



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE EM ENGENHARIA ELÉTRICA**

**JOÃO LUCAS SANTOS AZEVEDO**

**ANÁLISE DE CLASSIFICADORES MULTICLASSE APLICADOS A  
SINAIS DE EEG BASEADOS EM SSVEP**

Tucuruí  
2025



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE EM ENGENHARIA ELÉTRICA**

**JOÃO LUCAS SANTOS AZEVEDO**

**ANÁLISE DE CLASSIFICADORES MULTICLASSE APLICADOS A  
SINAIS DE EEG BASEADOS EM SSVEP**

Trabalho de Conclusão de Curso apresentado ao Programa de Graduação em Engenharia Elétrica da Universidade Federal do Pará, como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Área de Concentração: Processamento Digital de Sinais

Orientador: Dr. Cleison Daniel Silva

Tucuruí  
2025

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD  
Sistema de Bibliotecas da Universidade Federal do Pará  
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

---

S237a Santos Azevedo, João Lucas.  
ANÁLISE DE CLASSIFICADORES MULTICLASSE  
APLICADOS A SINAIS DE EEG BASEADOS EM SSVEP / João  
Lucas Santos Azevedo. — 2025.  
55 f. : il. color.

Orientador(a): Prof. Dr. Cleison Daniel Silva  
Trabalho de Conclusão (Graduação) - Universidade Federal do  
Pará, Campus Universitário de Tucuruí, Faculdade de Engenharia  
Elétrica, Tucuruí, 2025.

1. Interface Cérebro Máquina. 2. Steady State Visual  
Evoked Potential. 3. Machine Learning. I. Título.

CDD 621.3822

---



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE DE ENGENHARIA ELÉTRICA**

**TÍTULO: ANÁLISE DE CLASSIFICADORES MULTICLASSE  
APLICADOS A SINAIS DE EEG BASEADOS EM SSVEP**

**DISCENTE:** João Lucas Santos Azevedo

**MATRÍCULA:** 202033940032

#	BANCA EXAMINADORA	CONDIÇÃO
1	<i>Prof. Dr. Cleison Daniel Silva – NDAE/UFPA</i>	<i>Orientador</i>
2	<i>Prof. Dr. Raphael Barros Teixeira – FEE/UFPA</i>	<i>Membro Interno</i>
3	<i>Prof. Me. Jhoanyn Fantin – PMJ - SC</i>	<i>Membro externo</i>

**Data da Defesa:** 13/08/2025 | **Hora Início:** 10:00 | **Hora Término:** 11:30

Trabalho Escrito (0 a 10 pontos por critério)	Examinador 1	Examinador 2	Examinador 3
Formatação	9,5	-	10
Linguagem (gramática e semântica)	9,5	-	9
Conteúdo técnico	9	-	9

Defesa Oral (0 a 10 pontos por critério)	Examinador 1	Examinador 2	Examinador 3
Sequência lógica de apresentação	10	-	10
Administração do tempo	9,5	-	9
Expressão oral	9,5	-	9,5
Domínio do tema	9	-	9,5

<b>Média por examinador</b>	9,42	-	9,42
<b>Média Final</b>	9,42		
<b>Conceito Final</b>	<b>EXCELENTE</b>		

Tucuruí-PA, 13 de agosto de 2025.

Certifico que a defesa de TCC realizou-se com a participação a distância do(s) membros(s) Prof. Dr. Raphael Barros Teixeira, Prof. Me. Jhoanyn Fantin, e que, depois das arguições e deliberações realizadas, cada participante a distância afirmou estar de acordo com o conteúdo do parecer da banca examinadora, redigido nesta ata.

