi o valor de sobreposição que apresentou melhores valores de acurácia. Os resultados foram armazenados em *arrays 3-D*, sendo a primeira dimensão o número de amostras para o treinamento, a segunda dimensão o número de características PSD e a terceira dimensão o número de canais. Para a primeira base de dados esse *array 3-D* possui o formato  $15900 \times 51 \times 16$ , enquanto para a segunda base de dados esse *array 3-D* possui o formato de  $13500 \times 51 \times 16$ .

A Figura 18 mostra dois exemplos de periodogramas modificados obtidos de duas janelas de 1 segundo. A Figura 18a é referente a um indivíduo diagnosticado com depressão e a Figura 18b é referente a um indivíduo saudável. É perceptível que a pessoa depressiva possui um pico acentuado na região da banda alpha (8-12Hz) quando comparada à pessoa saudável.

### 3.3 Métricas

Já que amostras classificadas como janelas de alguém depressivo são positivas e amostras classificadas como janelas de alguém saudável são negativas, o problema é considerado de classificação binária. Para essa etapa, as funções *precision\_recall\_fscore\_support* e *confusion\_matrix* do pacote *metrics* da biblioteca *Scikit Learn* (Pedregosa et al., 2011b) foram usadas.

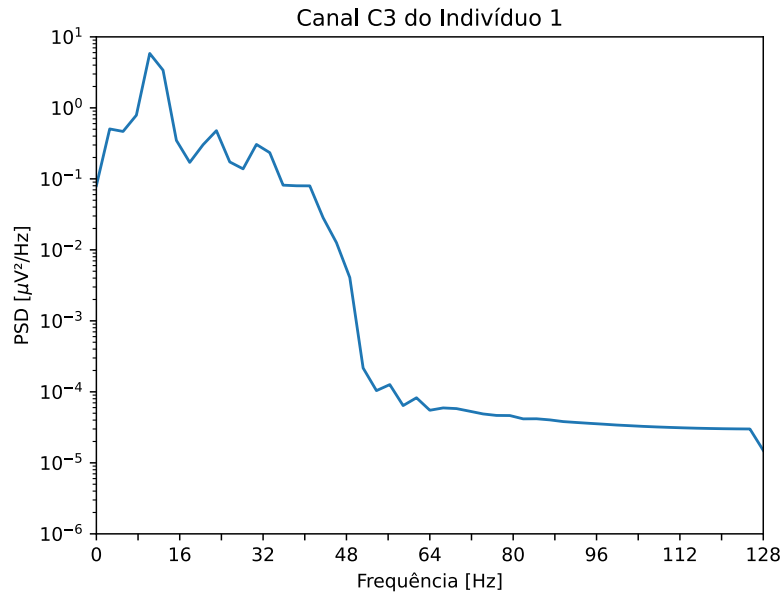
A avaliação considera as seguintes métricas:

- a acurácia ( $Ac$ ), que avalia a capacidade do modelo de classificar corretamente as

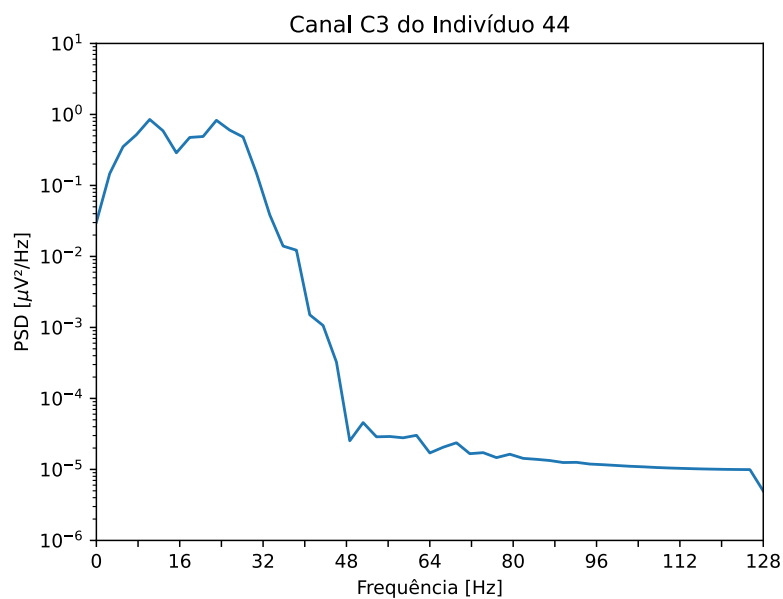
<sup>5</sup> *Downsampling* é um processo de redução da frequência de amostragem por um fator inteiro  $M$ , deixando no sinal toda  $M^a$  amostra e seus múltiplos (HARRIS, 2004).

Figura 18 – Periodogramas de uma janela de 1 segundo de uma pessoa depressiva e de uma pessoa saudável

(a) Pessoa depressiva



(b) Pessoa saudável



Fonte: Produção da própria autora.

amostras como depressão ou saudável, mostrada na Equação 21;

- a precisão ( $Pr$ ), que avalia a capacidade do modelo em classificar corretamente as amostras como depressão, evitando falsas classificações dessa classe, mostrada na Equação 22;
- a sensibilidade ( $Se$ ), que avalia a capacidade do modelo em classificar corretamente

as amostra como depressão, evitando falsas classificações como saudável, mostrada na Equação 23;

- a medida  $F_1$ , que é a média harmônica entre a sensibilidade e precisão, mostrada na Equação 24.

