

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo Borges Silva Oliveira

**Redes neurais convolucionais baseadas em
filtros espaciais para prognóstico de pacientes
em coma via sinais EEG**

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo Borges Silva Oliveira

**Redes neurais convolucionais baseadas em filtros
espaciais para prognóstico de pacientes em coma via
sinais EEG**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Murillo Guimarães Carneiro

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Bacharelado em Sistemas de Informação				
Defesa de:	FACOM31802 - Trabalho de Conclusão de Curso				
Data:	02/05/2024	Hora de início:	09:05	Hora de encerramento:	10:45
Matrícula do Discente:	11921BSI240				
Nome do Discente:	Gustavo Borges Silva Oliveira				
Título do Trabalho:	Redes neurais convolucionais baseadas em filtros espaciais para prognóstico de pacientes em coma via sinais EEG				
A carga horária curricular foi cumprida integralmente?	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não				

Reuniu-se remotamente a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Sistemas de Informação, assim composta: Prof. Dr. Bruno Augusto Nassif Travençolo (FACOM/UFU); Me. Lucas Zago Bissaro (Dell); e Prof. Dr. Murillo Guimarães Carneiro (FACOM/UFU), orientador do candidato.

Iniciando os trabalhos, o presidente da mesa, Dr. Murillo Guimarães Carneiro, apresentou a Comissão Examinadora e o candidato(a), agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado Nota [100]

OU

Aprovado(a) sem nota.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo, Professor(a) do Magistério Superior**, em 02/05/2024, às 10:50, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

Resumo

Nos últimos anos, houveram avanços significativos na detecção e recuperação de pacientes que sofrem de diferentes graus de perda de consciência. No entanto, a obtenção de um prognóstico confiável continua sendo um desafio. O desenvolvimento de novas arquiteturas, bem como o uso de arquiteturas existentes de aprendizado de máquina, aplicados aos exames de EEG, podem beneficiar a tarefa de obtenção de prognósticos mais confiáveis para pacientes em coma. Portanto, o objetivo deste trabalho foi o de desenvolver e avaliar modelos de aprendizado profundo com o intuito de gerar prognósticos confiáveis para pacientes em coma utilizando dados de EEG. Foram utilizadas 121 amostras de traçado de EEG, que consistiu em 42 amostras com desfecho favorável e 79 com desfecho desfavorável. Dentro de cada registro, os neurologistas segmentaram 10 trechos considerados significativos para a elaboração do prognóstico do paciente. Estes trechos, com duração de 2 segundos cada, apresentam diferentes resoluções temporais, com a frequência de amostragem dos exames variando entre 100Hz e 600Hz. Foram investigadas duas abordagens para o processamento dos dados. Na primeira, foram usados 9 de 10 trechos previamente separados pelos especialistas. Na segunda, foram selecionados aleatoriamente 9 trechos diretamente do traçado original, sem possibilidade de sobreposição. Em ambos os casos, as amostras passaram por um processo de reamostragem, padronizando a frequência de amostragem do conjunto de dados em 100Hz e 200Hz. As arquiteturas escolhidas para o desenvolvimento dos modelos foram a *Deep ConvNet*, *Shallow ConvNet* e *EEGNet*, todas apoiadas, majoritariamente, em camadas convolucionais para aprendizado de padrões temporais e espaciais relevantes para o problema em estudo. Simulações foram conduzidas considerando validação cruzada com 10 pastas, sendo que em cada iteração o conjunto de treino foi submetido a um algoritmo de aumento de dados realizado por uma implementação da técnica *FTS surrogate*. As métricas de avaliação utilizadas foram a acurácia, especificidade, sensibilidade e F1-Score. As três arquiteturas apresentaram bom desempenho, principalmente quando submetidas à classificação de trechos de EEG selecionados por especialistas. Nesse contexto, a arquitetura *Shallow ConvNet* obteve destaque nos experimentos que consideraram a frequência de amostragem dos exames em 100Hz e 200Hz, enquanto a *Deep ConvNet* se destacou quando a frequência foi padronizada em 200Hz. Os modelos desenvolvidos e avaliados neste trabalho apresentaram bom desempenho, com potencial de contribuir significativamente para a prática clínica, como ferramentas auxiliares, eficazes e confiáveis, no prognóstico de pacientes em estado de coma.

Palavras-chave: Aprendizado de máquina, Aprendizado profundo, Sinais de EEG, Prognóstico de pacientes em coma.

Abstract

In recent years, there have been significant advances in the detection and recovery of patients suffering from varying degrees of loss of consciousness. However, obtaining a reliable prognosis remains a challenge. The development of new architectures, as well as the use of existing machine learning architectures, applied to EEG exams, can benefit the task of obtaining more reliable prognoses for coma patients. Therefore, the objective of this work was to develop and evaluate deep learning models with the aim of generating reliable prognoses for coma patients using EEG data. 121 EEG tracing samples were used, which consisted of 42 samples with a favorable outcome and 79 with an unfavorable outcome. Within each record, the neurologists segmented 10 sections considered significant for preparing the patient's prognosis. These excerpts, lasting 2 seconds each, have different temporal resolutions, with the sampling frequency of the exams varying between 100Hz and 600Hz. Two approaches to data processing were investigated. In the first, 9 out of 10 excerpts previously separated by experts were used. In the second, 9 sections were randomly selected directly from the original route, without the possibility of overlapping. In both cases, the samples went through a resampling process, standardizing the sampling frequency of the data set at 100Hz and 200Hz. The architectures chosen for the development of the models were *Deep ConvNet*, *Shallow ConvNet* and *EEGNet*, all mainly supported by convolutional layers for learning temporal and spatial patterns relevant to the problem in question. Simulations were conducted considering cross-validation with 10 folders, and in each iteration the training set was subjected to a data augmentation algorithm performed by an implementation of the *FTSurrogate* technique. The evaluation metrics used were accuracy, specificity, sensitivity and F1-Score. The three architectures performed well, especially when subjected to classification of EEG excerpts selected by experts. In this context, the *Shallow ConvNet* architecture stood out in experiments that considered the sampling frequency of exams at 100Hz and 200Hz, while *Deep ConvNet* stood out when the frequency was standardized at 200Hz. The models developed and evaluated in this work showed good performance, with the potential to contribute significantly to clinical practice, as effective and reliable auxiliary tools in the prognosis of patients in a state of coma.

Keywords: Machine learning, Deep learning, EEG signals, Prognosis of patients in coma.

Lista de ilustrações

Figura 1 – Esquema de posicionamento de eletrodos seguindo o Sistema Internacional	12
Figura 2 – Exemplo de traçado de exame de EEG	13
Figura 3 – Hierarquia Clássica de Tarefas de Aprendizado de Máquina	14
Figura 4 – Representação de um sistema de aprendizado supervisionado	14
Figura 5 – Ilustração da Rede Perceptron	15
Figura 6 – Arquitetura de uma rede Perceptron Multicamadas	16
Figura 7 – Arquitetura básica de uma rede convolucional	17
Figura 8 – Exemplo de operação de convolução.	17
Figura 9 – Exemplo de operações de Max Pooling e Average Pooling.	18
Figura 10 – Exemplo de aplicação de quatro filtros CSP	19
Figura 11 – Aplicação de filtro CSP projetado sobre a posição dos eletrodos	20
Figura 12 – Arquitetura proposta para a técnica FBCSP	20
Figura 13 – Arquitetura Deep ConvNet	21
Figura 14 – Arquitetura Shallow ConvNet	23
Figura 15 – Ilustração da arquitetura de rede EEGNet	23
Figura 16 – Fluxograma do processo de desenvolvimento do trabalho	27
Figura 17 – Seleção de dados aleatória	30
Figura 18 – Eletrodo FP2 de amostra com frequência original de amostragem 400Hz (a) e após <i>downsampling</i> para 100Hz (b).	31
Figura 19 – Arquivo .YAML de configuração do experimento	33
Figura 20 – K-Fold Cross Validation com $K = 10$	37
Figura 21 – Exemplo de aplicação da técnica FTSurrogate: (a) Trecho de exame de EEG, eletrodo F7, de paciente comatoso; (b) Trecho de exame de EEG gerado a partir de FTSurrogate aplicada ao eletrodo F7; e (c) Trecho de exame original sobreposto ao sinal gerado após aplicação de FTSurrogate.	38
Figura 22 – Exemplo de matriz de confusão	39
Figura 23 – Página inicial da ferramenta Optuna	41
Figura 24 – Histórico de experimento	41
Figura 25 – Importância de hiperparâmetros	42
Figura 26 – Detalhamento de experimentos	42

Lista de tabelas

Tabela 1 – Parâmetros considerados para otimização da arquitetura Deep ConvNet.	34
Tabela 2 – Parâmetros considerados para otimização da arquitetura Shallow Conv-Net	35
Tabela 3 – Parâmetros considerados para otimização da arquitetura EEGNet . . .	36
Tabela 4 – Resultados de modelos que utilizam trechos selecionados por especialistas em porcentagem, com média e desvio-padrão.	43
Tabela 5 – Resultados de modelos que utilizam trechos selecionados aleatoriamente, com média e desvio-padrão	44

Lista de abreviaturas e siglas

CSP	<i>Common Spatial Patterns</i>
EEG	Eletroencefalograma
ESN	Redes echo-state
FBCSP	<i>Filter Bank Common Spatial Patterns</i>
FTSurrogate	<i>Fourier Transform Surrogates</i>
KNN	<i>K-Nearest Neighbours</i>
LR	<i>Logistic Regression</i>
MLP	Perceptron Multi-Camadas
OC	Óbito clínico
OME	Óbito por morte encefálica
SVM	<i>Support Vector Machines</i>

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.2	Justificativa	10
1.3	Organização do Trabalho	10
2	REVISÃO BIBLIOGRÁFICA	12
2.1	Eletroencefalograma	12
2.2	Aprendizado de Máquina	13
2.2.1	Redes Neurais Artificiais	14
2.2.2	Redes Neurais Convolucionais	16
2.3	Redes Neurais Convolucionais para EEG	18
2.3.1	Filter Bank Common Spatial Patterns (FBCSP)	18
2.3.2	Deep ConvNet	21
2.3.3	Shallow ConvNet	22
2.3.4	EEGNet	22
2.4	Fourier Transform Surrogates	24
2.5	Trabalhos Correlatos	24
3	MATERIAIS E MÉTODOS	27
3.1	Conjunto EEG-Coma	27
3.2	Ambiente e Ferramentas	28
3.3	Aprendizado de Sinais EEG usando FBCSP e Redes Convolucionais	30
3.3.1	Seleção e pré-processamento	30
3.3.2	Montagem de Experimentos	32
3.3.3	Treinamento de Modelos	36
3.3.4	Avaliação de Modelos	38
3.3.5	Avaliação de Experimentos	40
4	RESULTADOS E DISCUSSÃO	43
4.1	Resultados com seleção de trechos por especialistas	43
4.2	Resultados com trechos selecionados aleatoriamente	44
4.3	Discussão dos resultados experimentais	45
5	CONCLUSÃO	47
	REFERÊNCIAS	49

1 Introdução

Aprendizado de máquina é um campo da ciência da computação que busca a construção de modelos utilizando uma combinação de técnicas matemáticas, estatísticas e algoritmos computacionais avançados. Esses modelos exploram uma ampla gama de atributos dos dados para detectar padrões e relacionamentos complexos, muitas vezes incompreensíveis para humanos, com o objetivo de fazer previsões confiáveis sobre resultados futuros (PAIXÃO et al., 2022).

De acordo com um relatório publicado em 2020 pela Fundação Oswaldo Cruz (Fiocruz), a aplicação dessas técnicas tem demonstrado resultados excelentes em diversos domínios, especialmente na área da saúde. Já se observam vantagens significativas em sua utilização, como uma maior precisão nos diagnósticos, melhor acesso aos cuidados de saúde e redução da carga de trabalho dos médicos. Estas técnicas são frequentemente empregadas em uma variedade de propósitos, incluindo o desenvolvimento de medicamentos, detecção de doenças e avaliação de tratamentos, dentre outros, com resultados muito positivos.

Um tipo de exame que pode se beneficiar da aplicação de novas técnicas de aprendizado de máquina é o eletroencefalograma (EEG), que visa medir a atividade elétrica em diversos locais do cérebro. Os dados obtidos a partir desse exame apresentam alta irregularidade e uma grande riqueza de informações que podem ser analisados de diversas maneiras, incluindo seus aspectos morfológicos e temporais (RAMOS, 2022). Este tipo de exame apresenta alternativas mais sofisticadas, como a ressonância magnética funcional (fMRI), que tem a vantagem de fornecer dados como nível de oxigenação do cérebro durante o exame, a partir de uma perspectiva espacial. No entanto, devido ao seu custo elevado, tais alternativas geralmente não são viáveis em hospitais públicos brasileiros (MULERT; LEMIEUX, 2009).

O coma, derivado do termo latino para “sono profundo”, refere-se a uma condição na qual o paciente apresenta um grau parcial ou total de perda de consciência, movimento e sensibilidade. Geralmente, essa condição é resultado de lesões cerebrais ou outros problemas, como questões relacionadas a metabolismo ou funções endócrinas, embora o paciente possa ainda manter um certo nível de funcionamento adequado das funções vitais do corpo (PUGGINA; SILVA, 2009). Pacientes que se encontram nessa condição são atendidos e tratados em unidades de terapia intensiva, presentes em diversos hospitais públicos brasileiros, como o Hospital de Clínicas da Universidade Federal de Uberlândia (HC-UFU), e envolvem monitoramento constante por meio de exames como o EEG.

O processo de monitoramento geralmente envolve a interpretação do traçado de EEG, onde especialistas buscam por padrões conhecidos e realizam a atribuição de um

possível desfecho. No entanto, enfrenta diversos desafios, como a longa duração do exame, a alta dimensionalidade dos dados e a presença significativa de ruídos nas séries temporais registradas. Além disso, o exame pode gerar padrões altamente complexos, muitas vezes irreconhecíveis para os seres humanos, mas que poderiam ser interpretados com maior eficiência e precisão por algoritmos avançados de aprendizado de máquina (KANDA et al., 2014).

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver e avaliar modelos de aprendizado profundo com o intuito de gerar prognósticos confiáveis para pacientes em coma utilizando dados de EEG. Os objetivos específicos são:

- Analisar e padronizar diferentes frequências de amostragem para tratamento de dados de exames de EEG.
- Avaliar de maneira comparativa arquiteturas de aprendizado profundo para a decodificação de traçados de EEG.
- Avaliar o impacto da seleção de trechos de exame, realizada de forma aleatória e por especialistas, na performance dos modelos desenvolvidos.
- Desenvolver e avaliar diferentes modelos de aprendizado profundo com base nas arquiteturas selecionadas.

1.2 Justificativa

Nos últimos anos, houveram avanços significativos na detecção e recuperação de pacientes que sofrem de diferentes graus de perda de consciência. No entanto, a obtenção de um prognóstico confiável continua sendo um desafio (EDLOW et al., 2021). O desenvolvimento de novas arquiteturas, bem como o uso de arquiteturas existentes de aprendizado de máquina, aplicados aos exames de EEG, podem beneficiar a tarefa de obtenção de prognósticos mais confiáveis para pacientes em coma.

1.3 Organização do Trabalho

O restante deste trabalho está dividido nos seguintes capítulos:

- No capítulo 2, é realizada a revisão bibliográfica, abordando os conceitos de eletroencefalograma, aprendizado de máquina, redes neurais artificiais e convolucionais, e outros conceitos fundamentais para o entendimento e desenvolvimento do trabalho.

-
- No capítulo 3, são apresentados os materiais e métodos, incluindo o conjunto de dados, ambiente experimental, técnica de pré-processamento, e medidas de avaliação utilizadas na realização dos experimentos.
 - No capítulo 4, os resultados dos modelos desenvolvidos são analisados, interpretados, discutidos e comparados com estudos realizados anteriormente na literatura, fornecendo uma base sólida para reflexão.
 - Por fim, o capítulo 5 apresenta as principais descobertas, contribuições, limitações do estudo e proposições para trabalhos futuros.

2 Revisão Bibliográfica

Este capítulo revisa conceitos fundamentais dos principais tópicos para entendimento deste trabalho, como sinais de eletroencefalograma, aprendizado de máquina, redes neurais artificiais, técnica *Filter Bank Common Spatial Patterns* (FBCSP), medidas de avaliação e arquiteturas utilizadas para treinamento dos modelos.

2.1 Eletroencefalograma

O eletroencefalograma (EEG) é um exame que registra os sinais elétricos do córtex cerebral em várias regiões do cérebro, utilizando eletrodos colocados sobre o couro cabeludo. Esses sinais são gerados pela atividade elétrica dos neurônios, que comunicam entre si por meio de impulsos elétricos (BISSARO, 2021).