Documento assinado digitalmente



CLEISON DANIEL SILVA

Data: 14/08/2025 18:04:42-0300

Verifique em <https://validar.iti.gov.br>

Cleison Daniel Silva  
Presidente da sessão

## LISTA DE ILUSTRAÇÕES

Figura 1 – Divisão dos lobos cerebrais. Fonte: (JORGE, 2022) . . . . .	14
Figura 2 – Mapeamento do couro cabeludo com a localização e nomenclatura dos eletrodos de acordo com o sistema 10-10. Fonte: (DIAS, 2023) . . . . .	15
Figura 3 – Resposta SSVEP a estímulos codificados por frequência na região occipital do cérebro. Fonte: (DEMIR; ARSLAN; UYSAL, 2019) . . . . .	16
Figura 4 – Exemplo visual do funcionamento do LDA. Fonte: GEEKSFORGEEKS (2020). . . . .	23
Figura 5 – Exemplo visual do funcionamento do KNN. Fonte: SCIKIT-LEARN (2024). . . . .	24
Figura 6 – Fluxograma simplificado do funcionamento do <i>Random Forest</i> . Fonte: DIDA (2024). . . . .	25
Figura 7 – Exemplo de Matriz de Confusão. Fonte: (AMBIENTAL, 2020). . . . .	27
Figura 8 – Diagrama Simplificado do Processo Classificação. Fonte: O autor. . . . .	29
Figura 9 – Exemplo do experimento e da duração. Fonte:(DIAS, 2023). . . . .	30
Figura 10 – Resposta em Frequência dos Filtros. Fonte: O autor. . . . .	31
Figura 11 – Exemplo de extração de épocas do sinal. Fonte: O autor. . . . .	32
Figura 12 – Gráfico Periodogramas sinal 8Hz filtrado em 8Hz. Fonte: O autor. . . . .	33
Figura 13 – <i>Features</i> 8Hz, 14Hz e 28Hz. Fonte: O autor. . . . .	36
Figura 14 – <i>Features</i> 8Hz, 14Hz e 28Hz após Tratamento 1. Fonte: O autor. . . . .	37
Figura 15 – <i>Features</i> 8Hz, 14Hz e 28Hz após Tratamento 2. Fonte: O autor. . . . .	38
Figura 16 – <i>Features</i> 8Hz, 14Hz e 28Hz após Tratamento 3. Fonte: O autor. . . . .	39
Figura 17 – <i>Features</i> 8Hz, 14Hz e 28Hz após Tratamento 4. Fonte: O autor. . . . .	40

## LISTA DE TABELAS

Tabela 1 – Principais bibliotecas utilizadas . . . . .	30
Tabela 2 – Hiperparâmetros da Função de OB . . . . .	34
Tabela 3 – Espaços de busca para os hiperparâmetros ajustáveis . . . . .	35
Tabela 4 – Métricas de Classificação sem Tratamento . . . . .	41
Tabela 5 – Hiperparâmetros otimizados sem Tratamento . . . . .	41
Tabela 6 – Métricas de Classificação com Tratamento 1 . . . . .	42
Tabela 7 – Hiperparâmetros otimizados com Tratamento 1 . . . . .	42
Tabela 8 – Métricas de Classificação com Tratamento 2 . . . . .	43
Tabela 9 – Hiperparâmetros otimizados com Tratamento 2 . . . . .	43
Tabela 10 – Métricas de Classificação com Tratamento 3 . . . . .	44
Tabela 11 – Hiperparâmetros otimizados com Tratamento 3 . . . . .	44
Tabela 12 – Métricas de Classificação com Tratamento 4 . . . . .	45
Tabela 13 – Hiperparâmetros otimizados com Tratamento 4 . . . . .	45
Tabela 14 – Tabela Comparativo Geral Classificadores . . . . .	46