Os termos  $VP$  são os verdadeiros positivos (número de janelas corretamente classificadas como positivas),  $VN$  são os verdadeiros negativos (número de janelas corretamente classificadas como negativas),  $FP$  são os falsos positivos (número de janelas erroneamente classificadas como positivas), e  $FN$  são falsos negativos (número de janelas classificadas erroneamente classificadas como negativas).

$$Ac = \frac{VP + VN}{VP + VN + FP + FN} \quad (21)$$

$$Pr = \frac{VP}{VP + FP} \quad (22)$$

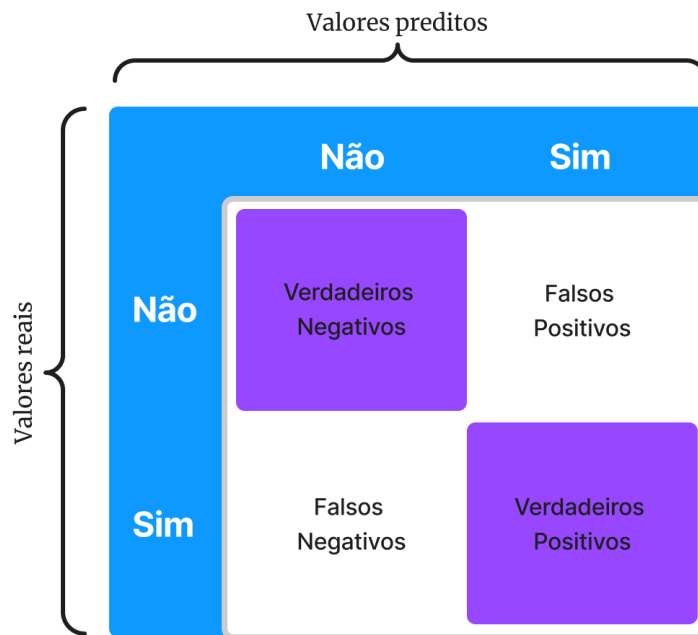
$$Se = \frac{VP}{VP + FN} \quad (23)$$

$$F_1 = 2 \frac{Se \cdot Pr}{Se + Pr} \quad (24)$$

Esses termos também podem estar dispostos em uma representação visual, chamada de matriz de confusão. Ela organiza as previsões do modelo em relação às classes reais dos dados em uma tabela. A Figura 19 mostra a organização dessa matriz. A diagonal principal, em roxo, representa as classificações corretas do modelo: a célula superior esquerda representa os verdadeiros negativos ( $VN$ ) e a célula inferior direita representa os verdadeiros positivos ( $VP$ ). A diagonal secundária, em branco, mostra os casos em que o modelo falhou: a célula superior direita representa os falsos positivos ( $FP$ ) e a célula inferior esquerda representa os falsos negativos ( $FN$ ). A intensidade da cor dos blocos pode ser usada para indicar os valores correspondentes. Geralmente, quanto mais escuro o bloco, maior é o valor associado.



Figura 19 – Estrutura de uma matriz de confusão



Fonte: Produção da própria autora.

### 3.4 Criação do Modelo

Para a construção do modelo, foi usado módulo *Keras* da biblioteca *TensorFlow* (ABADI et al., 2015), o qual permite a modelagem de redes usando uma aplicação de alto nível. Inicialmente, foi montada uma arquitetura com duas camadas CNN unidimensionais ativadas pela função ReLU para extração de características profundas do sinal, uma camada de células GRU para considerar características sequenciais do sinal e uma última camada totalmente conectada ativada por uma softmax. A Figura 20 mostra a arquitetura escolhida após o ajuste dos hiperparâmetros, o qual será explicado nos parágrafos seguintes.

Para escolher os hiperparâmetros de rede e de treinamento é necessário escolher uma estratégia que permita que retorne um modelo robusto, com alta capacidade de generalização e, conseqüentemente, evite o *overfitting*<sup>6</sup>. Dessa forma, é realizado o ajuste dos hiperparâmetros pelo método *grid search*, que consiste em treinar a rede utilizando combinações de hiperparâmetros diferentes previamente definidos (CHICCO, 2017).

A base de dados 1 foi utilizada para realizar o ajuste dos hiperparâmetros, sendo 20% dos dados separados para teste (3180 amostras) e 80% para os treinamentos (12720 amostras). Para a normalização dos dados, primeiro foi calculada a média e o desvio padrão dos dados

<sup>6</sup> *Overfitting* ocorre quando os pesos do modelo se sobreajustam nos dados de treinamento, tendo baixa taxa de acerto em dados novos (CUKIC; LOPEZ; PAVON, 2020).

de treino e depois a média foi subtraída de todos os dados e o resultado foi dividido pelo desvio padrão. Essa técnica de normalização é conhecida como *z-score* e é muito usada para a normalização de sinais de EEG (SAFAYARI; BOLHASANI, 2021). Na primeira fase do ajuste, os espaços de busca dos valores de hiperparâmetros da rede foram:

- Taxa de aprendizagem: 0,01; 0,005
- Otimizador: Adam; RMSProp
- Tamanho do kernel: 3; 5
- Número de filtros da primeira camada convolucional: 128; 256
- Número de filtros da segunda camada convolucional: 128; 256
- Células de GRU: 64; 128; 256

Os outros hiperparâmetros foram fixados em 0,2 para a fração de unidades que serão excluídas no *Dropout*<sup>7</sup> realizado após a última camada de *pooling*, 2 para o tamanho do grupo usado na camada de *pooling* (realizando a operação de valor máximo), ReLU para as funções de ativação usadas nas camadas convolucionais e softmax para função de ativação na camada de saída. O número de épocas e o tamanho dos *batches* nessa primeira fase do ajuste foram fixados em 10 e 256, respectivamente. Isso permitiu que fossem encontrados mais modelos candidatos em um intervalo de tempo menor, para que na próxima fase do ajuste esses hiperparâmetros fossem redimensionados. Foram encontrados 96 modelos e a Tabela 1 mostra os 20 modelos que apresentaram melhores acurácias sobre os dados de teste. É possível observar que o otimizador Adam foi usado em todas as 20 arquiteturas expostas, sendo preterido seu uso em relação ao RMSProp.

As arquiteturas que obtiveram maiores acurácias sobre os dados de teste foram selecionadas para a próxima fase do ajuste: 70, 15, 71 e 65. Nessa segunda fase, os hiperparâmetros de treinamento são variados:

- Número de épocas: 80, 90, 100
- Tamanho dos *batches*: 128, 256

<sup>7</sup> *Dropout* é uma camada que exclui aleatoriamente alguns neurônios da camada oculta para permitir uma melhor generalização e evitar o *overfitting* (SAFAYARI; BOLHASANI, 2021).