Existem várias maneiras de organizar a disposição dos eletrodos no couro cabeludo, sendo o sistema mais amplamente utilizado definido pelo Comitê da Federação Internacional de Sociedades de Eletroencefalografia e Neurofisiologia Clínica (IFSECN), conhecido como sistema internacional 10-20, que estabelece um intervalo de 10% a 20% entre cada eletrodo. Os eletrodos posicionados no hemisfério esquerdo do escalpo têm numeração ímpar, enquanto os do hemisfério direito tem numeração par. Além disso, letras são usadas para definir a localização do eletrodo, por exemplo, a letra “F” refere-se à região frontal, conforme ilustrado na Figura 1 (RAMOS, 2022).

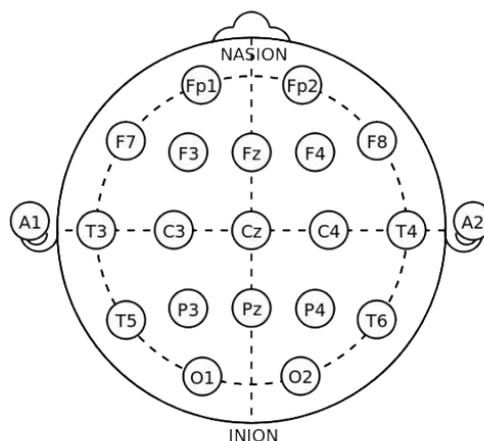


Figura 1 – Esquema de posicionamento de eletrodos seguindo o Sistema Internacional 10-20.

Fonte: Ramos (2022).

Os sinais de EEG possuem alguns padrões normais e anormais morfológicos bem difundidos entre os especialistas, apesar de sua grande variedade de ritmos, que variam de

acordo com o estado mental do sujeito no momento da captura, o que torna a observação de características quase impossível sem o auxílio de rotinas de processamento (RAMOS, 2022). A Figura 2 mostra um exemplo de traçado de EEG.



Figura 2 – Exemplo de traçado de exame de EEG.

Fonte: [MaestroVirtuale \(2024\)](#).

2.2 Aprendizado de Máquina

Atualmente, um dos temas que desperta considerável interesse na computação é o de algoritmos de aprendizado de máquina, que engloba o desenvolvimento de sistemas computacionais capazes de identificar novos padrões nos dados de forma autônoma, usando o conhecimento passado como base para seu aprendizado. Ao longo do tempo, uma variedade de algoritmos foram propostos, cada um com sua abordagem única para resolver problemas em diversos domínios. Esses algoritmos têm sido aplicados em uma ampla gama de contextos, desde a detecção de transações fraudulentas em cartões de crédito até o aprendizado sobre preferências de leitura e consumo de usuários, além de contribuírem para o desenvolvimento de veículos autônomos (MITCHELL, 1997).

Nesse campo, uma variedade de modelos é construída com objetivo de auxiliar em uma tarefa de aprendizado. De acordo com a hierarquia clássica, o aprendizado de máquina pode ser dividido em tarefas de aprendizado supervisionado e não supervisionado. Conforme ilustrado na Figura 3, as tarefas do aprendizado supervisionado podem ser divididas em preditivas e descritivas. As tarefas preditivas visam a predição de um ou mais rótulos, em casos de tarefa de classificação, ou valores contínuos, em casos de tarefa de regressão. Já as tarefas de aprendizado descritivas têm como missão a identificação de padrões e relações entre os dados, sem a necessidade de predição de valores (FACELI et al., 2021).

Por sua vez, as tarefas de aprendizado preditivas são denominadas de classificação quando o conjunto de dados possui uma quantidade finita de rótulos disponíveis para predição. Por outro lado, se o atributo-alvo pertencer a um conjunto contínuo de valores,

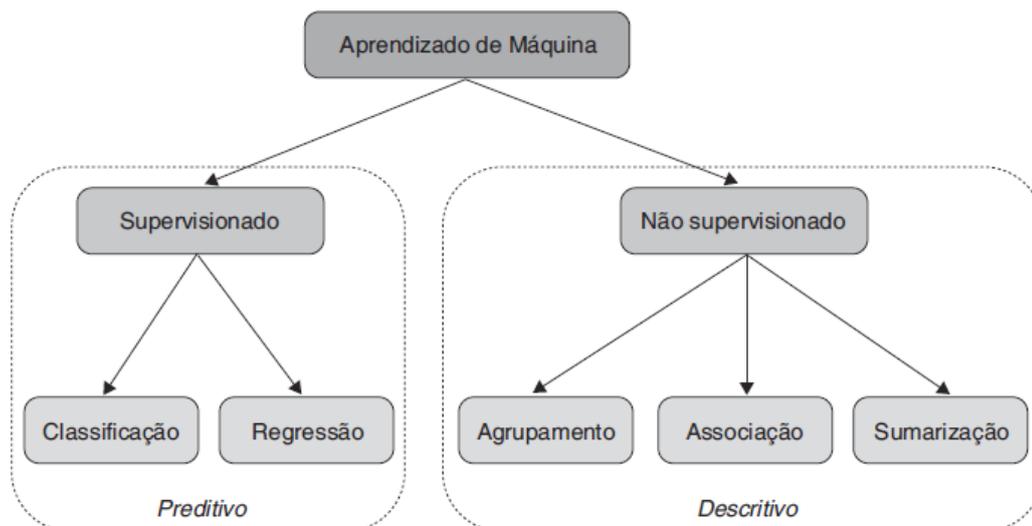


Figura 3 – Hierarquia clássica de tarefas de aprendizado de máquina.

Fonte: Faceli et al. (2021).

essas tarefas são chamadas de regressão. Os relacionamentos entre os valores de entrada e saída dos modelos de aprendizado são mapeados por parâmetros que, quando não são extraídos das amostras de entrada, devem ser estimados (LIU; WU, 2012).

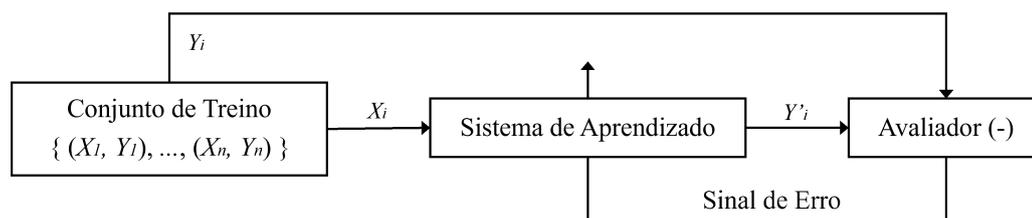


Figura 4 – Representação de um sistema de aprendizado supervisionado.

Fonte: Liu e Wu (2012).

A Figura 4 representa um sistema de aprendizado supervisionado, onde um conjunto de dados de treinamento é dividido em conjuntos X_i e Y_i , destinados ao treinamento e avaliação posterior, respectivamente. As amostras do conjunto X são submetidas ao algoritmo, que então realiza a criação do modelo preditivo. Posteriormente, as amostras do conjunto Y_i são introduzidas ao modelo, que gera rótulos de saída Y'_i . Por fim, esses rótulos são comparados com os rótulos reais Y_i por um avaliador. Caso sejam observadas diferenças, o avaliador envia um sinal de erro para que o sistema calibre os parâmetros do modelo utilizado (LIU; WU, 2012).

2.2.1 Redes Neurais Artificiais

Uma rede neural é um modelo de aprendizado de máquina projetado para simular o funcionamento e a estrutura dos neurônios, com o objetivo de tomar decisões de maneira semelhante a seres humanos. Essas redes são compostas por uma camada de entrada, pelo

menos uma camada oculta e uma camada de saída, composta por neurônios artificiais interconectados (IBM, 2024).

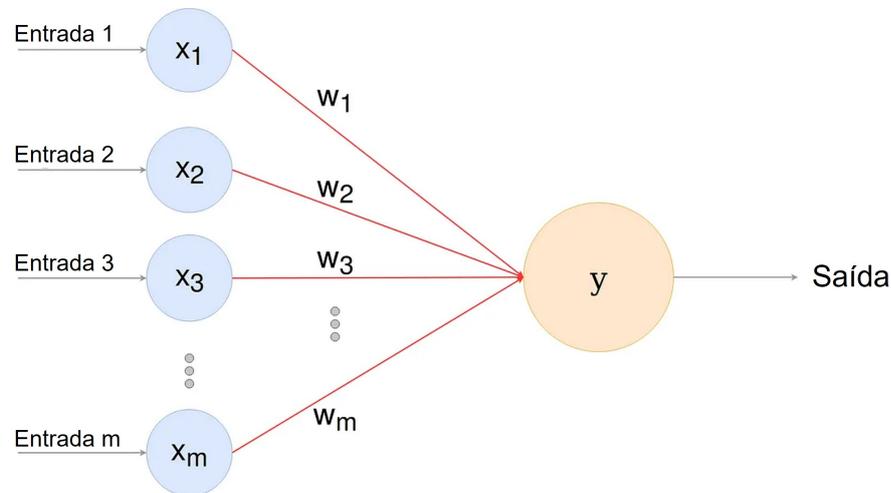


Figura 5 – Ilustração da Rede Perceptron.
Fonte: Kanagamani et al. (2023).

Uma maneira simplificada de representar esse conceito é através da rede Perceptron, mostrado na Figura 5. Ela inicia sua operação na camada de entrada, onde recebe uma ou várias entradas X_i . Cada seta vermelha simboliza uma sinapse, sempre associada a um peso W_i . Os valores de cada atributo de entrada são multiplicados por seu respectivo peso, resultando em uma soma ponderada (KANAGAMANI et al., 2023).

No entanto, apenas com a soma ponderada, ainda não é possível inferir um rótulo-alvo para o conjunto de entrada. Para isso, é necessário aplicar uma função de ativação, responsável por converter a entrada em uma saída correspondente em um formato pré-determinado. Existem diversas funções disponíveis para cumprir esse propósito, como a função Unidade Linear Retificada (ReLU), tangente hiperbólica (Tanh) e a função sigmóide. A escolha de uma função adequada para o problema é crucial para o sucesso dos modelos preditivos desenvolvidos (BHARDWAJ, 2020).

Uma forma simples de ilustrar o conceito de função de ativação é por meio da definição de um limiar (*threshold*):

$$\text{saída} = \begin{cases} 0 & \text{se } \sum_m w_m x_m \leq \text{limiar} \\ 1 & \text{se } \sum_m w_m x_m > \text{limiar} \end{cases} \quad (2.1)$$

Na equação 2.1, adaptada de Nielsen (2015), é definida uma função de ativação na qual o rótulo-alvo assume o valor 0 se a soma ponderada for menor ou igual ao limiar, e 1 se for maior. Essa função é conhecida como função degrau e é uma das funções de ativação mais simples e antigas. Em um exemplo prático, o valor de saída 1 do neurônio

deste exemplo pode representar um prognóstico positivo, enquanto o valor de saída 0 pode representar a prognóstico negativo.

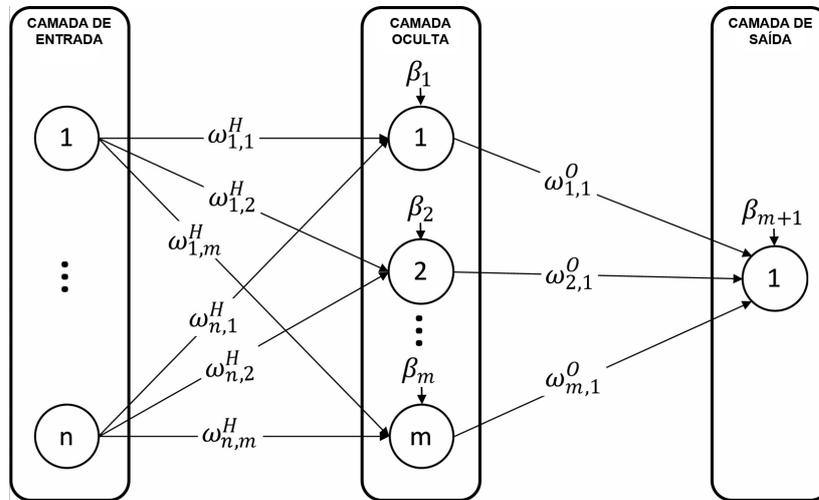


Figura 6 – Arquitetura de uma Rede Perceptron Multicamadas.
Fonte: Rojas, Olivera e Vidal (2022).

Ao combinar múltiplos neurônios Perceptron, é possível criar uma rede neural artificial, comumente conhecida como Perceptron Multi-Camadas (MLP). Como mostrado na Figura 6, os dados fluem da camada de entrada em direção à camada de saída. Esse tipo de rede é chamado de rede *feedforward*, onde todos os neurônios de uma camada i estão conectados aos neurônios da camada subsequente $i + 1$ (ROJAS; OLIVERA; VIDAL, 2022). Com o passar do tempo, outros tipos de modelos de redes neurais foram desenvolvidos para atender tipos específicos de problemas ou entradas, como as redes neurais recorrentes, consideradas estado da arte para representação de problemas sequenciais, e as redes neurais convolucionais, que são reconhecidas como estado da arte para domínios que envolvem classificação de imagens (BISSARO, 2021).

2.2.2 Redes Neurais Convolucionais

As redes neurais convolucionais foram inicialmente desenvolvidas para análise de imagens, respondendo aos estímulos de maneira semelhante ao córtex visual humano, concentrando-se em pequenas regiões da visão. o que as torna eficazes na extração de características importantes das imagens. A Figura 7 oferece uma representação resumida da estrutura de uma rede neural convolucional básica (BISSARO, 2021).

O funcionamento da camada convolucional se baseia na operação matemática de convolução. Nesta operação, uma máscara de tamanho $m \times n$ é sobreposta a uma matriz de atributos bidimensional de igual tamanho e multiplicada pelos valores correspondentes, gerando um único valor na célula resultante. Em seguida, a máscara é deslocada para uma célula vizinha, o tamanho do deslocamento pode ser controlado por um parâmetro chamado *stride* (passo), e a operação é repetida, produzindo uma matriz de características

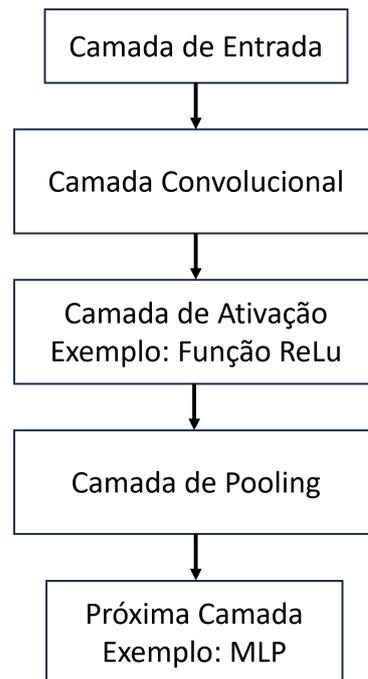


Figura 7 – Arquitetura básica de uma rede convolucional.
Fonte: Goodfellow, Bengio e Courville (2016).

como resultado (SHAFKAT, 2018). A Figura 8 mostra a operação de convolução para uma matriz 5×5 e uma máscara 3×3 com os valores $\{0, 1, 2, 2, 2, 0, 0, 1, 2\}$.

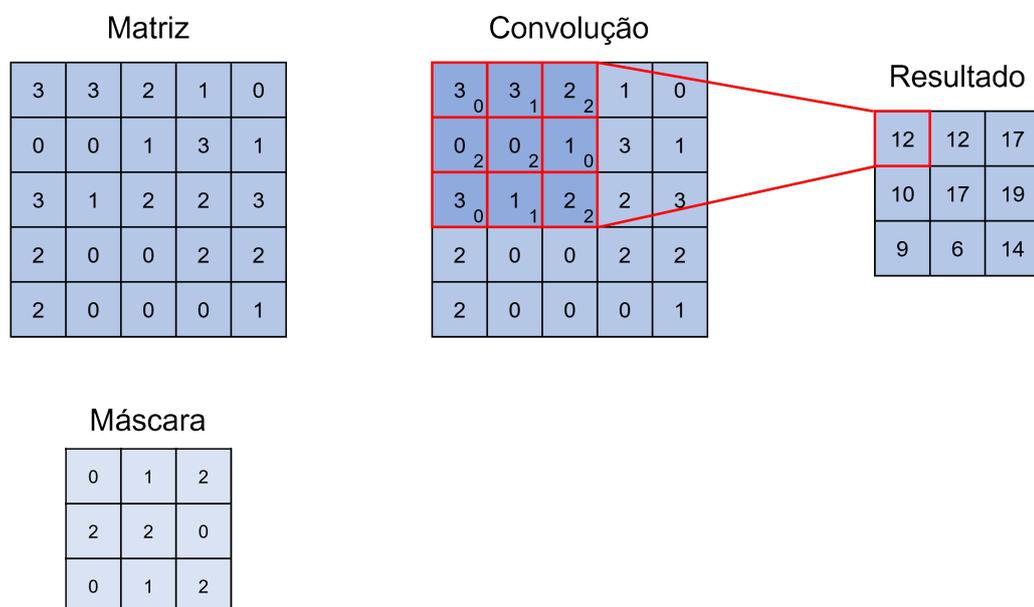


Figura 8 – Exemplo de operação de convolução.
Fonte: Shafkat (2018).