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos</b>	<b>12</b>
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
<b>2.1</b>	<b>Introdução à Anatomia Cerebral</b>	<b>14</b>
<b>2.2</b>	<b>Eletroencefalografia (EEG)</b>	<b>15</b>
<b>2.3</b>	<b>Potencial Evocado Visual em Estado Estacionário</b>	<b>16</b>
<b>2.4</b>	<b>Pré-processamento de Sinais de EEG</b>	<b>17</b>
2.4.1	Filtro Passa-Faixa	17
2.4.2	Extração de Épocas	18
<b>2.5</b>	<b>Extração de <i>Features</i></b>	<b>18</b>
2.5.1	Periodograma	19
<b>2.6</b>	<b>Tratamento das <i>Features</i></b>	<b>19</b>
2.6.1	Tratamento 1: Normalização de <i>features</i>	20
2.6.2	Tratamento 2: Normalização <i>Z-score</i> Global	20
2.6.3	Tratamento 3: Normalização <i>Z-score</i> de <i>features</i>	21
2.6.4	Tratamento 4: Normalização <i>Z-score</i> de amostras	21
<b>2.7</b>	<b>Algoritmos de Classificação</b>	<b>22</b>
2.7.1	Análise Discriminante Linear (LDA)	22
2.7.2	KNN	24
2.7.3	<i>Random Forest</i>	25
<b>2.8</b>	<b>Otimização Bayesiana</b>	<b>26</b>
<b>2.9</b>	<b>Métricas de Avaliação</b>	<b>27</b>
2.9.1	Acurácia	27
2.9.2	Precisão	28
<b>3</b>	<b>METODOLOGIA</b>	<b>29</b>
<b>3.1</b>	<b>Descrição da Base de dados</b>	<b>29</b>
<b>3.2</b>	<b>Ambiente Computacional e Tecnologias Utilizadas</b>	<b>29</b>
<b>3.3</b>	<b>Pré-processamento dos dados</b>	<b>31</b>
3.3.1	Remoção do sinal DC, filtragem e seleção do canal de interesse	31
3.3.2	Aplicação da Extração de Épocas	32
3.3.3	Aplicação do Periodograma e Extração de <i>Features</i>	32
<b>3.4</b>	<b>Aplicação das Técnicas de Tratamento</b>	<b>33</b>
<b>3.5</b>	<b>Classificação dos Sinais de EEG.</b>	<b>33</b>
<b>3.6</b>	<b>Otimização de Hiperparâmetros</b>	<b>34</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>36</b>

<b>4.1</b>	<b>Resultados do Tratamento dos dados</b>	<b>36</b>
4.1.1	<i>Features</i> Geradas sem Tratamento	36
4.1.2	<i>Features</i> Geradas com Tratamento 1	37
4.1.3	<i>Features</i> Geradas com Tratamento 2	38
4.1.4	<i>Features</i> Geradas com Tratamento 3	39
4.1.5	<i>Features</i> Geradas com Tratamento 4	40
<b>4.2</b>	<b>Desempenho de Classificação sem Tratamento</b>	<b>41</b>
<b>4.3</b>	<b>Desempenho de Classificação com Tratamento 1</b>	<b>42</b>
<b>4.4</b>	<b>Desempenho de Classificação com Tratamento 2</b>	<b>43</b>
<b>4.5</b>	<b>Desempenho de Classificação com Tratamento 3</b>	<b>44</b>
<b>4.6</b>	<b>Desempenho de Classificação com Tratamento 4</b>	<b>45</b>
<b>4.7</b>	<b>Comparação dos desempenhos</b>	<b>46</b>
<b>5</b>	<b>CONCLUSÕES</b>	<b>47</b>
<b>6</b>	<b>SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>49</b>
	<b>REFERÊNCIAS</b>	<b>50</b>

**ANEXOS** **53**

**ANEXO A – LINK DE ACESSO AO REPOSITÓRIO DO GITHUB** . . **54**

Dedico este trabalho postumamente à minha mãe, a qual esteve ao meu lado durante toda a minha trajetória acadêmica e pessoal. Ainda que não esteja mais fisicamente ao nosso lado, viverá eternamente no coração e na memória de todos que tiveram o privilégio de a conhecer.

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente a Deus por tudo que conquistei em minha vida, e sem o qual nada teria alcançado ao longo desses anos de trajetória acadêmica e pessoal.

Aos meus familiares que sempre deram suporte, apoio, incentivo, encorajamento e orientação em todos os momentos, e sem os quais tudo teria sido mais difícil, agradeço de todo o meu coração. Em especial à minha tia por todo o amor, carinho, cuidado, preocupação e apoio dados durante os mais de dez anos que tenho vivido ao lado dela e à minha namorada por estar sempre me encorajando, motivando, fortalecendo e auxiliando nos momentos mais difíceis que passei em minha vida.

Aos meus amigos e colegas de turma, os quais foram peça fundamental da minha formação acadêmica, sempre compartilhando conhecimentos e ajudando de bom grado uns aos outros. Em especial ao meu amigo Gabriel Lima Ferreira, o qual além de amigo tornou-se como um irmão para mim, sendo uma das pessoas mais inteligentes que eu conheço e certamente um jovem com um futuro brilhante pela frente.