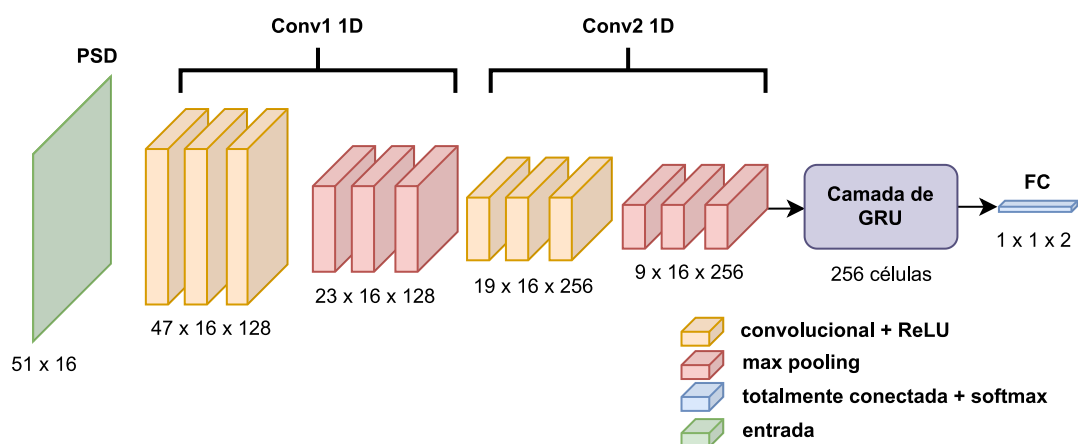
Tabela 1 – Ajuste dos parâmetros de rede

ID	Taxa	Otim.	Tam. kernel	N° filtros 1	N° filtros 2	N° GRU	Ac.	Perda
70	0,005	adam	5	256	256	128	0,9717	0,0811
15	0,01	adam	5	128	256	64	0,9714	0,0773
71	0,005	adam	5	256	256	256	0,9686	0,0957
65	0,005	adam	5	128	256	256	0,9676	0,0937
50	0,005	adam	3	128	128	256	0,9673	0,0936
60	0,005	adam	5	128	128	64	0,9664	0,1006
20	0,01	adam	5	256	128	256	0,9654	0,0919
58	0,005	adam	3	256	256	128	0,9654	0,0995
63	0,005	adam	5	128	256	64	0,9654	0,0922
68	0,005	adam	5	256	128	256	0,9654	0,0984
69	0,005	adam	5	256	256	64	0,9654	0,1010
67	0,005	adam	5	256	128	128	0,9651	0,0988
52	0,005	adam	3	128	256	128	0,9648	0,1013
2	0,01	adam	3	128	128	256	0,9645	0,1044
4	0,01	adam	3	128	256	128	0,9638	0,1033
62	0,005	adam	5	128	128	256	0,9632	0,1033
7	0,01	adam	3	256	128	128	0,9619	0,1082
21	0,01	adam	5	256	256	64	0,9619	0,1083
59	0,005	adam	3	256	256	256	0,9619	0,1016
0	0,01	adam	3	128	128	64	0,9616	0,1033

Fonte: Produção da própria autora.

A Tabela 2 mostra a acurácia e perdas calculadas sobre os dados de teste. A configuração escolhida foi a que apresentou maior acurácia: arquitetura 65, usando as 90 épocas e 256 de tamanho dos *batches*.

Figura 20 – Arquitetura adotada



Fonte: Produção da própria autora.

Tabela 2 – Ajuste dos parâmetros de treinamento

ID Rede	ID Treino	Nº épocas	Tam. batch	Ac.	Perda
65	3	90	256	0,9884	0,0702
65	4	100	128	0,9877	0,0674
65	2	90	128	0,9865	0,0809
71	3	90	256	0,9846	0,1119
70	1	80	256	0,9836	0,0952
70	5	100	256	0,9827	0,1114
71	5	100	256	0,9821	0,0887
71	0	80	128	0,9818	0,0759
65	5	100	256	0,9818	0,0725
2	2	90	128	0,9811	0,0864
65	1	80	256	0,9805	0,0905
15	0	80	128	0,9799	0,0726
70	2	90	128	0,9792	0,0779
71	4	100	128	0,9792	0,1096
70	0	80	128	0,9783	0,0803
15	1	80	256	0,9764	0,1023
15	3	90	256	0,9748	0,0946
65	0	80	128	0,9745	0,0910
71	1	80	256	0,9739	0,0805
70	3	90	256	0,9736	0,1075
15	2	90	128	0,9723	0,0945
70	4	100	128	0,9717	0,1289
15	5	100	256	0,9689	0,1063
15	4	100	128	0,9657	0,1356

Fonte: Produção da própria autora.

## 4 RESULTADOS

Este capítulo expõe os resultados obtidos do modelo proposto em relação às duas bases de dados disponíveis. A divisão de amostras de cada base de dados para a etapa de treinamento do modelo obtido é mostrada na Tabela 3. Para ambas as bases de dados foi usada a proporção de  $0,81 \times 0,09 \times 0,1$  na divisão dos dados em treino, validação e teste. A fim de normalizar os dados foi usada a técnica *z-score*.

Tabela 3 – Distribuição das amostras

	Amostras com rótulo “depressão”	Amostras com rótulo “saudável”	Total de amostras
Base de dados 1	7200	8700	15900
Base de dados 2	5700	7800	13500

Fonte: Produção da própria autora.

### 4.1 Base de dados 1

Os resultados obtidos no treinamento do modelo usando a base de dados 1 estão mostrados na Figura 21. A Figura 21a apresenta o gráfico de acurácia a cada época do treinamento, enquanto a Figura 21b apresenta o gráfico de perda a cada época do treinamento.