O resultado da camada de convolução é utilizado como entrada para uma camada de ativação. Em seguida, o resultado é alimentado a uma camada de *pooling*, que substitui o valor de saída da camada em um ponto específico por uma sumarização estatística de seus vizinhos. Essa operação possui extrema utilidade no treinamento deste tipo de mo-

delo, pois agrupa os valores de pontos próximos, tornando-os menos sensíveis a pequenas variações na camada de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016). Por exemplo, uma camada *Max Pooling* seleciona o valor máximo dos vizinhos cobertos pela máscara, enquanto uma camada de *Average Pooling* calcula a média aritmética desses valores, ambas são ilustradas na Figura 9.

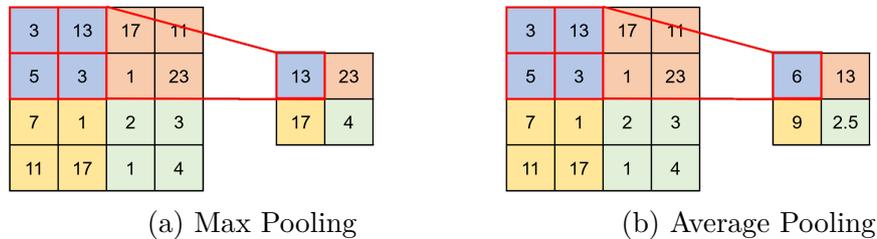


Figura 9 – Exemplo de operações de Max Pooling e Average Pooling.
Fonte: Aljaafari (2018).

2.3 Redes Neurais Convolucionais para EEG

Os algoritmos tradicionais de aprendizado de máquina geralmente dividem a extração de características e a classificação em etapas distintas, sendo a seleção de atributos frequentemente realizada manualmente. No entanto, a introdução de técnicas de visão computacional, especialmente as redes neurais convolucionais, oferece uma solução para esse problema. Quando alimentadas com sinais de traçado de EEG, essas redes são capazes de automatizar a seleção de características relevantes e classificação dentro de um mesmo processo (BORRA; FANTOZZI; MAGOSSO, 2020). As subseções seguintes oferecem uma introdução à técnica FBCSP e às arquiteturas estudadas neste trabalho.

2.3.1 Filter Bank Common Spatial Patterns (FBCSP)

No processamento de traçados de EEG, é comum enfrentar a baixa resolução espacial e a alta resolução temporal do exame, agravadas pela duração e pela frequência de amostragem da captura (BISSARO, 2021). Diversos procedimentos podem ser adotados para mitigar esse problema, incluindo a seleção de descritores e características relevantes obtidas por inspeção manual. No entanto, essa prática nem sempre é viável e pode ser substituída pela aplicação de técnicas como o *Common Spatial Patterns* (CSP) (CARIELLO et al., 2023).

A técnica CSP é utilizada para a análise de componentes de traçados de EEG voltados à classificação entre duas condições ou classes, resultando na computação de uma série de filtros espaciais. Os filtros calculados, ao serem aplicados aos traçados, resultam em padrões que maximizam a variância do sinal nas regiões mais relevantes para

uma classe, enquanto minimizam a variância em outras regiões de menor importância (BLANKERTZ et al., 2008).

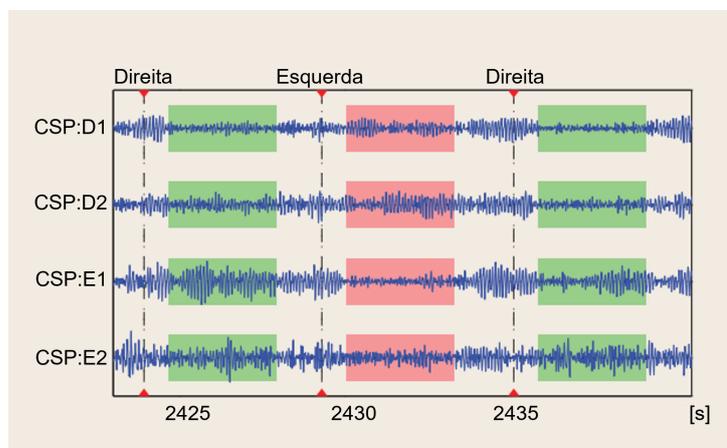


Figura 10 – Exemplo da aplicação de quatro filtros CSP.
Fonte: Blankertz et al. (2008).

A Figura 10 apresenta o resultado da aplicação de quatro filtros CSP calculados em um contexto no qual o classificador busca distinguir entre o movimento imaginado da mão esquerda e da mão direita a partir de um traçado de EEG. As marcações no topo indicam o momento em que se inicia o pensamento referente a cada movimento, com os trechos em verde correspondendo à mão direita e, em vermelho, à mão esquerda. É possível observar que os sinais correspondentes aos intervalos de movimento da mão direita exibem maior variância em CSP:E1 e CSP:E2, enquanto os trechos referentes aos intervalos de movimento da mão esquerda apresentam maior variância em CSP:D1 e CSP:D2 (BLANKERTZ et al., 2008).

A Figura 11, por sua vez, exemplifica a aplicação de dois filtros CSP para destacar as regiões ativadas pelo movimento imaginado da mão direita e esquerda, respectivamente. As áreas de maior variância são representadas em verde, enquanto as de menor variância aparecem em roxo. No exemplo à esquerda, observa-se que, ao imaginar o movimento da mão direita, a variância é maximizada na região do hemisfério esquerdo, enquanto é minimizada no hemisfério direito. No exemplo à direita, verifica-se que, ao imaginar o movimento da mão esquerda, a variância é ressaltada no hemisfério sul, com os maiores valores concentrados na região sudoeste Blankertz et al. (2008). Apesar de seu impacto positivo, o desempenho de classificadores que utilizam os padrões extraídos pelo CSP depende da pré-filtragem de faixas específicas de frequência e pode ser afetado de maneira negativa caso o traçado não seja adequadamente filtrado (ANG et al., 2008).

Em Ang et al. (2008), uma nova abordagem é proposta, denominada *Filter Bank Common Spatial Patterns* (FBCSP), cuja ideia principal é otimizar a extração de características do sinal de EEG a partir da combinação da técnica CSP com um banco de filtros aplicado a múltiplas bandas de frequência, o que pode melhorar a extração de caracte-

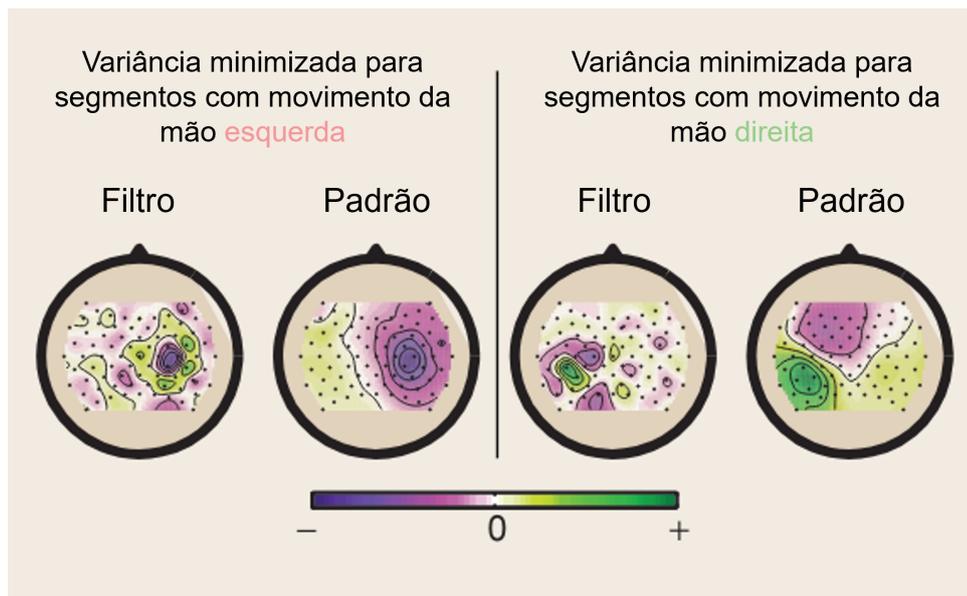


Figura 11 – Aplicação de filtro CSP projetado sobre a posição dos eletrodos.
 Fonte: Blankertz et al. (2008).

rísticas complexas do traçado do exame. Esse processo ocorre em 4 etapas, ilustradas na Figura 12.

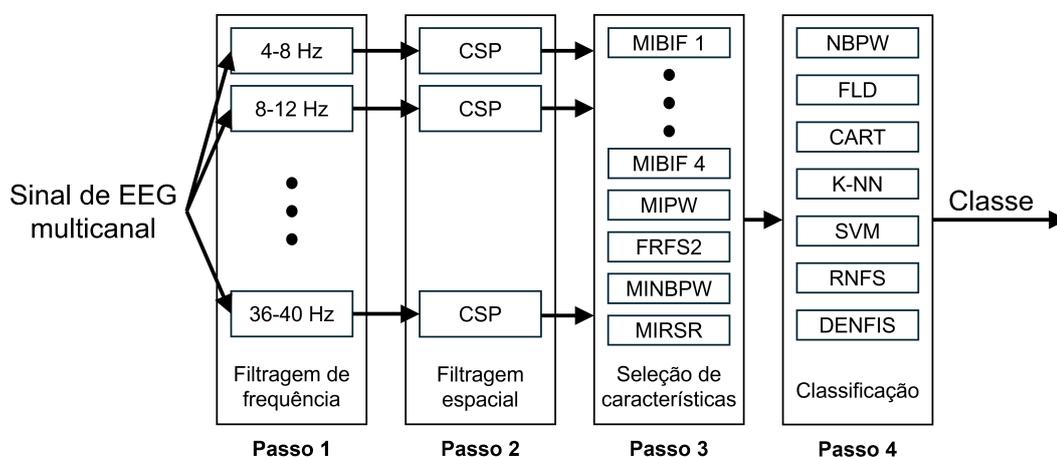


Figura 12 – Arquitetura proposta para a técnica FBCSP.
 Adaptado de: Ang et al. (2008).

O processamento proposto pela arquitetura do FBCSP têm início com a aplicação de uma série de filtros passa-banda Chebyshev Tipo 2 em todos os canais do traçado de EEG, visando a segmentação do sinal bruto em diversas faixas de banda distintas. Em seguida, cada uma dessas bandas é submetida à filtragem espacial pelo algoritmo CSP, o que resulta na obtenção de novas características para cada faixa. No terceiro estágio, um algoritmo de seleção de características é empregado no banco de filtros obtidos na etapa anterior, com o objetivo de obter características que discriminem as classes do problema. Por fim, na quarta etapa, é proposto a utilização de um algoritmo supervisionado, com o objetivo de obter a classe da amostra processada (ANG et al., 2008).

2.3.2 Deep ConvNet

O crescente entendimento e adesão das redes neurais convolucionais as posicionou como uma das opções mais promissoras para o treinamento de arquiteturas profundas destinadas à análise e classificação de sinais de EEG. Nesse contexto, em [Schirrneister et al. \(2017\)](#) propõe-se uma arquitetura profunda denominada como *Deep ConvNet*, visando criar uma estrutura profunda capaz de agir de forma abrangente sobre os sinais de EEG por meio de uma extração de características mais eficiente e flexível, ao mesmo tempo em que mantém o desempenho competitivo em relação a outras arquiteturas.

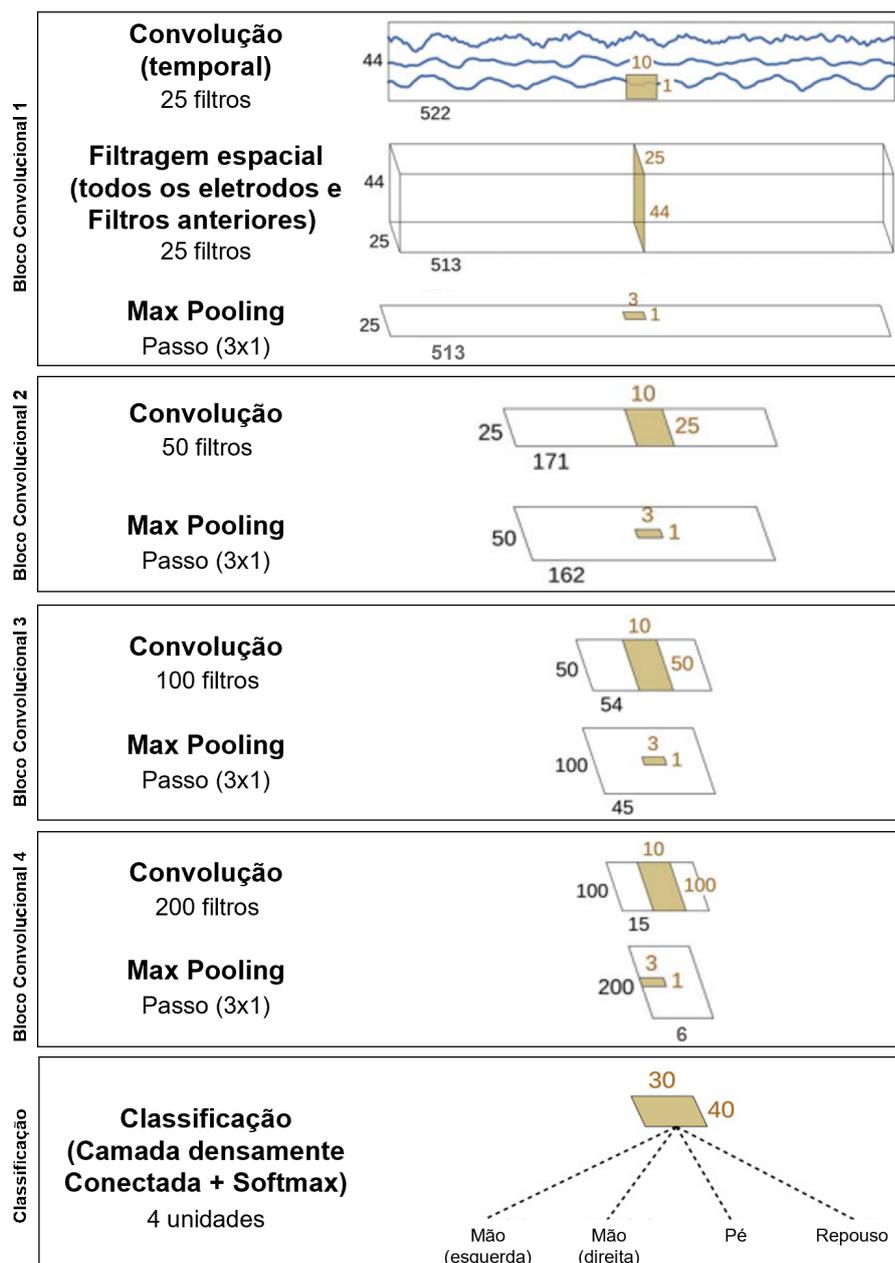


Figura 13 – Arquitetura Deep ConvNet.

Fonte: [Schirrneister et al. \(2017\)](#).

O diagrama disposto na Figura 13 oferece um resumo da arquitetura. O primeiro

bloco convolucional incorpora a técnica FBCSP. Nesta fase, uma camada convolucional é empregada para realizar uma convolução sobre os canais temporais do sinal de EEG. Os filtros gerados são então encaminhados para outra camada convolucional, cujo propósito é extrair filtros espaciais capazes de discriminar as características do sinal. O resultado é então processado por uma camada de *MaxPooling*. Os blocos subsequentes da rede operam de maneira similar a uma rede convolucional tradicional. No entanto, o número de filtros extraídos entre cada etapa aumenta, com o intuito de capturar características locais e de baixo nível em estágios iniciais, evoluindo gradualmente para características mais globais e de alto nível à medida que a profundidade da rede aumenta. A última camada é densamente conectada, com o objetivo de realizar a classificação final do registro (SCHIRRMEISTER et al., 2017).

2.3.3 Shallow ConvNet

A *Shallow ConvNet*, também proposta em Schirrneister et al. (2017), é uma alternativa mais rasa à arquitetura *Deep ConvNet*. Em comparação, a *Shallow ConvNet* não inclui os três blocos de convolução que precedem a camada densamente conectada. Além disso, a técnica de sumarização adotada para finalizar o bloco de extração de filtros temporais e espaciais é o *Average Pooling*. O diagrama disposto na Figura 14 fornece uma visão da disposição de camadas da arquitetura.

2.3.4 EEGNet

Uma outra arquitetura que adota a técnica FBCSP é a *EEGNet*. Esta arquitetura oferece uma abordagem compacta e flexível, além de ser adequada para uma diversidade de problemas de decodificação de sinais de EEG. Uma das suas vantagens é a capacidade de ser treinada com uma quantidade limitada de amostras disponíveis e a possibilidade de produzir atributos neurofisiológicos legíveis (LAWHERN et al., 2016).