Na Figura 22, é apresentada a matriz de confusão obtida após aplicar o modelo ao conjunto de teste. A partir desses valores, foram calculadas as medidas de acurácia, precisão, sensibilidade e  $F_1$ , mostradas na Tabela 4.

Tabela 4 – Métricas do modelo na base de dados 1

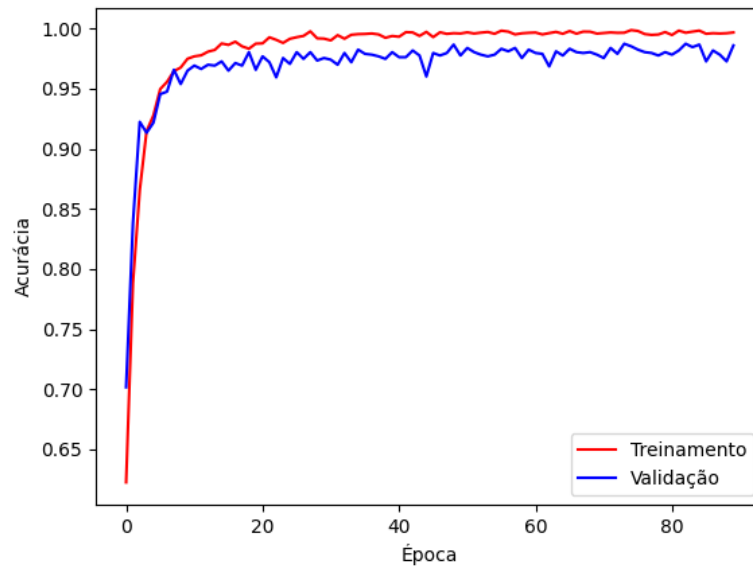
Acurácia	Precisão	Sensibilidade	F1
0,9836	0,9826	0,9798	0,9812

Fonte: Produção da própria autora.

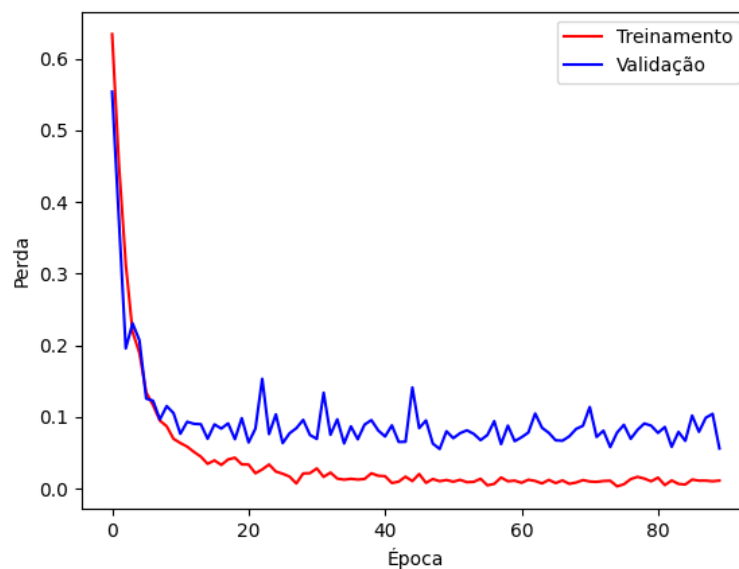
Para comparar o desempenho do modelo gerado pela arquitetura formada por camadas convolucionais unidimensionais e uma camada GRU (CNN1D+GRU) em relação a outras arquiteturas, foram criadas outras 4 arquiteturas. A comparação do desempenho das arquiteturas é feita na Figura 23. A arquitetura CNN1D+GRU foi modificada ao remover a camada GRU, resultando na arquitetura CNN1D. Já a arquitetura GRU foi criada ao remover as camadas CNN da arquitetura CNN1D+GRU. A arquitetura LSTM foi

Figura 21 – Resultados de treinamento para a base de dados 1

(a) Gráfico de Acurácia



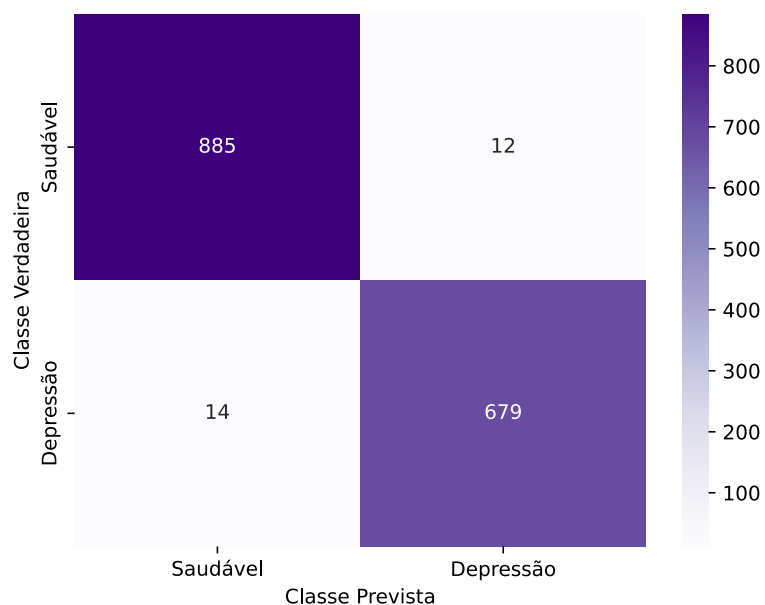
(b) Gráfico de Perda



Fonte: Produção da própria autora.

construída substituindo a camada GRU por uma camada LSTM e retirando as camadas CNN da arquitetura CNN1D+GRU. Por fim, a arquitetura CNN1D+LSTM foi criada substituindo a camada GRU por uma camada LSTM na arquitetura CNN1D+GRU. A arquitetura CNN1D+GRU foi a arquitetura dentre as testadas que teve a maior acurácia, menor perda e segundo menor tempo de treinamento para essa base de dados. A acurácia alcançada de 98,36% é maior que a encontrada por Wang et al. (2022). Os autores obtiveram sobre a mesma base de dados 96% de acurácia, porém usando o classificador

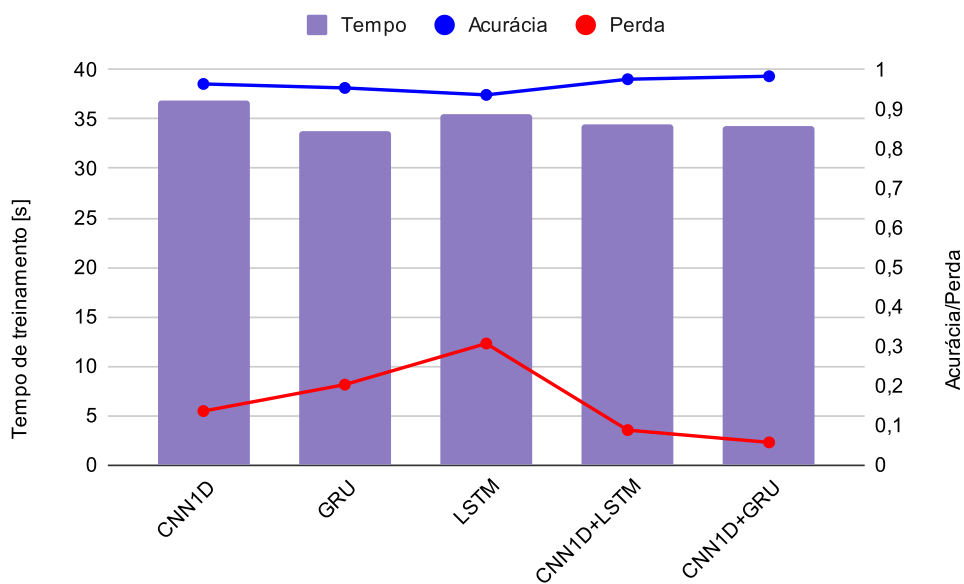
Figura 22 – Matriz de confusão obtida para a base de dados 1



Fonte: Produção da própria autora.

RMSProp e sem a segunda camada de *pooling*.

Figura 23 – Comparativo entre modelos com respeito a base de dados 1



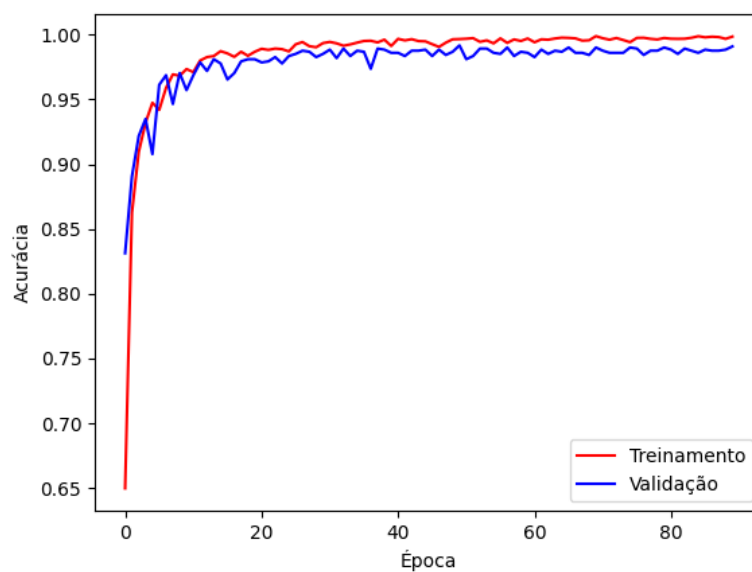
Fonte: Produção da própria autora.

## 4.2 Base de dados 2

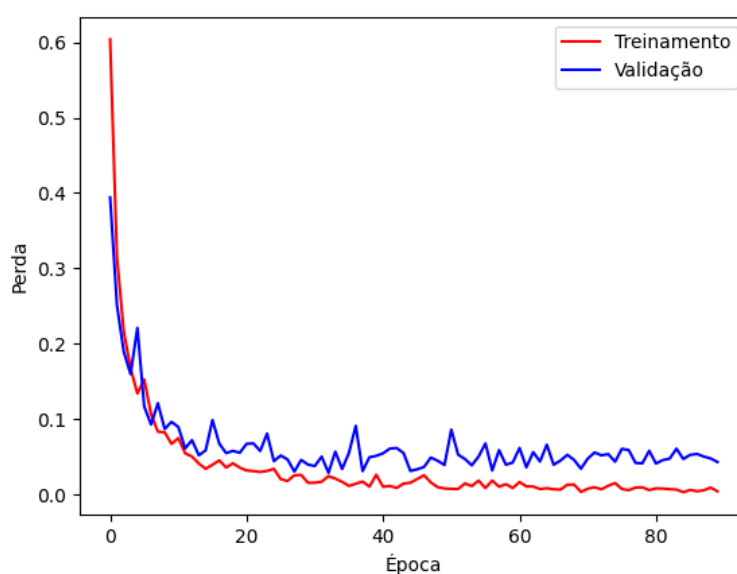
Os resultados obtidos no treinamento do modelo usando a base de dados 2 estão mostrados na Figura 24. A Figura 24a apresenta o gráfico de acurácia a cada época do treinamento, enquanto a Figura 24b apresenta o gráfico de perda a cada época do treinamento.