Em sua concepção, essa arquitetura é composta por quatro etapas distintas (Figura 15). Na primeira etapa, uma camada de convolução temporal é aplicada para segmentar o sinal em várias faixas de frequência diferentes. Em seguida, cada uma dessas faixas é submetida a uma camada convolucional *Depthwise*, operando separadamente em cada eletrodo. Isso permite que a rede aprenda filtros espaciais específicos para cada faixa de frequência. Essas duas etapas iniciais são fortemente inspiradas pelo FBCSP. Em seguida, ocorre uma convolução *Separable*, que sintetiza as informações adquiridas anteriormente e determina a melhor maneira de combinar os filtros obtidos. Por fim, a sequência termina com uma camada densamente conectada para realizar a classificação final (LAWHERN et al., 2016).

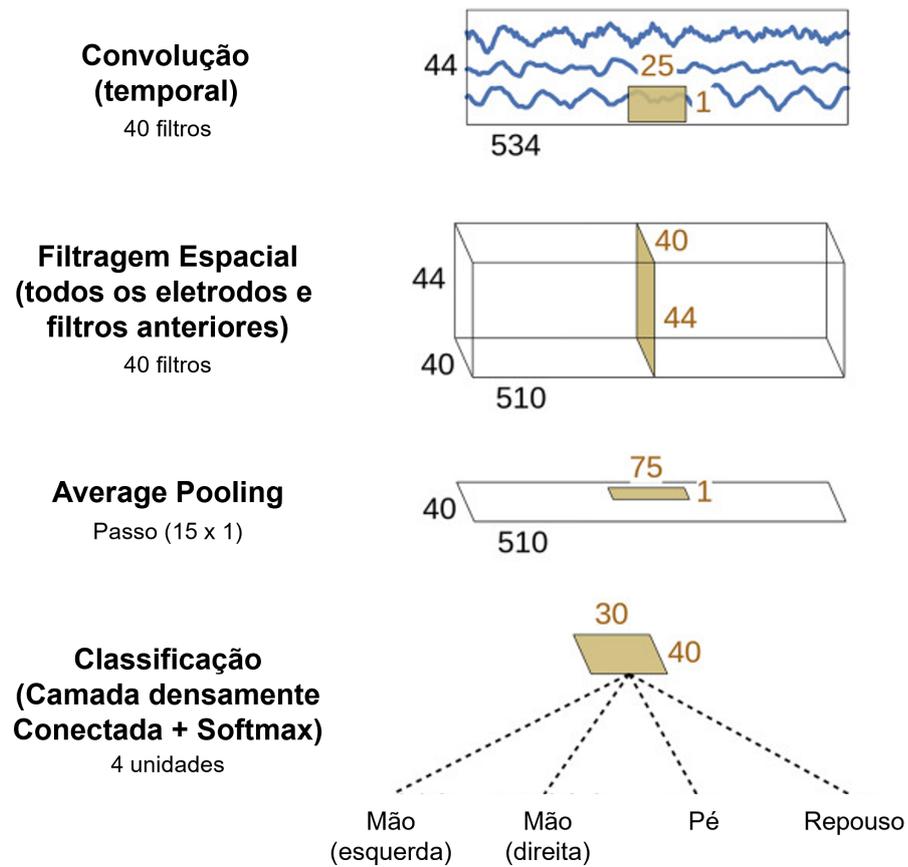


Figura 14 – Arquitetura Shallow ConvNet.

Fonte: Schirrneister et al. (2017).

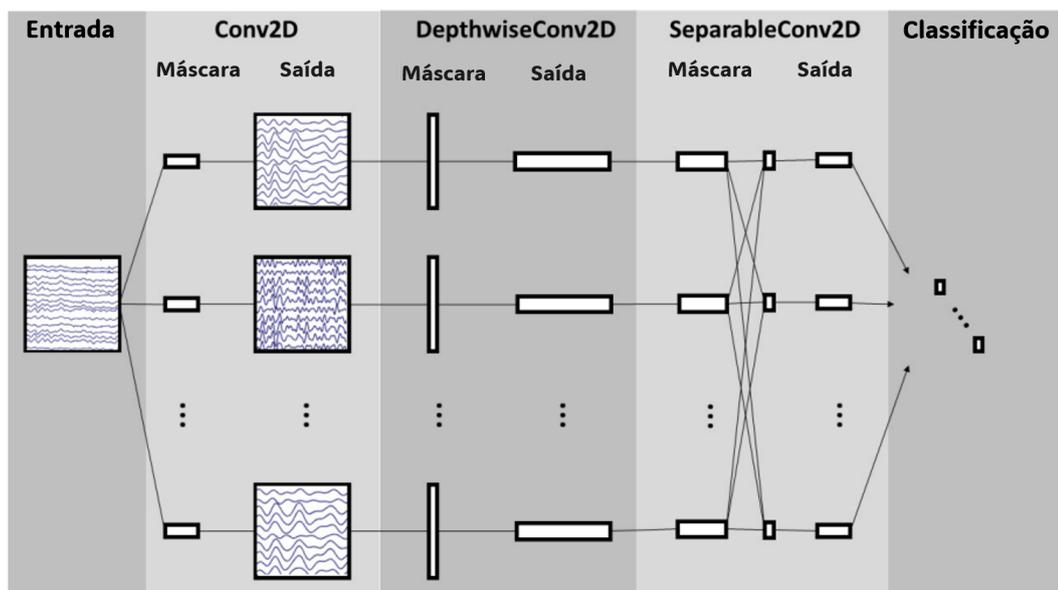


Figura 15 – Ilustração da arquitetura de rede EEGNet.

Fonte: Lawhern et al. (2016).

2.4 Fourier Transform Surrogates

Os desafios de aprendizado de máquina em domínios médicos frequentemente estão associados à escassez de dados, tanto em termos de quantidade quanto de qualidade. Esses problemas geralmente envolvem desequilíbrios nas classes, ocorrendo quando há uma disparidade na quantidade de amostras entre diferentes alvos. Para superar esses obstáculos, é comum aplicar técnicas de aumento de dados durante a fase de treinamento. Essas técnicas buscam aumentar artificialmente a quantidade de amostras de um determinado tipo, ajudando a melhorar a robustez e a capacidade de generalização dos modelos desenvolvidos (AL-HUSSAINI; MITCHELL, 2023).

Uma das técnicas disponíveis para essa tarefa de aumento de dados é o *FTSurrogate*, que utiliza da capacidade de decomposição de componentes complexos de Fourier s_n de um sinal x_n em amplitudes a_n , tal que, $s_n = a_n e^{i\Phi_n}$. Essas amplitudes definem unicamente o sinal, enquanto as fases pertencem a valores aleatórios no intervalo $[0, 2\pi)$. A partir dessas duas premissas, é possível gerar um novo sinal y_n que é estatisticamente independente de x_n a partir da substituição das fases de s_n por valores no intervalo $[0, 2\pi)$ e finalmente aplicar a transformada inversa (SCHWABEDAL et al., 2018).

2.5 Trabalhos Correlatos

Este trabalho foi inspirado no trabalho de Bissaro (2021), com posterior publicação em (BISSARO et al., 2023), onde o desafio de decodificação de sinais de EEG para prognóstico de pacientes em coma foi abordado considerando dois e três desfechos possíveis. Foram desenvolvidos modelos baseados em redes echo-state (ESN), os quais foram aplicados sobre trechos pré-selecionados do exame, com arquiteturas de processamento mono e multi-sequenciais. Além disso, em outro experimento, os dados foram alimentados à arquiteturas de redes neurais convolucionais, juntamente com a adição de uma dimensão espacial ao exame, que modela a disposição dos eletrodos no escalpo do paciente e seus relacionamentos, com base em regras de vizinhança, apresentando resultados superiores às abordagens do estado da arte para dois e três desfechos.

Diversas estratégias podem ser consideradas para decodificar o sinal de EEG. No trabalho de Ramos (2022), foi abordada a classificação de prognósticos de pacientes comatosos em dois e três desfechos a partir do cálculo de quantificadores no domínio do tempo e da frequência em dois espectros de sinal distintos, de 1 a 30Hz e de 1 a 100Hz, quando alimentados a algoritmos de aprendizado de máquina clássicos como *Support Vector Machines* (SVM), *K-Nearest Neighbours* (KNN) e *Logistic Regression* (LR). Para o prognóstico em dois desfechos, a autora considerou três experimentos, o primeiro analisa a performance de classificação com o desfecho positivo e óbito clínico (OC), a segunda observa os resultados de classificação para desfecho positivo e óbito por morte encefálica

(OME) e a terceira visa classificar os pacientes com desfecho negativo em OC e OME. Características como a etiologia do coma e escala de grau de consciência também foram considerados como atributos de entrada para os modelos, o que permitiu a obtenção de melhores resultados para a medida F1-Macro. Os classificadores SVM e KNN obtiveram a melhor performance para os dois espectros considerados. Além disso, foi possível constatar que os modelos apresentaram melhor performance quando foram combinados quantificadores no domínio do tempo e da frequência. Para o espectro de frequência 1-30Hz, o modelo com melhor performance foi obtido no experimento que classifica entre desfecho positivo e OC, com um valor de F1-Macro de 58% e desvio padrão de 16%. Os modelos obtidos no espectro de frequência 1-100Hz tiveram maior dificuldade para classificar em todos os experimentos, considerando a média de F1-Macro de 49% entre os melhores modelos. Ambos os experimentos foram impactados fortemente pela quantidade baixa de dados, desbalanceamento entre classes e o tipo de quantificação do sinal de EEG.

[Carneiro et al. \(2023\)](#) exploram a aplicação de técnicas de classificação de alto-nível para o prognóstico de pacientes em coma considerando o problema em dois e três desfechos. A metodologia envolveu a criação de uma técnica que utiliza a assortatividade e menor caminho médio para obter modelos de aprendizado confiáveis. A aplicação desta técnica foi comparada com nove outras abordagens que incluíram aplicação de redes neurais convolucionais, sendo possível observar que a arquitetura proposta têm potencial para aprimorar a performance preditiva obtidos por esses modelos.

As causas do coma geralmente são diversas. No estudo de [Zubler e Tzovara \(2023\)](#), o prognóstico de pacientes comatosos é abordado em dois desfechos, com amostras oriundas de pessoas que sofreram parada cardíaca. Foi desenvolvido uma abordagem que combina o poder das redes neurais convolucionais e redes neurais recorrentes com células Long-Short Term Memory (LSTM), para aprendizado de padrões temporais de curto e longo prazo. Foram empregados registros de EEG capturados ao longo de vários meses, os pacientes que apresentaram um indicador *Cerebral Performance Categories* (CPC) com valores entre 3 e 5 após períodos de 3 a 6 meses possuem desfecho negativo, enquanto valores de 1 a 2 são considerados desfechos positivos. A abordagem resultou em sensibilidade de 77% e 75% para desfechos positivos e negativos com especificidade em 90%, respectivamente. Com a especificidade em 99%, a sensibilidade obtida foi de 50% para desfecho negativo. Esses resultados corroboraram a ideia de que observar a evolução do sinal de EEG do paciente comatoso ao longo do tempo melhora a performance preditiva para prognóstico.

O estudo de [Schirrmeister et al. \(2017\)](#) analisou diferentes arquiteturas de ConvNets (*Shallow ConvNet*, *Deep ConvNet* e *Hybrid ConvNet*) para decodificar tarefas imaginadas ou executadas a partir de EEG bruto, revelando desempenho comparável ou superior ao de métodos convencionais. Os resultados mostraram que ConvNets aprende-

ram a usar modulações de potência espectral em diferentes frequências, além de mapear espacialmente os recursos aprendidos. Isso evidencia o potencial das ConvNets profundas combinadas com técnicas avançadas de visualização para mapeamento cerebral baseado em EEG, sem a necessidade de recursos pré-definidos. Os pesquisadores ressaltam que há um interesse crescente no uso de ConvNets profundos para análise de EEG de ponta a ponta, enfatizando a importância e relevância de estudos nesse campo.

3 Materiais e Métodos

Este capítulo descreve os materiais e métodos utilizados para a elaboração deste trabalho. O processo é resumido pela Figura 16, que começa com a coleta dos dados pelos especialistas, descritos na Seção 3.1. A Seção 3.2 apresenta as ferramentas utilizadas ao longo do estudo. Na Seção 3.3, são abordados os procedimentos de seleção e pré-processamento de dados na Subseção 3.3.1, seguidos pela montagem dos experimentos na Subseção 3.3.2. A forma como os modelos são treinados é descrita na Subseção 3.3.3. Por fim, as Subseções 3.3.4 e 3.3.5 detalham, respectivamente, a avaliação dos modelos e a coleta dos resultados obtidos neste estudo.

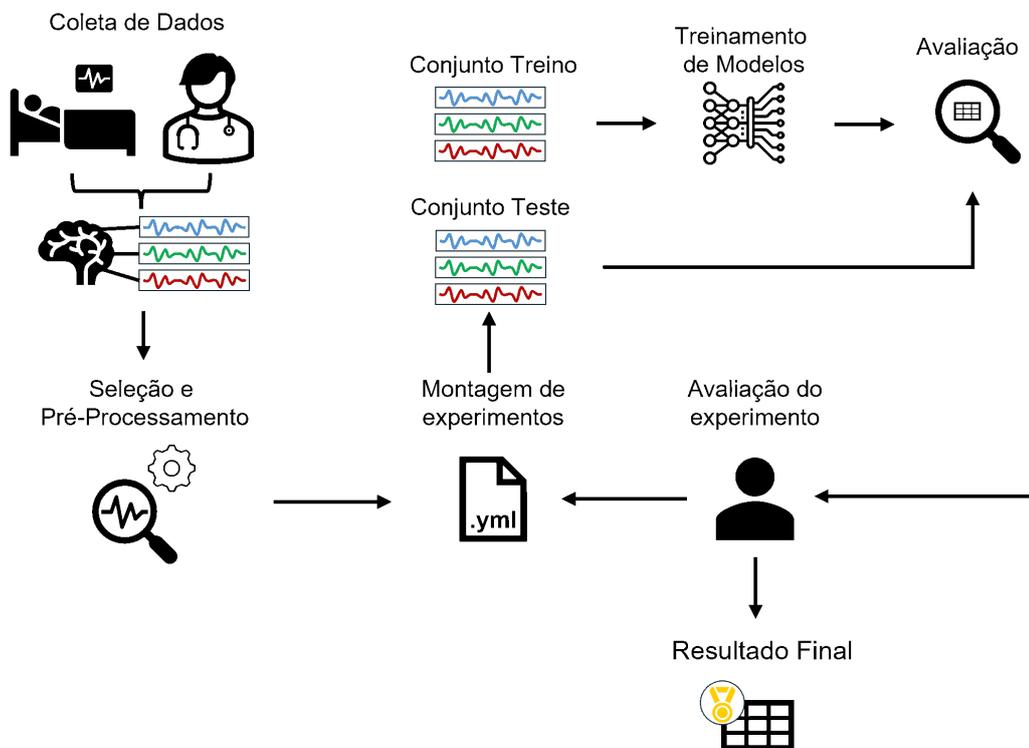


Figura 16 – Fluxograma do processo de desenvolvimento do trabalho.
Fonte: Autoria Própria.

3.1 Conjunto EEG-Coma

As amostras de traçado de EEG de pacientes comatosos foram coletadas clinicamente na unidade de tratamento intensivo adulta do Hospital de Clínicas da Universidade Federal de Uberlândia (HCU-UFU). A base de dados considerada neste estudo foi obtida a partir de dois protocolos de coleta de dados autorizados pelo comitê de ética. O primeiro protocolo, relacionado ao trabalho de Paiva et al. (2018), recebeu autorização pelo Comitê

de Ética em Pesquisas da Universidade Federal de Uberlândia (CEP/UFU) sob nº 369/11, em 2 de fevereiro de 2012. O segundo protocolo, relacionado ao trabalho de Costa (2022), no qual pacientes em coma também foram submetidos a diferentes estímulos musicais, recebeu autorização sob nº 82824017.5.0000.5152, em 2 de março de 2018.

Neste estudo, em que os modelos foram desenvolvidos com o objetivo de contribuir no prognóstico de pacientes em coma, os sinais de EEG foram categorizados em dois desfechos: favorável e desfavorável. O desfecho favorável é alcançado quando o paciente recebe alta da unidade de terapia intensiva com uma condição de saúde considerada satisfatória. Por outro lado, o desfecho desfavorável é caracterizado pela morte do paciente, seja por causas clínicas diversas ou por apresentar um quadro de morte encefálica.

Ao todo, foram utilizadas 121 amostras de traçado de EEG, que consiste em 42 amostras com desfecho favorável e 79 com desfecho desfavorável, com a frequência de amostragem dos exames variando entre 100Hz e 600Hz.