Figura 24 – Resultados de treinamento para a base de dados 2

(a) Gráfico de Acurácia



(b) Gráfico de Perda



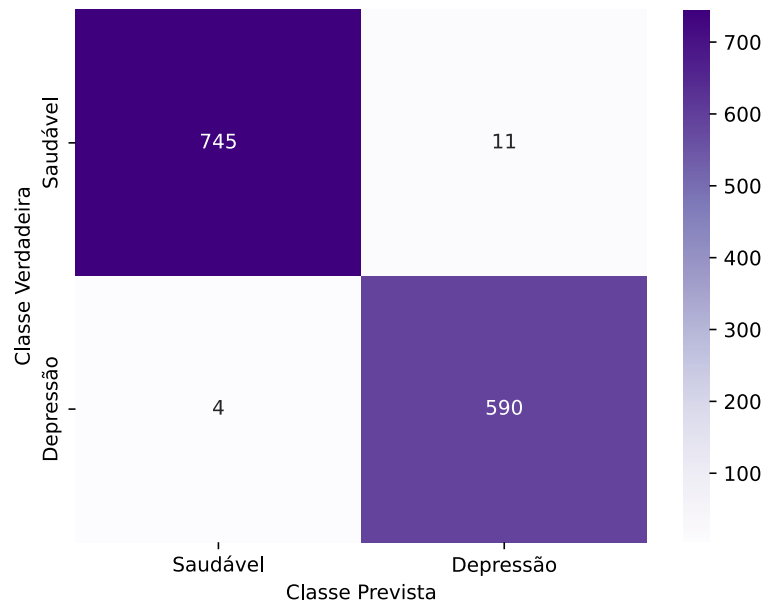
Fonte: Produção da própria autora.

Na Figura 25, é apresentada a matriz de confusão obtida após aplicar o modelo ao conjunto



de teste, conforme explicado na Seção 4.1. A partir dos valores apresentados na matriz de confusão, foram calculadas as medidas de acurácia, precisão, sensibilidade e  $F_1$ , mostradas na Tabela 5.

Figura 25 – Matriz de confusão para a base de dados 2



Fonte: Produção da própria autora.

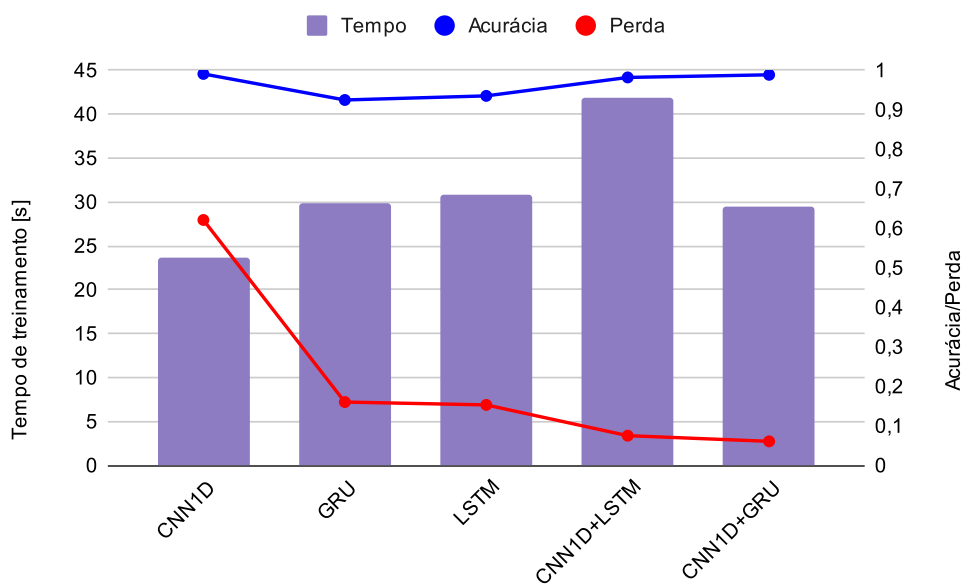
Tabela 5 – Métricas do modelo na base de dados 2

Acurácia	Precisão	Sensibilidade	F1
0,9888	0,9817	0,9933	0,9874

Fonte: Produção da própria autora.

A criação e comparação entre os modelos foi feita de forma análoga à feita na Seção 4.1, sendo mostrada na Figura 26. A arquitetura CNN1D+GRU foi a arquitetura dentre as testadas que teve a segunda maior acurácia, menor perda e segundo menor tempo para essa base de dados. A acurácia alcançada de 98,88% é similar à encontrada por Qayyum, Razzak e Mumtaz (2020). Os autores obtiveram sobre a mesma base de dados 98,87% de acurácia, porém foram usadas duas camadas de GRU e 3 canais a mais (Fz, Cz, Pz) no treinamento do modelo. Além disso, não foi realizada a etapa de extração de características PSD e a taxa de aprendizagem usada foi 0,0004.

Figura 26 – Comparativo entre modelos com respeito a base de dados 2



Fonte: Produção da própria autora.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

### 5.1 Conclusão

Este trabalho objetivou a implementação de uma rede híbrida para auxiliar no diagnóstico de depressão que obtivesse alta assertividade em seus resultados. O objetivo foi alcançado, já que foi possível alcançar e até mesmo superar a acurácia de algoritmos similares da literatura. Foi montada uma arquitetura híbrida rasa utilizando duas camadas convolucionais unidimensionais e uma camada de GRU para gerar o modelo que teve resultados elevados para ambas as bases de dados utilizadas. Para a base de dados 1 (da coleção de dados MODMA) foi obtida uma acurácia de 98,36% e para a base de dados 2 (da coleção de dados *MDD Patients and Healthy Controls EEG Data*) foi obtida uma acurácia de 98,88%.

### 5.2 Trabalhos futuros

Uma sugestão de trabalho futuro seria a criação de uma aplicação com uma interface intuitiva e amigável para os usuários. Essa aplicação seria especialmente relevante para o contexto hospitalar, visando facilitar o processo de teste e treinamento da rede neural desenvolvida. Isso representaria um passo importante na busca por soluções práticas e implementáveis em hospitais, contribuindo para um melhor diagnóstico de depressão.

Outra possibilidade seria mudar o foco da detecção para a predição de depressão. Em vez de apenas identificar a presença ou ausência de depressão, o objetivo seria desenvolver modelos capazes de prever o risco ou o desenvolvimento futuro da depressão em indivíduos. Isso poderia envolver a análise de dados longitudinais, o uso de técnicas de aprendizado de máquina para identificar padrões e fatores de risco, e a criação de modelos preditivos para auxiliar na intervenção precoce e no acompanhamento de pacientes (KUMAR; SUBHA, 2019).

A técnica de LOO (*Leave-One-Out*) pode ser uma estratégia valiosa para o treinamento de redes em trabalhos futuros. Ela permite uma avaliação mais abrangente do desempenho do modelo, utilizando de forma mais eficiente os dados disponíveis. Dessa forma, pode ser considerada a aplicação dessa técnica para melhorar a precisão e a generalização dos modelos de rede neural. No entanto, a técnica de LOO pode ser computacionalmente

intensiva, já que envolve o treinamento de vários modelos para cada amostra.

## REFERÊNCIAS

- ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Software available from [tensorflow.org](https://www.tensorflow.org/). Disponível em: <<https://www.tensorflow.org/>>. Citado na página 40.
- ACHARYA, U. R. et al. Automated eeg-based screening of depression using deep convolutional neural network. Computer Methods and Programs in Biomedicine, v. 161, p. 103–113, 2018. ISSN 0169-2607. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169260718301494>>. Citado na página 13.
- Akdemir Akar, S. et al. Nonlinear analysis of eegs of patients with major depression during different emotional states. Computers in Biology and Medicine, v. 67, p. 49–60, 2015. ISSN 0010-4825. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0010482515003297>>. Citado na página 31.
- American Psychiatric Association. What Is Depression? 2023. Disponível em: <<https://www.psychiatry.org/patients-families/depression/what-is-depression>>. Acesso em: 26 jun. 2023. Citado na página 30.
- ATTAR, E. T. Review of electroencephalography signals approaches for mental stress assessment. Neurosciences, v. 27, n. 4, p. 209–215, out. 2022. ISSN 1319-6138. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9749579/>>. Citado na página 16.
- AY, B. et al. Automated depression detection using deep representation and sequence learning with eeg signals. Journal of Medical Systems, v. 43, n. 7, p. 205, May 2019. ISSN 1573-689X. Disponível em: <<https://doi.org/10.1007/s10916-019-1345-y>>. Citado na página 13.
- CAI, H. et al. MODMA dataset: a multi-modal open dataset for mental- disorder analysis. p. 15, 2020. Citado 2 vezes nas páginas 16 e 30.
- CAI, H. et al. Pervasive eeg diagnosis of depression using deep belief network with three-electrodes eeg collector. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). [S.l.: s.n.], 2016. p. 1239–1246. Citado na página 13.
- CHICCO, D. Ten quick tips for machine learning in computational biology. BioData Min, v. 10, p. 35, 2017. Citado na página 40.
- CHO, K. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014. Citado na página 27.
- CONNIE, R. et al. Biology. OpenStax, 2016. Disponível em: <<https://openstax.org/books/biology/pages/1-introduction/>>. Acesso em: 02 fev. 2023. Citado na página 17.
- CUKIC, M.; LOPEZ, V.; PAVON, J. Classification of depression through resting-state electroencephalogram as a novel practice in psychiatry: Review. v. 22, n. 11, p. e19548, 2020. ISSN 1438-8871. Disponível em: <<https://www.jmir.org/2020/11/e19548>>. Citado 2 vezes nas páginas 12 e 40.

- DING, Y. et al. A depression recognition method for college students using deep integrated support vector algorithm. IEEE Access, v. 8, p. 75616–75629, 2020. Citado na página 12.
- FERREIRA, A. Uma proposta de interface cérebro-computador para comando de cadeiras de rodas. Tese de doutorado — Universidade Federal do Espírito Santo, 2008. Citado 2 vezes nas páginas 18 e 20.
- FU, J. et al. Sch-net: a deep learning architecture for automatic detection of schizophrenia. v. 20, n. 1, p. 75, 2021. ISSN 1475-925X. Disponível em: <<https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-021-00915-2>>. Citado na página 12.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Back-propagation and other differentiation algorithms. In: Deep Learning. [S.l.]: MIT Press, 2016. p. "200". Citado na página 26.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 3 vezes nas páginas 25, 26 e 27.
- GOOGLE. Google Colabotory. 2022. Disponível em: <<https://colab.research.google.com/notebooks/intro.ipynb>>. Acesso em: 22 jan. 2023. Citado na página 29.
- Gramfort, A. et al. MNE-Python: Enabling Reproducible MEG/EEG Research. Frontiers in Neuroscience, v. 7, p. 267, 2013. Disponível em: <<https://doi.org/10.3389/fnins.2013.00267>>. Citado na página 34.
- HARRIS, F. J. Multirate Signal Processing for Communication Systems. Philadelphia, PA: Prentice Hall, 2004. Citado na página 37.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. Neural computation, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 27.
- HYVÄRINEN, A.; OJA, E. Independent component analysis: algorithms and applications. Neural networks, Elsevier, v. 13, n. 4-5, p. 411–430, 2000. Citado na página 19.
- IHME. Global Health Data Exchange (GHDx). 2019. Disponível em: <<https://vizhub.healthdata.org/gbd-results>>. Acesso em: 11 jan. 2023. Citado na página 12.
- JIA, X.; KOHN, A. Gamma rhythms in the brain. v. 9, n. 4, p. e1001045, 2011. ISSN 1545-7885. Disponível em: <<https://dx.plos.org/10.1371/journal.pbio.1001045>>. Citado na página 16.
- JWO, D.-J.; CHANG, W.-Y.; WU, I.-H. Windowing techniques, the welch method for improvement of power spectrum estimation. CMC-Computers Materials & Continua, TECH SCIENCE PRESS, 2021. Citado na página 23.
- KIM, S.-H.; GEEM, Z. W.; HAN, G.-T. Hyperparameter optimization method based on harmony search algorithm to improve performance of 1d cnn human respiration pattern recognition system. Sensors, v. 20, n. 13, 2020. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/20/13/3697>>. Citado na página 26.