Dentro de cada registro, os especialistas em análise de sinais de EEG segmentaram 10 trechos considerados significativos para a elaboração do prognóstico do paciente. Estes segmentos possuem duração de 2 segundos, sendo considerados pelos especialistas trechos com menor incidência de artefatos e ruídos que possam prejudicar a análise do exame. Nesta monografia esses trechos selecionados pelos especialistas são denominados padrão ouro de seleção (*gold standard selection*).

Além de considerar os trechos selecionados pelos especialistas, este trabalho também investiga uma estratégia de seleção de segmentos automatizada, a partir da seleção de trechos do exame original de forma aleatória sem sobreposição dos dados, o que foi denominado de seleção aleatória (*random selection*).

3.2 Ambiente e Ferramentas

O desenvolvimento deste trabalho foi apoiado fortemente sobre a linguagem de programação Python, em sua versão 3.8.10, devido sua grande flexibilidade, agilidade e facilidade de desenvolvimento e extensa gama de bibliotecas disponíveis para as mais diversas tarefas, características que foram essenciais para esse trabalho.

O pré-processamento dos dados foi realizado com apoio da biblioteca Numpy e Scipy. A biblioteca scikit-learn foi utilizada para cálculo das medidas de avaliação e para divisão do conjunto de dados em treinamento e teste, com sua implementação de validação cruzada *K-Fold* estratificada.

A implementação das arquiteturas *Shallow ConvNet*, *Deep ConvNet* e *EEGNet* foram realizadas pelo pacote *braindecode*. O balanceamento do conjunto de dados utiliza a implementação da técnica *FTS surrogate* disponibilizada pela biblioteca, embora tenha

sido necessário desenvolvimento para adaptá-la para a geração de amostras deste trabalho. O *braindecode* utiliza outra biblioteca em seu *back-end*, denominada MNE (MEG + EEG Analysis & Visualization), que foi utilizada para padronizar as frequências de amostragem dos traçados em 100Hz e 200Hz.

Para a otimização de hiperparâmetros foi utilizada a ferramenta Optuna, um *framework* construído para abordar esse problema de maneira genérica, sendo compatível com qualquer biblioteca de aprendizado de máquina da linguagem disponível no mercado, desde que sua interface seja construída na linguagem Python. O banco de dados MySQL foi utilizado como a fonte de dados externa para armazenamento de informações.

As arquiteturas escolhidas para avaliação foram a *Deep ConvNet*, *Shallow ConvNet* e *EEGNet*, todas apoiadas, majoritariamente, em camadas convolucionais para aprendizado de padrões temporais e espaciais relevantes para o problema em estudo. Esse tipo de rede neural foi escolhida devido aos resultados positivos alcançados no trabalho de Bissaro (2021), onde foram capazes de obter uma boa performance de classificação para prognóstico de pacientes comatosos em dois desfechos.

Para uma avaliação mais confiável da performance dos modelos, optou-se pela utilização da técnica de validação cruzada *K-Fold* estratificada, considerando o prognóstico de todos os pacientes. Essa abordagem foi escolhida para preservar o desbalanceamento entre classes durante a avaliação do modelo, pois essa condição é frequentemente encontrada em problemas da área da saúde. Após o particionamento em pastas, o conjunto utilizado para treino é submetido a um algoritmo de aumento de dados realizado por uma implementação da técnica *FTSurrogate*.

O processo de treinamento ocorreu em um ambiente conectado a uma ferramenta de otimização de hiperparâmetros que, por sua vez, é conectada a uma fonte de dados externa para armazenamento dos experimentos. Os parâmetros que foram estudados neste trabalho são inseridos como pares chave-valor de arquivos de texto em formato YAML (YAML Ain't Markup Language™), com suas configurações iniciais sendo levantadas por meio de experimentação manual. Durante esse processo, os trechos selecionados pelos especialistas são divididos em 9 subamostras individuais, cada uma com seu respectivo rótulo original, para cumprir com o formato de entrada aceito pelas arquiteturas avaliadas. O mesmo procedimento é repetido na etapa de avaliação de performance, com o prognóstico final do paciente sendo obtido pela moda da classificação de todos os trechos de seu exame.

As métricas de avaliação incluem acurácia, especificidade, sensibilidade e F1-Score, sendo utilizadas para ranqueamento da performance, devido ao foco de encontrar modelos que melhor identifiquem pacientes com maior probabilidade de recuperação do coma e que apresentem um desempenho equilibrado na classificação em dois desfechos. Ao final, é calculado a média e desvio padrão de todas as medidas, que são armazenadas e enviadas

à ferramenta de otimização e extração de novos valores para experimentação.

3.3 Aprendizado de Sinais EEG usando FBCSP e Redes Convolucionais

Este capítulo tem como objetivo resumir o processo desenvolvido neste trabalho. A Subseção 3.3.1 apresenta os procedimentos adotados para a seleção e o pré-processamento de dados. A Subseção 3.3.2 detalha a montagem dos experimentos automatizados. A Subseção 3.3.3 aborda a separação das amostras em conjuntos de treino e teste, além do treinamento dos modelos. Por fim, a Subseção 3.3.4 descreve as métricas de avaliação escolhidas para as arquiteturas desenvolvidas, enquanto a Subseção 3.3.5 explica como os resultados foram coletados e tabelados utilizando a ferramenta de otimização de hiperparâmetros.

3.3.1 Seleção e pré-processamento

Foram investigadas duas abordagens para o pré-processamento dos dados. Na primeira, utilizaram-se 9 de 10 trechos de traçados de EEG previamente selecionados por especialistas, de modo que não foi necessário o desenvolvimento de técnicas de filtragem ou seleção. Na segunda abordagem, 9 trechos foram selecionados aleatoriamente a partir do traçado original do exame, sem sobreposição entre os dados.

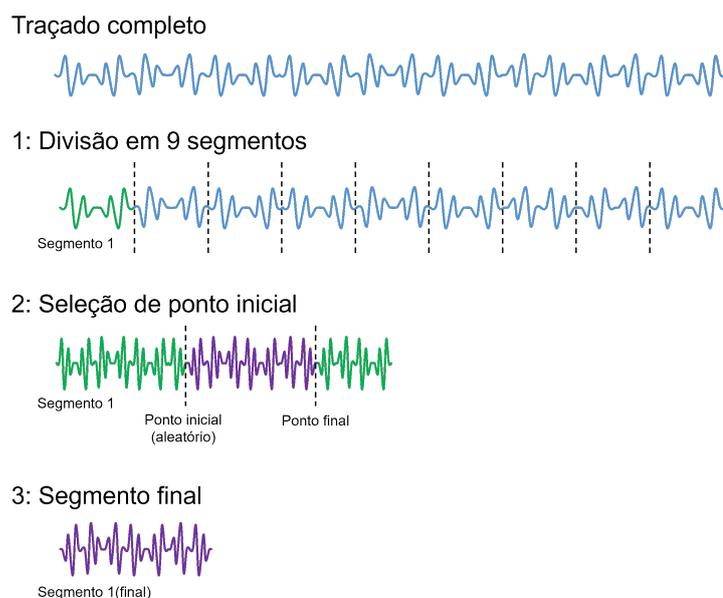


Figura 17 – Seleção de dados aleatória.
Fonte: Autoria Própria.

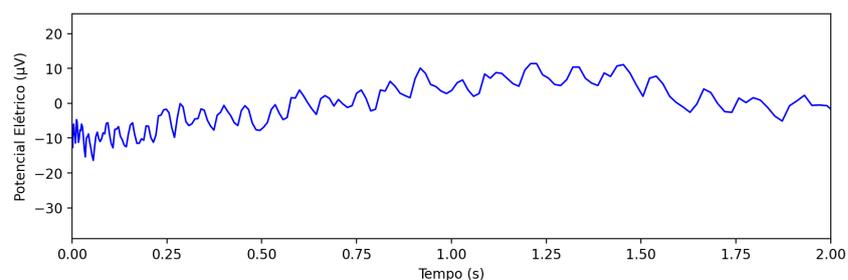
A Figura 17 ilustra os três passos do procedimento de seleção aleatória de segmentos, aplicado ao traçado de EEG original de um paciente. O processo começa com a

divisão do exame em nove segmentos de tamanho igual, conforme mostrado no passo 1. No passo 2, um ponto inicial é sorteado, e o ponto final é calculado de modo a formar uma janela de 2 segundos, sem sobreposição com o início de outros segmentos. No último passo, esses dois pontos são utilizados para extrair cada segmento do exame original, que é então armazenado em um conjunto de amostras.

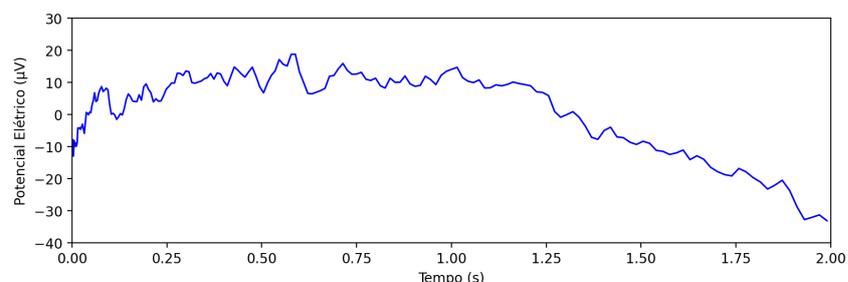
O eletroencefalograma, apesar de apresentar uma capacidade de capturar informações valiosas acerca da atividade elétrica do cérebro do paciente e possuir custo acessível, costuma gerar sequências temporais com grande quantidade de ruídos, geralmente capturados do ambiente em que o paciente está inserido (DELORME, 2023). Embora já existam abordagens automatizadas e semi-automatizadas, não há uma maneira universal de se processar dados de EEG, sendo que a escolha pode afetar o resultado do estudo realizado (COELLI et al., 2024).

O conjunto de dados utilizado é composto de amostras que foram capturadas com configurações de amostragem variadas. O pré-processamento aplicado no conjunto consiste em realizar o *downsampling* ou *upsampling* das amostras para a frequência de amostragem de 100Hz e 200Hz, considerando assim dois conjuntos de dados para a pesquisa.

A implementação utilizada, fornecida pela biblioteca MNE, aplica um filtro passa-baixa antes de realizar a seleção de pontos ou interpolação, para *downsampling* e *upsampling*, respectivamente. Na Figura 18a, é exibida a atividade elétrica do eletrodo de um paciente em coma, capturada com uma frequência de amostragem de 400Hz, enquanto na Figura 18b é mostrada a série temporal após a aplicação do *downsampling* para 100Hz.



(a)



(b)

Figura 18 – Eletrodo FP2 de amostra com frequência original de amostragem 400Hz (a) e após *downsampling* para 100Hz (b).

3.3.2 Montagem de Experimentos

Uma das tarefas inerentes ao desenvolvimento de arquiteturas de aprendizado de máquina eficazes é a otimização de seus hiperparâmetros. Esses parâmetros são configurados no momento de instanciação dos modelos e, ao contrário dos pesos, não podem ser ajustados durante o processo de treinamento. No entanto, essa atividade geralmente consome uma grande quantidade de tempo, recursos computacionais e humanos, devido a grande quantidade de combinações possíveis que devem ser avaliadas, e que tendem a aumentar com a complexidade dos modelos (YANG; SHAMI, 2020).

Inicialmente, os experimentos nesse trabalho foram realizados de maneira interativa, por meio da ferramenta Jupyter Notebook, com o objetivo de explorar as funcionalidades e possibilidades da biblioteca *braindecode*. Entretanto, essa estratégia se mostrou ineficaz para avaliação e a busca por hiperparâmetros que maximizam a performance dos modelos.

Para simplificar o processo de otimização de hiperparâmetros, foi adotada a biblioteca Optuna, que disponibiliza uma interface programática genérica em Python, de fácil uso e compreensão. Nela, é disponibilizado uma abstração *Study*, que representa um estudo de otimização, definindo as métricas a serem maximizadas ou minimizadas, e *Trial*, que representa cada iteração do processo. Dentro de cada iteração, é possível solicitar sugestões de valores categóricos, inteiros ou pontos flutuantes ao *framework*, que retorna sugestões com base nos parâmetros e resultados obtidos anteriormente.

Entretanto, para a utilização de parâmetros e modelos diferentes, ainda é preciso criar outros arquivos de código-fonte. A depender da quantidade de experimentos realizados e da diversidade dos hiperparâmetros, essa estratégia pode gerar desorganização e duplicação de forma rápida, podendo levar a lentidão nos momentos em que haja necessidade de modificações ou observação de resultados, prejudicando a pesquisa como um todo.

Para atender a essa necessidade, foi desenvolvida a abstração *ExperimentConfiguration*, cujo propósito é facilitar a configuração das sugestões da biblioteca através de arquivos de configuração em texto YAML, um formato de serialização chave-valor compatível com todas as linguagens de programação disponíveis no mercado. A escolha desse formato se deu pela sua facilidade de compreensão e alta legibilidade, tornando-o ideal para lidar com a variabilidade dos hiperparâmetros das arquiteturas estudadas neste trabalho. Na Figura 19 é mostrado um exemplo compacto de um arquivo de configuração para a arquitetura *Shallow ConvNet*.

As chaves localizadas no topo da estrutura, no nível mais alto, representam os parâmetros de inicialização do experimento e de configuração para a leitura dos seus dados. Ao final, é estabelecido o *random_state*, um valor semente crucial para segmentos

```
1 name: ShallowFBCSPNet_FTSurrogate_100Hz
2 model: ShallowFBCSPNet
3 trials: 200
4 n_folds: 10
5 sampling_frequency: 100
6 module_input:
7   n_filters:
8     type: options
9     choices: [8, 10, 14, 18, 24, 30]
10    map_to: [n_filters_spat, n_filters_time]
11  final_conv_length:
12    type: value
13    value: 'auto'
14  lr:
15    type: float
16    range:
17      low: 0.0002
18      high: 0.09
19      step: 0.0006
20  max_epochs:
21    type: value
22    value: 300
23  batch_size:
24    type: options
25    choices: [11, 32, 64]
26  random_state: 10000
```

Figura 19 – Arquivo .YAML de Configuração do Experimento.
Fonte: Autória Própria.

de código ou funções que demandam geração de números aleatórios, como a inicialização de pesos dos modelos ou geração de novas amostras. A configuração desse parâmetro é extremamente importante, pois garante a reprodução consistente de todas as etapas deste estudo.

Os hiperparâmetros das arquiteturas são definidos na chave *module_input* e o exemplo fornecido destaca as possibilidades cobertas por essa abstração. Um hiperparâmetro com tipo *options* requer uma lista de valores para serem escolhidos pela biblioteca. A propriedade *map_to* não tem relação com o *framework*, mas é uma ferramenta para mapear um valor sugerido para vários hiperparâmetros. Além disso, é possível sugerir valores inteiros ou de ponto flutuante dentro de intervalos específicos ou distribuições, utilizando os tipos *int* e *float*, respectivamente. Por fim, os hiperparâmetros do tipo *value* são valores fixos que não são tratados por nenhuma rotina da biblioteca.

Essa aplicação é fortemente inspirada em soluções de mapeamento e serialização de objetos presentes em várias linguagens de programação. Um exemplo comum pode ser observado nos aplicativos móveis, onde um *smartphone* envia informações serializadas em um formato textual, geralmente em Javascript Object Notation (JSON), que são recebidas por um servidor e convertidas em comandos ou objetos internos daquela linguagem. Nesse estudo, os valores inseridos no arquivo YAML são convertidos na execução de comandos

da biblioteca Optuna. Dessa forma, os modelos e hiperparâmetros a serem treinados são definidos em tempo de execução, sem modificações no código fonte.

A Tabela 1 apresenta as opções consideradas para otimização da arquitetura *Deep ConvNet*. Os hiperparâmetros $n_filters_time$ e $n_filters_spat$ se referem às duas primeiras camadas de convolução temporal e espacial da arquitetura. Os parâmetros $n_filters_2$, $n_filters_3$ e $n_filters_4$, definem a quantidade de filtros das três camadas de convolução subsequentes. O número de filtros das camadas de entrada aumenta conforme a rede se aprofunda, o que permite capacitar ao modelo aprender características mais globais do sinal à medida que avança nas camadas convolucionais.

Tabela 1 – Parâmetros considerados para otimização da arquitetura Deep ConvNet.