- KIRANYAZ, S. et al. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, v. 151, p. 107398, 2021. ISSN 0888-3270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0888327020307846>>. Citado na página 13.
- KUMAR, S.; SUBHA, D. Prediction of depression from eeg signal using long short term memory(lstm). In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. [S.l.: s.n.], 2019. p. 1248–1253. Citado na página 50.
- LAKSSHMAN, S. et al. Deepbipolar: Identifying genomic mutations for bipolar disorder via deep learning. *Human mutation*, Wiley Online Library, v. 38, n. 9, p. 1217–1224, 2017. Citado na página 12.
- MALMIVUO, J.; PLONSEY, R. Electroencephalography. In: *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. [S.l.]: Oxford University Press, 1995. p. 247–264. ISBN 978-0195058239. Citado na página 18.
- MAO, W. et al. Resting state eeg based depression recognition research using deep learning method. In: WANG, S. et al. (Ed.). *Brain Informatics*. Cham: Springer International Publishing, 2018. p. 329–338. ISBN 978-3-030-05587-5. Citado na página 13.
- MCCALLAN, N. et al. Seizure classification of eeg based on wavelet signal denoising using a novel channel selection algorithm. 09 2021. Citado 2 vezes nas páginas 17 e 34.
- MUMTAZ, W. MDD Patients and Healthy Controls EEG Data (New). 2016. Disponível em: <[https://figshare.com/articles/dataset/EEG\\_Data\\_New/4244171](https://figshare.com/articles/dataset/EEG_Data_New/4244171)>. Acesso em: 30 mai. 2023. Citado na página 31.
- MUMTAZ, W. et al. Electroencephalogram (eeg)-based computer-aided technique to diagnose major depressive disorder (mdd). *Biomedical Signal Processing and Control*, v. 31, p. 108–115, 2017. ISSN 1746-8094. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1746809416300866>>. Citado na página 31.
- OJA, E.; YUAN, Z. The fastICA algorithm revisited: convergence analysis. *IEEE Trans Neural Netw, United States*, v. 17, n. 6, p. 1370–1381, nov. 2006. Citado na página 34.
- Pedregosa, F. et al. Metrics and scoring: quantifying the quality of predictions. In: *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 2011. Disponível em: <[https://scikit-learn.org/stable/modules/model\\_evaluation.html#log-loss](https://scikit-learn.org/stable/modules/model_evaluation.html#log-loss)>. Acesso em: 02 jul. 2023. Citado na página 25.
- Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Disponível em: <<https://scikit-learn.org/>>. Acesso em: 28 ago. 2022. Citado na página 37.
- PION-TONACHINI, L.; KREUTZ-DELGADO, K.; MAKEIG, S. Iclabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage*, Elsevier, v. 198, p. 181–197, 2019. Citado 2 vezes nas páginas 21 e 22.

- QAYYUM, A.; RAZZAK, I.; MUMTAZ, W. Hybrid deep shallow network for assessment of depression using electroencephalogram signals. In: YANG, H. et al. (Ed.). Neural Information Processing. Cham: Springer International Publishing, 2020. p. 245–257. ISBN 978-3-030-63836-8. Citado 3 vezes nas páginas 13, 37 e 48.
- SAEEDI, A. et al. Major depressive disorder diagnosis based on effective connectivity in eeg signals: a convolutional neural network and long short-term memory approach. Cognitive Neurodynamics, v. 15, n. 2, p. 239–252, Apr 2021. ISSN 1871-4099. Disponível em: <<https://doi.org/10.1007/s11571-020-09619-0>>. Citado na página 13.
- SAFAYARI, A.; BOLHASANI, H. Depression diagnosis by deep learning using EEG signals: A systematic review. v. 12, p. 100102, 2021. ISSN 25900935. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S2590093521000461>>. Citado 2 vezes nas páginas 13 e 41.
- SANEI, S.; CHAMBERS, J. A. EEG signal processing. [S.l.]: John Wiley & Sons, 2013. Citado na página 16.
- SHARMA, G.; JOSHI, A. M. Novel EEG based schizophrenia detection with IoMT framework for smart healthcare. 2021. Disponível em: <<http://arxiv.org/abs/2111.11298>>. Citado na página 12.
- SINGH, B.; WAGATSUMA, H. A removal of eye movement and blink artifacts from eeg data using morphological component analysis. Computational and mathematical methods in medicine, 2017. ISSN 1748-670X. Citado na página 34.
- SIULY, S.; LI, Y.; ZHANG, Y. Significance of eeg signals in medical and health research. In: EEG signal analysis and classification. [S.l.]: Springer, 2016. p. 23–41. Citado na página 16.
- VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, v. 17, p. 261–272, 2020. Citado 2 vezes nas páginas 32 e 36.
- WAJID, M.; ABDUL, Q. A deep learning framework for automatic diagnosis of unipolar depression. International Journal of Medical Informatics, v. 132, p. 103983, 2019. ISSN 1386-5056. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1386505619307154>>. Citado na página 13.
- WANG, Z. et al. A depression diagnosis method based on the hybrid neural network and attention mechanism. v. 12, n. 7, p. 834, 2022. ISSN 2076-3425. Disponível em: <<https://www.mdpi.com/2076-3425/12/7/834>>. Citado 5 vezes nas páginas 13, 24, 29, 37 e 45.
- WELCH, P. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Transactions on audio and electroacoustics, IEEE, v. 15, n. 2, p. 70–73, 1967. Citado na página 20.
- WHO. Covid-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide. 2022. Citado na página 12.
- WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. Chemometrics and Intelligent Laboratory Systems, v. 2, n. 1, p. 37–52, 1987. ISSN 0169-7439. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.



Disponível em: <<https://www.sciencedirect.com/science/article/pii/0169743987800849>>. Citado na página 20.

ZHANG, A. et al. Dive into Deep Learning. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2106.11342>>. Acesso em: 31 jan. 2023. Citado na página 27.