Seleção	Freq. Amostragem	Parâmetro	Valores
Pré-seleção	100Hz	$n_filters_spat$	10, 15, 20, 14, 18, 22
		$n_filters_time$	10, 15, 20, 14, 18, 22
		$n_filters_2$	24, 26, 30, 32
		$n_filters_3$	36, 42, 30, 32
		$n_filters_4$	34, 38, 40, 50, 55
		$drop_prob$	0.35, 0.50, 0.60
		$learning_rate$	low: 0.001 high: 0.099 step: 0.0015
		$batch_size$	11, 32, 48, 64, 72
Pré-seleção	200Hz	$n_filters_spat$	24, 28, 22, 20
		$n_filters_time$	24, 28, 22, 20
		$n_filters_2$	35, 42, 34, 36, 30
		$n_filters_3$	52, 58, 64, 42, 48, 54, 58, 60
		$n_filters_4$	64, 68, 70, 58, 60, 48, 52
		$drop_prob$	0.35, 0.50, 0.60
		$learning_rate$	low: 0.001, 0.0001 high: 0.099, 0.099 step: 0.0015, 0.0008
		$batch_size$	32, 48, 64, 72
Aleatória	100Hz	$n_filters_spat$	10, 15, 20
		$n_filters_time$	10, 15, 20
		$n_filters_2$	30, 32
		$n_filters_3$	36, 42
		$n_filters_4$	50, 55
		$drop_prob$	0.35, 0.50, 0.60
		$learning_rate$	low: 0.000001, high: 0.0009, step: 0.000008
		$batch_size$	11, 32, 64
Aleatória	200Hz	$n_filters_spat$	24, 28
		$n_filters_time$	24, 28
		$n_filters_2$	35, 42
		$n_filters_3$	52, 58, 64
		$n_filters_4$	64, 68, 70
		$drop_prob$	0.35, 0.50, 0.60
		$learning_rate$	low: 0.0001, high: 0.099, step: 0.00065
		$batch_size$	11, 32, 64

A Tabela 2 exhibe a configuração do estudo para otimização de hiperparâmetros

da arquitetura *Shallow ConvNet*. Devido aos modelos gerados por essa arquitetura apresentarem menor profundidade em comparação à *Deep ConvNet*, o custo computacional para treinamento é reduzido em relação às demais. Isso permitiu uma busca mais ampla de hiperparâmetros, como o *pool_time_stride* (tamanho do passo da máscara na camada de pooling), *filter_time_length* (tamanho do filtro da convolução temporal) e *pool_time_length* (tamanho da janela para o pooling realizado na convolução temporal).

Tabela 2 – Parâmetros considerados para otimização da arquitetura Shallow ConvNet

Seleção	Freq. Amostragem	Parâmetro	Valores
Pré-seleção	100Hz	n_filters_spat	8, 10, 14, 18, 24, 30, 12
		n_filters_time	8, 10, 14, 18, 24, 30, 12
		filter_time_length	21, 39, 52, 63, 69, 86, 15, 20, 25
		pool_time_stride	7, 9, 12, 21, 18, 15
		pool_time_length	80, 40, 50, 83, 60, 100, 35, 45
		drop_prob	0.35, 0.50, 0.60
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64
Pré-seleção	200Hz	n_filters_spat	18, 24, 28, 30, 20, 22
		n_filters_time	18, 24, 28, 30, 20, 22
		filter_time_length	20, 25, 30, 35, 40
		pool_time_stride	10, 15
		pool_time_length	40, 50, 55
		drop_prob	0.35, 0.50, 0.60
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64
Aleatória	100Hz	n_filters_spat	10, 12, 14, 16, 18, 22, 25
		n_filters_time	10, 12, 14, 16, 18, 22, 25
		filter_time_length	15, 20, 25
		pool_time_stride	10, 15
		pool_time_length	35, 40, 45, 50
		drop_prob	0.35, 0.50, 0.60
		learning_rate	low: 0.0002, high: 0.09, step: 0.0006
		batch_size	11, 32, 48, 64
Aleatória	200Hz	n_filters_spat	18, 24, 28, 30
		n_filters_time	18, 24, 28, 30
		filter_time_length	20, 25, 30, 35, 40
		pool_time_stride	10, 15
		pool_time_length	40, 50, 55
		drop_prob	0.35, 0.50, 0.60
		learning_rate	low: 0.0002, high: 0.09, step: 0.0006
		batch_size	11, 32, 48, 64

A Tabela 3 mostra os parâmetros considerados para os experimentos que envolvem a arquitetura *EEGNet*. O hiperparâmetro F1 é a quantidade de filtros temporais de saída do primeiro bloco convolucional, esse valor é multiplicado pelo hiperparâmetro D (multiplicador de profundidade), e é utilizado para definir o tamanho da saída da camada

convolucional seguinte. Por fim, esses valores são utilizados em conjunto com o parâmetro F2 para definir a quantidade de filtros da camada convolucional de tipo *Separable*.

Tabela 3 – Parâmetros considerados para otimização da arquitetura EEGNet

Seleção	Freq. Amostragem	Parâmetro	Valores
Pré-seleção	100Hz	F1	8, 12, 16
		D	1, 2, 3
		F2	16, 18, 22, 26
		kernel_length	32, 48, 64, 72
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64, 72
Pré-seleção	200Hz	F1	16, 18, 20, 26
		D	4, 5, 6
		F2	20, 24, 28, 36, 40
		kernel_length	32, 48, 64, 72
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64, 72
Aleatória	100Hz	F1	8, 12, 16
		D	1, 2, 3
		F2	16, 18, 22, 26
		kernel_length	32, 48, 64, 72
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64, 72
Aleatória	200Hz	F1	16, 18, 20, 26
		D	4, 5, 6
		F2	20, 24, 28, 36, 40
		kernel_length	32, 48, 64, 72
		learning_rate	low: 0.0002 high: 0.09 step: 0.0006
		batch_size	11, 32, 48, 64, 72

3.3.3 Treinamento de Modelos

Após a definição do experimento, o programa é inicializado a partir da linha de comando com dois parâmetros obrigatórios: o caminho do arquivo .yml do experimento e a estratégia de seleção de dados. As opções de seleção são *RANDOM_SAMPLES*, para seleção aleatória, ou *NEUROLOGIST_SAMPLES*, para segmentos selecionados por especialistas. Em seguida, o programa se conecta à ferramenta de otimização de hiperparâmetros e ao seu banco de dados.

A partir desse ponto, o código desenvolvido coleta os parâmetros sugeridos pela ferramenta de otimização, iniciando um *Trial*. Em sequência, os dados dos pacientes, a

frequência de amostragem e os traçados brutos são descartados, mantendo-se apenas um vetor *numpy*, que armazena cada segmento e seu respectivo desfecho do coma.

Logo após o início do experimento, o vetor de amostras é submetido a um algoritmo que modela uma técnica de validação. Para este trabalho, a técnica escolhida foi a *K-Fold Cross Validation*, que consiste em dividir o conjunto de dados em K partições ou “pastas”, de tamanho igual ou similar.

Após a separação das pastas, o *loop* de treinamento é iniciado. A cada iteração do processo de treinamento, um dos grupos é separado para treinamento, enquanto os $K - 1$ grupos são utilizados como entrada para o modelo, como ilustrado na Figura 20. Ao final, é garantido que todas as amostras do conjunto de dados são alimentadas ao algoritmo e testadas, com as medidas de avaliação sendo agrupadas e sumarizadas para se avaliar a performance geral do modelo.

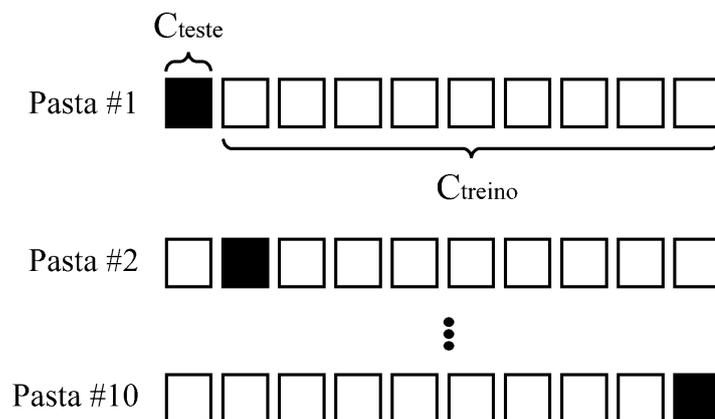


Figura 20 – K-Fold Cross Validation com valor de K igual a 10.
Fonte: Berrar (2019).

O resultado da aplicação do *K-Fold Cross Validation*, em cada iteração, é a obtenção de um conjunto de treino e outro de teste. Essa separação é fundamental para este trabalho, pois garante que as técnicas de aumento de dados sejam aplicadas exclusivamente durante o treinamento, evitando que amostras criadas pelo processo de aumento influenciem o desempenho dos modelos desenvolvidos.

A distribuição de classes do conjunto de dados revela uma proporção de aproximadamente 2 objetos de desfecho desfavorável para cada amostra de desfecho favorável. Essa disparidade caracteriza um problema de balanceamento de dados. Para contornar essa situação, foi escolhida a técnica de aumento de dados *FTS surrogate* de maneira semelhante aos trabalhos de Al-Hussaini e Mitchell (2023) e Schwabedal et al. (2018).

Essa técnica é aplicada a cada um dos 20 eletrodos presentes nos traçados de EEG utilizados neste trabalho, considerando os valores padrão para os parâmetros *phase_noise_magnitude* e *channel_indep*, resultando em um conjunto de treino com o mesmo número de amostras com desfecho favorável e desfavorável. A Figura 21a mostra os valores origi-

nais da série temporal do eletrodo F7 de um paciente comatoso. Disposta logo abaixo, a Figura 21b apresenta o resultado da aplicação de *FTSurrogate* na mesma série temporal. Por fim, na Figura 21c, os dois sinais são sobrepostos sobre o mesmo gráfico para melhor visualização de suas características e semelhanças.

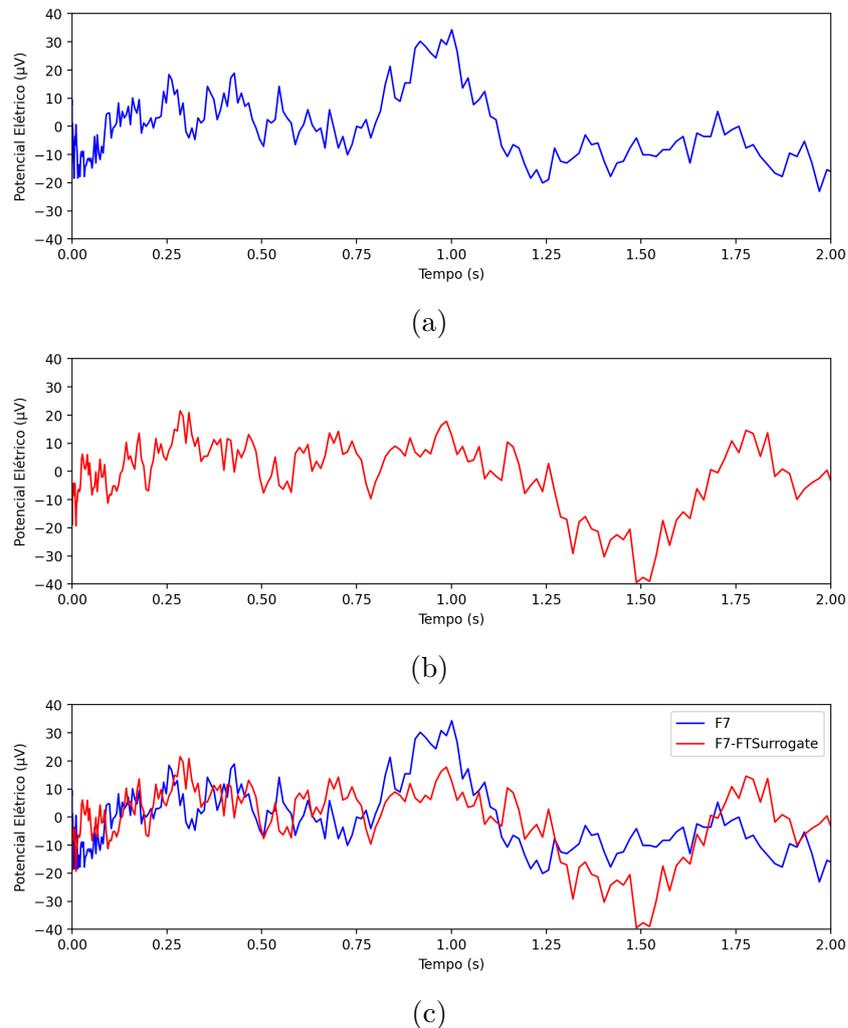


Figura 21 – Exemplo de aplicação da técnica *FTSurrogate*: (a) Trecho de exame de EEG, eletrodo F7, de paciente comatoso; (b) Trecho de exame de EEG gerado a partir de *FTSurrogate* aplicada ao eletrodo F7; e (c) Trecho de exame original sobreposto ao sinal gerado após aplicação de *FTSurrogate*.

Por fim, o vetor de treino *numpy* é convertido para uma estrutura *Epoch*, compatível com a biblioteca MNE e *braindecode*, e o processo de treinamento é iniciado.

3.3.4 Avaliação de Modelos

Após o treinamento de um modelo com o algoritmo ou arquitetura selecionados ao final de uma iteração do *K-Fold Cross Validation*, todas as amostras reservadas para o conjunto de teste são submetidas ao processo de avaliação. Os resultados das amostras de teste são usados para gerar uma matriz 2×2 , conhecida como matriz de confusão,

em que as linhas representam os rótulos reais de cada amostra, e as colunas, as classes preditas pelo modelo.

		Valor Predito	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 22 – Exemplo de matriz de confusão.
Fonte: [Nogare \(2020\)](#).

As células da matriz mostrada na Figura 22 representam as previsões em relação aos rótulos reais. Os objetos com rótulos correspondentes são denominados verdadeiros positivos (VP) e verdadeiros negativos (VN), quando são corretamente classificados como positivos e negativos, respectivamente. Por outro lado, os objetos com previsão de classes discrepantes são chamados de falsos positivos (FP) e falsos negativos (FN), quando são erroneamente classificados como positivos e negativos, respectivamente.

A partir dessa matriz, podemos calcular a performance do modelo desenvolvido. No entanto, apenas observar a quantidade de objetos em verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN) não é suficiente. Para obter uma interpretação mais completa, foi necessário utilizar medidas de avaliação como acurácia (3.1), sensibilidade (3.2), especificidade (3.3) e f1-score 3.4 (JUNIOR, 2023).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.1)$$

$$Sensibilidade = \frac{VP}{VP + FN} \quad (3.2)$$

$$Especificidade = \frac{VN}{VN + FP} \quad (3.3)$$

$$F1-Score = \frac{2}{\frac{1}{sensibilidade} + \frac{1}{especificidade}} \quad (3.4)$$

A acurácia (3.1) representa a proporção total de previsões corretas feitas pelo modelo. A sensibilidade (3.2) indica a proporção de previsões corretas feitas pelo modelo para os objetos cuja classe real é positiva. Por outro lado, a especificidade (3.3) mostra a proporção de amostras classificadas corretamente como pertencentes à classe negativa

entre os objetos cuja classe real é negativa. O F1-Score é frequentemente compreendido como a média harmônica da precisão; no entanto, neste trabalho, ele é utilizado como a média harmônica entre a sensibilidade e a especificidade, visando valorizar modelos que apresentem bons resultados na classificação de ambos os desfechos.

Os modelos desenvolvidos são alimentados com cada segmento do traçado de EEG do paciente, que atribui um desfecho a cada trecho. No contexto do problema abordado neste trabalho, o desfecho é determinado pela combinação de todos os trechos de traçado do paciente. Diante dessa diferença, considera-se que o prognóstico do paciente é obtido a partir da moda do vetor de rótulos de todos os segmentos de seu exame. Esse rótulo é adicionado a outro vetor de resultados, que forma a matriz de confusão que será utilizada para cálculo das medidas de avaliação.

Ao final desse processo, os resultados das medidas de avaliação são enviados para a ferramenta de otimização de hiperparâmetros. Em específico, as métricas F1-Score e Sensibilidade são fornecidas como argumentos para que a ferramenta utilize na definição dos valores de hiperparâmetros do próximo *Trial*.

3.3.5 Avaliação de Experimentos

A otimização de hiperparâmetros pode exigir um longo período de tempo e um grande número de execuções para identificar parâmetros promissores no treinamento das arquiteturas estudadas. Além disso, a quantidade de informações geradas durante o processo pode ser extensa e difícil de interpretar. Neste estudo, os resultados de cada *Trial* são armazenados pela ferramenta, juntamente com outras informações dos experimentos, e posteriormente acessados e analisados por meio de um painel de dados disponível no navegador.

As informações disponibilizadas pelo painel podem ser acessadas tanto durante a execução quanto após os estudos atingirem o limite de *Trials* definido no arquivo de experimento. A Figura 23 mostra a página inicial da ferramenta, que exibe todos os experimentos já iniciados. Os experimentos foram nomeados com base na arquitetura, técnica de aumento de dados, frequência de amostragem e estratégia de seleção de dados. Por exemplo, o nome *Deep4Net_FTSurrogate_100HZ_Final_Neurologist* indica que o modelo treinado é uma *Deep ConvNet*, com dados padronizados pelo processo de *resampling* em 100 Hz, e que as amostras utilizadas foram selecionadas por especialistas.

Ao clicar em qualquer um dos experimentos, uma página é aberta com uma série de visualizações das informações geradas. Entre elas, destaca-se o histórico, mostrado na Figura 24, que exibe o desempenho do modelo ao longo do experimento. Cada círculo azul representa o valor da métrica F1-macro ou sensibilidade, correspondendo aos valores *Objective 0* e *Objective 1* escolhidos na caixa de seleção, respectivamente. A linha vermelha

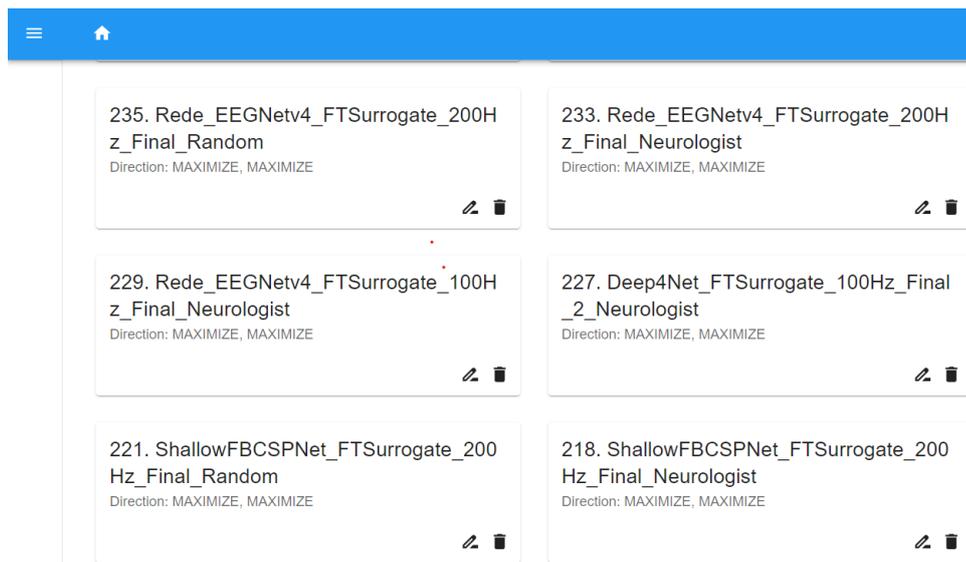


Figura 23 – Página inicial da ferramenta Optuna.
Fonte: Autoria própria.

conecta os valores das métricas aos *Trial* que obtiveram melhor desempenho, e foram usados para extrair os resultados discutidos neste trabalho.



Figura 24 – Histórico de experimento.
Fonte: Autoria própria.

A Figura 25, por sua vez, apresenta uma visualização da importância de cada hiperparâmetro na obtenção de melhores valores de F1-macro ou sensibilidade em um estudo. Nesse caso, observa-se que o hiperparâmetro "lr", correspondente à taxa de aprendizado dos modelos, foi considerado o mais relevante para melhorar os valores de F1-macro. Já os hiperparâmetros *n_filters* e *filter_time_length* mostraram-se mais importantes para otimizar a métrica de sensibilidade.

Por fim, a página exibida na Figura 26 pode ser acessada pelo painel lateral ou ao clicar em qualquer círculo do gráfico mostrado na Figura 24. Ela apresenta o detalhamento do experimento, incluindo os valores das demais métricas de avaliação, bem como os valores originais do vetor de predições, permitindo uma posterior verificação.

A coleta de resultados deste trabalho foi realizada por meio da inspeção das visualizações de histórico e detalhamento de cada experimento. Todos os resultados considerados

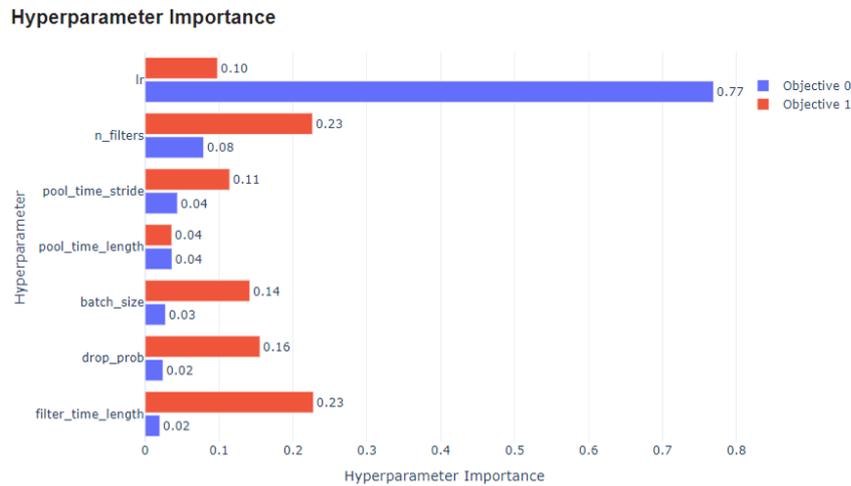


Figura 25 – Importância de hiperparâmetros.

Fonte: Autoria própria.

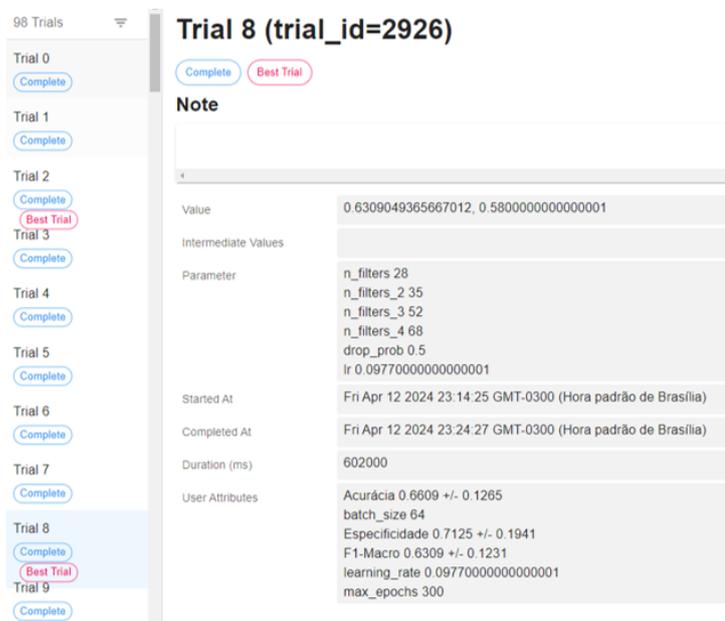


Figura 26 – Detalhamento de experimentos.

Fonte: Autoria própria.

promissores, ou seja, marcados como *Best Trial*, foram reunidos e registrados em uma planilha. Após a conclusão de todos os experimentos, os resultados foram priorizados com base nos melhores valores de F1-macro, seguidos pela sensibilidade. Além disso, foram priorizados os registros que apresentaram o menor desvio padrão nas métricas de F1-macro, sensibilidade, especificidade e acurácia.

4 Resultados e Discussão

Os resultados obtidos são apresentados neste capítulo e consideram as diferentes arquiteturas de redes neurais convolucionais aplicadas ao conjunto de dados. Cada arquitetura é especificada com a frequência de amostragem considerada no pré-processamento do conjunto de dados (100Hz ou 200Hz). Os resultados são apresentados em porcentagem, com média e desvio-padrão, em relação às várias métricas de desempenho, incluindo acurácia, especificidade, sensibilidade e F1-Score.

4.1 Resultados com seleção de trechos por especialistas

A Tabela 4 apresenta os resultados alcançados na classificação dos trechos de traçado de EEG selecionados pelos especialistas, considerando frequências de amostragem de 100Hz e 200Hz. Na tabela, a arquitetura *EEGNet* demonstrou um desempenho satisfatório quando os dados foram amostrados a 100Hz. No entanto, ao analisar as três arquiteturas, verificou-se que esta registrou os piores indicadores.

Tabela 4 – Resultados de modelos que utilizam trechos selecionados por especialistas em porcentagem, com média e desvio-padrão.

Arquitetura	EEG-Coma	Acurácia	Especificidade	Sensibilidade	F1-Score
EEGNet	100Hz	67,05 \pm 12,46	65,00 \pm 20,00	72,00 \pm22,16	63,60 \pm 15,53
EEGNet	200Hz	63,59 \pm 10,11	61,25 \pm 22,67	70,00 \pm 24,90	57,43 \pm 15,61
Shallow ConvNet	100Hz	70,45 \pm 10,87	71,25 \pm 19,41	70,50 \pm 20,30	66,62 \pm11,31
Shallow ConvNet	200Hz	67,95 \pm 09,80	72,50 \pm 20,00	61,00 \pm 24,58	59,63 \pm 15,01
Deep ConvNet	100Hz	68,65 \pm 11,43	70,00 \pm 18,71	67,00 \pm 30,59	60,36 \pm 23,36
Deep ConvNet	200Hz	73,59 \pm10,93	79,82 \pm18,64	63,00 \pm 18,73	66,53 \pm 11,52

Entre as opções de frequência de amostragem, a *Shallow ConvNet* apresentou bons resultados, com um F1 de 66,62% e uma sensibilidade de 70,50% em 100Hz, e 59,63% e 61% em 200Hz. Por fim, a arquitetura *Deep ConvNet* também obteve resultados satisfatórios, com uma sensibilidade de 67% e um F1-Score de 60,36% em 100Hz, e 63% e 66,53% em 200Hz. No entanto, observa-se uma grande variação na sensibilidade dessa arquitetura com frequência de 100Hz, com um desvio-padrão significativamente alto de \pm 30,59% na sensibilidade e \pm 23,36% na F1-Score, indicando certa instabilidade na capacidade do modelo em identificar corretamente os verdadeiros positivos.

Ao comparar estas duas últimas arquiteturas, é possível perceber que a arquitetura *Deep ConvNet* conseguiu melhores resultados quando alimentada com séries temporais

de maior resolução, enquanto a *Shallow ConvNet* teve um desempenho superior quando exposta a dados com menor resolução, o que indica que a maior quantidade de blocos convolucionais da *Deep ConvNet* permitiu uma melhor utilização da maior dimensionalidade, quando os dados foram amostrados em 200Hz.

Os resultados apresentados sugerem que a frequência de amostragem das arquiteturas impacta significativamente seu desempenho. Considerando a frequência de amostragem de 100Hz, as arquiteturas *EEGNet* e *Shallow ConvNet* tiveram desempenhos mais consistentes em comparação com suas análises de 200Hz. A *Shallow ConvNet* em 100Hz apresentou uma acurácia média de 70,45% com um desvio-padrão de $\pm 10,87\%$, enquanto a *EEGNet* em 100Hz obteve uma acurácia média de 67,05% com um desvio-padrão de $\pm 12,46\%$, sugerindo uma boa estabilidade em suas performances.

A arquitetura *Deep ConvNet* com frequência de amostragem de 200Hz, destacou-se pela maior acurácia e especificidade, ao mesmo tempo em que manteve um desempenho comparável à *Shallow ConvNet* em 100Hz. Enquanto isso, a *Shallow ConvNet* demonstrou o melhor desempenho global, com acurácia, especificidade e sensibilidade superiores a 70%, além de alcançar o melhor resultado de F1 neste experimento.

4.2 Resultados com trechos selecionados aleatoriamente

A Tabela 5 apresenta os resultados com trechos selecionados aleatoriamente. Considerando a frequência de amostragem de 100Hz, a arquitetura *Shallow ConvNet* apresentou a maior acurácia média (70,32%) entre todas as arquiteturas nessa frequência, destacando-se pela alta sensibilidade média (74,5%) e F1-Score médio (66,64%). A *Deep ConvNet* exibiu uma especificidade média maior (79,64%) e com menor variação, com a maior capacidade de identificar corretamente os verdadeiros negativos, contudo apresentou baixa especificidade.

Tabela 5 – Resultados de modelos que utilizam trechos selecionados aleatoriamente, com média e desvio-padrão

Arquitetura	EEG-Coma	Acurácia	Especificidade	Sensibilidade	F1-Score
EEGNet	100Hz	58,59 $\pm 15,52$	68,57 $\pm 22,39$	40,50 $\pm 42,69$	33,62 $\pm 30,81$
EEGNet	200Hz	67,88 $\pm 09,56$	78,75 $\pm 18,58$	47,50 $\pm 26,48$	51,80 $\pm 21,23$
Shallow ConvNet	100Hz	70,32 $\pm 11,62$	68,75 $\pm 21,83$	74,50 $\pm 18,90$	66,64 $\pm 13,38$
Shallow ConvNet	200Hz	72,76 $\pm 12,84$	75,00 $\pm 16,77$	69,50 $\pm 21,27$	69,15 $\pm 15,95$
Deep ConvNet	100Hz	68,65 $\pm 09,43$	79,64 $\pm 12,93$	46,50 $\pm 25,50$	52,93 $\pm 23,47$
Deep ConvNet	200Hz	61,92 $\pm 13,14$	56,07 $\pm 28,99$	75,00 $\pm 22,80$	53,45 $\pm 21,16$

Considerando a frequência de amostragem de 200Hz, a arquitetura *Shallow ConvNet* demonstrou o melhor desempenho global, com a mais alta acurácia (72,76%) e F1-

Score (69,15%) entre todas as arquiteturas nessa frequência. O modelo também manteve uma alta sensibilidade média (69,50%). As arquiteturas *EEGNet* e *Shallow ConvNet* exibiram uma alta especificidade de 78,75% ($\pm 18,58\%$) e 75% ($\pm 16,77\%$), respectivamente. A *Deep ConvNet* apresentou a maior sensibilidade média (75%), porém com a menor acurácia e especificidade médias entre todas as arquiteturas nessa frequência.

Os resultados apontaram que as arquiteturas *Shallow ConvNet* tendem a apresentar melhor desempenho em ambas as frequências de amostragem, com resultados superiores na versão de 200Hz. Por outro lado, as arquiteturas *EEGNet* apresentaram os resultados mais variáveis e mais dependentes de ajustes adicionais para melhorar o desempenho.

No geral, os experimentos que utilizam os segmentos de trechos selecionados aleatoriamente adotaram uma faixa de valores menores para o hiperparâmetro *learning_rate*. Essa escolha foi motivada pela rápida convergência dos modelos gerados a partir dessa abordagem. Como consequência, para obter melhores resultados, uma maior quantidade de passos de treinamento é necessária, o que coloca a *Shallow ConvNet* com vantagem em relação às demais arquiteturas, devido à sua estrutura mais simples e à menor quantidade de parâmetros para ajustar requerida, o que é corroborado pelo maior desempenho apresentado em ambos os conjuntos de dados.

4.3 Discussão dos resultados experimentais

Com base na análise dos resultados das Tabelas 4 e 5 e considerando os resultados selecionados por especialistas, a arquitetura *Deep ConvNet-200Hz* obteve os resultados mais consistentes e superiores em termos de acurácia média, especificidade média e F1-Score médio. Essa arquitetura foi a mais estável e teve um desempenho geral melhor em comparação com outras arquiteturas. Por outro lado, considerando os resultados selecionados aleatoriamente, a arquitetura *Shallow ConvNet-200Hz* apresentou os melhores resultados em termos de acurácia média e F1-Score médio. Ela também demonstrou boa estabilidade em seus resultados. Os melhores modelos considerando as duas abordagens de seleção de trechos, por especialistas e aleatório, foram os analisados com a frequência de amostragem de 200Hz. Possivelmente pela maior quantidade de informações capturadas, maior resolução temporal e superior adaptação das arquiteturas ao domínio do problema em questão.

Quando consideramos todos os resultados, a arquitetura *Shallow ConvNet* em 200Hz, com seleção aleatória, apresentou os melhores resultados gerais. Isso ressalta o potencial dessas arquiteturas em encontrar padrões escondidos nos dados, que muitas vezes podem passar despercebidos pelos especialistas. No entanto, ao analisarmos todas as arquiteturas e todas as frequências de amostragem testadas por seleção aleatória, é

possível observar que a F1-Score permaneceu abaixo de 60% na maioria dos casos, com o pior resultado de média (33,62%) e desvio padrão ($\pm 30,81\%$) obtidos pela arquitetura *EEGNet* em 100Hz. Isso sugere que o papel dos especialistas na seleção de trechos pode ter contribuído para selecionar dados mais relevantes, com menor ruído, mais adequados e confiáveis. Corroborando-se, portanto, que o especialista pode contribuir significativamente para a qualidade e relevância dos dados utilizados na análise, o que por sua vez pode levar a resultados mais precisos, confiáveis e interpretáveis por parte dos modelos de aprendizado de máquina.

A aplicação de arquiteturas convolucionais combinadas com camadas densamente conectadas para o aprendizado de padrões nos sinais de EEG também foi explorada nos experimentos de Bissaro (2021), que para permitir o aprendizado de padrões temporais e espaciais, adicionou uma nova dimensão que modela o posicionamento dos eletrodos no escalpo do paciente, com seus relacionamentos modelados a partir da média entre eletrodos vizinhos. Para este trabalho, a modelagem temporal e sequencial é realizada através da aplicação da técnica FBCSP, que possibilita o aprendizado de padrões em diferentes faixas de banda significativas para uma melhor discriminação das classes do problema. Embora as simulações de Bissaro (2021) não tenham sido reproduzidas, o modelo *Shallow ConvNet* em 200Hz, apresentou resultados gerais superiores em comparação com a arquitetura CNN-Padrão com heurística de atenção.

Em geral, os modelos apresentaram desvio padrão próximo ou superior a 10% em todas as métricas de avaliação. Com o pior desvio padrão sendo observado na sensibilidade da arquitetura *EEGNet-100Hz*, com segmentos aleatórios e sensibilidade da *Deep ConvNet-100Hz* para registros selecionados por especialistas. Diversos motivos podem ser levantados para esse problema, incluindo a própria complexidade inerente aos dados de EEG e a presença de ruídos, que pode ter afetado alguns pacientes, conforme constatado por Bissaro (2021) em seu trabalho. Delorme (2023) e Coelli et al. (2024) exploraram a aplicação de *pipelines* automáticos para a preparação de sinais de traçado de EEG. Esse tipo de abordagem automatizada, sendo aplicada a todos os registros do conjunto de dados, poderá ser explorada em futuros trabalhos.

5 Conclusão

Neste trabalho foram desenvolvidos e avaliados modelos de aprendizado profundo para gerar prognósticos confiáveis para pacientes em coma utilizando dados de EEG. Foi realizada uma análise comparativa das diferentes frequências de amostragem para o tratamento dos dados dos exames. Foram analisadas as arquiteturas *EEGNet*, *Shallow ConvNet* e *Deep ConvNet* em diferentes frequências de amostragem. Com base nas arquiteturas selecionadas, foram desenvolvidos e avaliados diferentes modelos de aprendizado profundo. Isso incluiu a análise do desempenho dos modelos em relação a métricas de avaliação. Os modelos foram avaliados tanto em trechos selecionados por especialistas quanto em trechos selecionados aleatoriamente, fornecendo uma visão abrangente do seu desempenho.

As três arquiteturas apresentaram bom desempenho, principalmente quando submetidas à classificação de trechos de EEG selecionados por especialistas. Nesse contexto, a arquitetura *Shallow ConvNet* obteve destaque nos experimentos que consideraram a frequência de amostragem dos exames em 100Hz e 200Hz, enquanto a *Deep ConvNet* se destacou quando a frequência foi padronizada em 200Hz. Foi possível observar que a arquitetura com menor quantidade de parâmetros para ajuste apresentou maior facilidade para encontrar hiperparâmetros que possibilitam melhor aprendizado de padrões complexos do sinal de EEG. No geral, as arquiteturas treinadas com trechos selecionados por especialistas performaram melhor quando comparadas com a performance de modelos expostos a trechos segmentados aleatoriamente, o que permite concluir que os especialistas exerceram um papel crucial na seleção de séries temporais que apresentaram maior qualidade de dados e relevância para o prognóstico, apesar da estratégia de seleção aleatória demonstrar potencial nos experimentos que utilizaram a *Shallow ConvNet*.

Este trabalho utilizou técnicas para reamostragem das séries temporais do EEG, o que permitiu analisar a influência de diferentes configurações de frequência de amostragem para o traçado, aplicadas em modelos de aprendizado profundo com diferentes arranjos de camadas e profundidades variadas. Também foi apresentado uma nova abordagem para configuração de ferramentas de otimização de hiperparâmetros, baseada em arquivos de serialização de objetos YAML, que proporcionou praticidade, organização e legibilidade, características que se mostraram essenciais para cenários de otimização que apresentam uma grande variabilidade nos valores de hiperparâmetros.

Algumas limitações relevantes observadas neste trabalho merecem ser discutidas. Apesar da exploração de diversas configurações de frequência de amostragem, resultando em uma variação na dimensionalidade de dados, é importante notar que os segmentos de

exame de EEG analisados ainda são limitados pelo tempo de apenas 2 segundos, o que impediu a exploração de arquiteturas com configurações mais profundas e que poderiam se beneficiar da maior dimensionalidade dos dados. As arquiteturas escolhidas apresentam uma abordagem genérica simples, baseadas na aplicação de redes neurais convolucionais e camadas densamente conectadas, e apesar de contarem com suas implementações da técnica FBCSP para aprendizado de características temporais do sinal, não incorporam células recorrentes especializadas para essa finalidade, nem permitem a inserção de uma dimensão extra que modele a posição e relacionamento dos eletrodos no escalpo do paciente. Além disso, para padronizar as frequências de amostragem, recorreu-se a processos de *downsampling* e *upsampling*, este último utilizado para aumentar a frequência de amostragem dos sinais. Contudo, essa técnica envolve a inserção de pontos artificiais na série temporal sem necessariamente considerar as características e condições de captura originais do exame, o que pode, em último caso, prejudicar o aprendizado quando utilizada em frequências de amostragem superiores às exploradas neste trabalho.

As arquiteturas de rede selecionadas para este trabalho foram responsáveis por aprimorar resultados obtidos anteriormente para o prognóstico de pacientes em coma. A possibilidade de adaptar essas arquiteturas, para incluir uma dimensão que modele a disposição dos eletrodos no escalpo do paciente, é uma abordagem que pode ser explorada no futuro. Considerando os problemas de escassez e balanceamento dos dados enfrentados por este trabalho, a seleção totalmente aleatória de uma maior quantidade de trechos ou seleção aleatória combinada com trechos selecionados por especialistas podem ser explorados, além de outras estratégias para processamento temporal dos sinais e avaliação dos modelos desenvolvidos em conjuntos de dados públicos.

Referências

- AL-HUSSAINI, I.; MITCHELL, C. S. Seizft: Interpretable machine learning for seizure detection using wearables. **Bioengineering**, v. 10, n. 8, 2023. ISSN 2306-5354. Disponível em: <<https://www.mdpi.com/2306-5354/10/8/918>>. Citado 2 vezes nas páginas 24 e 37.
- ALJAAFARI, N. **Ichthyoplankton Classification Tool using Generative Adversarial Networks and Transfer Learning**. Dissertação (Mestrado), 2018. Citado na página 18.
- ANG, K. K.; CHIN, Z. Y.; ZHANG, H.; GUAN, C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. **2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)**, p. 2390–2397, 2008. Disponível em: <<https://doi.org/10.1109/IJCNN.2008.4634130>>. Citado 2 vezes nas páginas 19 e 20.
- BERRAR, D. Cross-validation. In: RANGANATHAN, S.; GRIBSKOV, M.; NAKAI, K.; SCHÖNBACH, C. (Ed.). **Encyclopedia of Bioinformatics and Computational Biology**. Oxford: Academic Press, 2019. p. 542–545. ISBN 978-0-12-811432-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B978012809633820349X>>. Citado na página 37.
- BHARDWAJ, A. **What is a Perceptron? – Basics of Neural Networks**. 2020. Site Towards Data Science. Disponível em: <<https://towardsdatascience.com/what-is-a-perceptron-basics-of-neural-networks-c4cfea20c590>>. Acesso em: 2024-04-05. Citado na página 15.
- BISSARO, L. Z. **Aprendizado de padrões EEG para prognóstico precoce de pacientes em coma usando redes echo state e redes neurais convolucionais**. 36-37 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, 2021. Citado 6 vezes nas páginas 12, 16, 18, 24, 29 e 46.
- BISSARO, L. Z.; JUNIOR, O. O. N.; FILHO, J. B. D.; JIN, Y.; CARNEIRO, M. G. Towards the prognosis of patients in coma using echo state networks for eeg analysis. **Procedia Computer Science**, v. 222, p. 509–518, 2023. ISSN 1877-0509. International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050923009547>>. Citado na página 24.
- BLANKERTZ, B.; TOMIOKA, R.; LEMM, S.; KAWANABE, M.; MULLER, K.-r. Optimizing spatial filters for robust eeg single-trial analysis. **IEEE Signal Processing Magazine**, v. 25, n. 1, p. 41–56, 2008. Citado 2 vezes nas páginas 19 e 20.
- BORRA, D.; FANTOZZI, S.; MAGOSSO, E. Interpretable and lightweight convolutional neural network for eeg decoding: Application to movement execution and imagination. **Neural Networks**, v. 129, p. 55–74, 2020. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0893608020302021>>. Citado na página 18.

- CARIELLO, S.; SANALITRO, D.; MICALI, A.; BUSCARINO, A.; BUCOLO, M. Brain-computer-interface-based smart-home interface by leveraging motor imagery signals. **Inventions**, v. 8, p. 91, 07 2023. Citado na página 18.
- CARNEIRO, M. G.; RAMOS, C. D.; DESTRO-FILHO, J.-B.; ZHU, Y. tao; JI, D.; ZHAO, L. High-level classification for EEG analysis. **2023 International Joint Conference on Neural Networks (IJCNN)**, p. 1–8, 2023. Disponível em: <<https://doi.org/10.1109/IJCNN54540.2023.10191823>>. Citado na página 25.
- COELLI, S.; CALCAGNO, A.; CASSANI, C. M.; TEMPORITI, F.; REALI, P.; GATTI, R.; GALLI, M.; BIANCHI, A. M. Selecting methods for a modular eeg pre-processing pipeline: An objective comparison. **Biomedical Signal Processing and Control**, v. 90, p. 105830, 2024. ISSN 1746-8094. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1746809423012636>>. Citado 2 vezes nas páginas 31 e 46.
- COSTA, P. G. d. **Base de dados em eletroencefalografia (EEG)**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, 2022. Citado na página 28.
- DELORME, A. Eeg is better left alone. **Scientific Reports**, v. 13, n. 1, p. 2372, 02 2023. ISSN 2045-2322. Disponível em: <<https://doi.org/10.1038/s41598-023-27528-0>>. Citado 2 vezes nas páginas 31 e 46.
- EDLOW, B. L.; CLAASSEN, J.; SCHIFF, N. D.; GREER, D. M. Recovery from disorders of consciousness: mechanisms, prognosis and emerging therapies. **Nature Reviews Neurology**, v. 17, n. 3, p. 135–156, 03 2021. ISSN 1759-4766. Disponível em: <<https://doi.org/10.1038/s41582-020-00428-x>>. Citado na página 10.
- FACELI, K.; LORENA, A. C.; GAMA, J.; ALMEIDA, T. A. d.; CARVALHO, A. C. P. d. L. F. d. **Inteligência artificial: uma abordagem de aprendizado de máquina**. LTC, 2021. ISBN 9788521637349. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788521637509/>>. Citado 2 vezes nas páginas 13 e 14.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Disponível em: <<https://books.google.com.br/books?id=Np9SDQAAQBAJ>>. Citado 2 vezes nas páginas 17 e 18.
- IBM. **What is a neural network?** 2024. Site da IBM. Disponível em: <<https://www.ibm.com/topics/neural-networks>>. Acesso em: 2024-04-05. Citado na página 15.
- JUNIOR, E. **Principais métricas de avaliação de modelos em Machine Learning**. 2023. Disponível em: <<https://medium.com/@ejunior029/principais-m%C3%A9tricas-de-classifica%C3%A7%C3%A3o-de-modelos-em-machine-learning-94eeb4b40ea9>>. Acesso em: 2024-04-06. Citado na página 39.
- KANAGAMANI, T.; KRISHNAMURTHY, R.; CHAKRAVARTHY, S.; RAVINDRAN, B.; MENON, R. Memory consolidation with orthogonal gradients for avoiding catastrophic forgetting. 01 2023. Citado na página 15.

- KANDA, R.; CURY, I.; LOVATEL, A.; KANDA, P. Sugestões práticas para a descrição do EEG de rotina. **Revista Neurociências**, v. 22, p. 464–468, 09 2014. Citado na página 10.
- LAWHERN, V.; SOLON, A.; WAYTOWICH, N.; GORDON, S.; HUNG, C.; LANCE, B. EEGNet: A compact convolutional network for EEG-based brain-computer interfaces. **Journal of Neural Engineering**, v. 15, 11 2016. Citado 2 vezes nas páginas 22 e 23.
- LIU, Q.; WU, Y. Supervised learning. In: **Encyclopedia of the Sciences of Learning**. Boston, MA: Springer US, 2012. p. 3243–3245. ISBN 978-1-4419-1428-6. Disponível em: <https://doi.org/10.1007/978-1-4419-1428-6_451>. Citado na página 14.
- MAESTROVIRTUALE. **Eletroencefalograma (EEG): o que é e como é usado?** 2024. Website. Disponível em: <https://maestrovirtuale.com/eletroencefalograma-eeg-o-que-e-e-como-e-usado/?expand_article=1>. Acesso em: 2024-04-07. Citado na página 13.
- MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <<https://books.google.com.br/books?id=EoYBngEACAAJ>>. Citado na página 13.
- MULERT, C.; LEMIEUX, L. **EEG - fMRI: Physiological Basis, Technique, and Applications**. Springer Berlin Heidelberg, 2009. ISBN 9783540879190. Disponível em: <<https://books.google.com.br/books?id=1UOiD94HdXkC>>. Citado na página 9.
- NIELSEN, M. A. **Neural Networks and Deep Learning**. Determination Press, 2015. Disponível em: <<https://neuralnetworksanddeeplearning.com/>>. Citado na página 15.
- NOGARE, D. **Performance de Machine Learning – Matriz de Confusão**. 2020. Website. Disponível em: <<https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/>>. Acesso em: 2024-04-06. Citado na página 39.
- PAIVA, M. R. d.; JR, M. G.; RUFIM, E. S.; CAMPOS, M.; FILHO, J. B. D. Avaliação visual dos padrões eletroencefalográficos de pacientes clinicamente em coma / visual evaluation of electroencephalographic standards of patients clinically in coma. **Brazilian Journal of Health Review**, v. 1, n. 2, p. 447–455, Nov. 2018. Disponível em: <<https://ojs.brazilianjournals.com.br/ojs/index.php/BJHR/article/view/797>>. Citado na página 27.
- PAIXÃO, G. M. d. M.; SANTOS, B. C.; ARAUJO, R. M. d.; RIBEIRO, M. H.; MORAES, J. L. d.; RIBEIRO, A. L. Machine learning na medicina: Revisão e aplicabilidade. **Arquivos Brasileiros de Cardiologia**, Sociedade Brasileira de Cardiologia - SBC, v. 118, n. 1, p. 95–102, 01 2022. ISSN 0066-782X. Disponível em: <<https://doi.org/10.36660/abc.20200596>>. Citado na página 9.
- PUGGINA, A. C. G.; SILVA, M. J. P. d. Sinais vitais e expressão facial de pacientes em estado de coma. **Revista Brasileira de Enfermagem**, v. 62, n. 3, p. 435–441, 05 2009. ISSN 0034-7167. Disponível em: <<https://doi.org/10.1590/S0034-71672009000300016>>. Citado na página 9.

RAMOS, C. D. **Aprendizado de máquina como ferramenta para o prognóstico de pacientes em coma usando sinais eletroencefalográficos no espectro de 1 a 100 Hz**. 23 p. Tese (Doutorado), 01 2022. Citado 4 vezes nas páginas 9, 12, 13 e 24.

ROJAS, M. G.; OLIVERA, A. C.; VIDAL, P. J. Optimising multilayer perceptron weights and biases through a cellular genetic algorithm for medical data classification. **Array**, v. 14, p. 100173, 2022. ISSN 2590-0056. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2590005622000339>>. Citado na página 16.

SCHIRRMEISTER, R. T.; SPRINGENBERG, J. T.; FIEDERER, L. D. J.; GLASSTETTER, M.; EGGENSBERGER, K.; TANGERMANN, M.; HUTTER, F.; BURGARD, W.; BALL, T. Deep learning with convolutional neural networks for eeg decoding and visualization. **Human Brain Mapping**, v. 38, n. 11, p. 5391–5420, 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730>>. Citado 4 vezes nas páginas 21, 22, 23 e 25.

SCHWABEDAL, J. T. C.; SNYDER, J. C.; CAKMAK, A. S.; NEMATI, S.; CLIFFORD, G. D. Addressing class imbalance in classification problems of noisy signals by using fourier transform surrogates. **arXiv: Signal Processing**, 2018. Disponível em: <<https://api.semanticscholar.org/CorpusID:49398366>>. Citado 2 vezes nas páginas 24 e 37.

SHAFKAT, I. **Intuitively Understanding Convolutions for Deep Learning**. 2018. Site Towards Data Science. Disponível em: <<https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>>. Acesso em: 2024-04-05. Citado na página 17.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, v. 415, p. 295–316, 2020. ISSN 0925-2312. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231220311693>>. Citado na página 32.

ZUBLER, F.; TZOVARA, A. Deep learning for EEG-based prognostication after cardiac arrest: from current research to future clinical applications. **Front Neurol**, Switzerland, v. 14, p. 1183810, jul. 2023. Citado na página 25.