Ao meu orientador Cleison Daniel Silva por todo o auxílio e a excelência no ensino, tanto em sala de aula quanto no desenvolvimento deste trabalho. Um excelente profissional e um exemplo para todos os discentes que tiveram o privilégio de serem seus alunos.

"Mas buscai primeiro o Reino de Deus, e a sua justiça, e todas essas coisas vos serão acrescentadas"  
(Mateus 6:33).

## RESUMO

Interfaces Cérebro-Máquina (ICM) têm cada vez mais tomado uma posição de destaque dentro do meio acadêmico, sendo sua otimização e refinamento um dos objetos de maior interesse no campo da engenharia e neurociência, devido ao seu potencial de promover autonomia e melhorar a qualidade de vida, especialmente de pessoas com limitações motoras. Entre as abordagens existentes, as ICMs baseadas em Steady-State Visually Evoked Potentials (SSVEPs) se destacam pela praticidade e simplicidade na conversão de estímulos visuais em comandos. Os sistemas ICM baseados em SSVEP usam respostas cerebrais à estímulos visuais calibrados em frequências específicas, os quais são interpretados como diferentes comandos, a fim de serem utilizados em um dispositivo externo ou aplicação digital. Neste contexto, o presente trabalho realiza uma análise comparativa de classificadores multiclasse aplicados a sinais de EEG obtidos com o paradigma SSVEP, utilizando dados públicos do RIKEN-LABSP. Foram implementadas diferentes estratégias de extração de características e pré-processamento antes da aplicação dos classificadores *Linear Discriminant Analysis* (LDA), *K Nearest Neighbors* (KNN) e *Random Forest*. Os resultados demonstram a viabilidade da abordagem, com acurácias máximas de 100%, 83,33% e 74,81% para janelas temporais de 2s, 1 s e 500 ms, respectivamente. O classificador LDA apresentou o melhor desempenho entre os avaliados, sendo o mais adequado para a tarefa proposta.

**Palavras-chave:** Eletroencefalografia, SSVEP, Classificação Multiclasse, Otimização, Aprendizado de Máquina.

## ABSTRACT

Brain-Computer Interfaces (BCIs) have been gaining increasing prominence in academic research, with their optimization and refinement becoming a major focus in the fields of engineering and neuroscience. This is largely due to their potential to promote autonomy and improve quality of life, especially for individuals with motor impairments. Among the various approaches, BCIs based on Steady-State Visually Evoked Potentials (SSVEPs) stand out for their practicality and simplicity in converting visual stimuli into commands. SSVEP-based BCI systems rely on brain responses to visual stimuli calibrated at specific frequencies, which are interpreted as distinct commands for use in external devices or digital applications. In this context, the present study conducts a comparative analysis of multiclass classifiers applied to EEG signals acquired through the SSVEP paradigm, using public data from RIKEN-LABSP. Various feature extraction and preprocessing strategies were implemented prior to applying the Linear Discriminant Analysis (LDA), K Nearest Neighbors (KNN) and Random Forest classifiers. The results demonstrate the feasibility of the approach, with maximum accuracies of 100%, 83.33%, and 74.81% for time windows of 2 s, 1 s, and 500 ms, respectively. The LDA classifier achieved the best performance among those evaluated, proving to be the most suitable for the proposed task.

**Keywords:** Electroencephalography, SSVEP, Multiclass Classification, Optimization, Machine Learning.

# 1 INTRODUÇÃO

Nos últimos anos, tem havido um interesse crescente no desenvolvimento de interfaces cérebro-máquina (ICM) baseadas em sinais de eletroencefalografia (EEG), motivando inúmeros estudos voltados à avaliação e otimização de algoritmos de classificação (LOTTE et al., 2007). Esses estudos visam desenvolver abordagens que melhorem a acurácia e a robustez da classificação de sinais cerebrais, apesar dos desafios inerentes à variabilidade inter e intra-indivíduos (VILAS-BOAS; SILVA, 2020).

As ICMs são algoritmos capazes de traduzir a intenção produzida pelo cérebro de um indivíduo em um sinal de entrada para um dispositivo ou software, permitindo assim que o mesmo seja controlado pelos sinais cerebrais do usuário. Essa tradução das intenções do indivíduo é feita através da captação dos potenciais elétricos gerados pela atividade neuronal do cérebro através de um exame chamado eletroencefalografia (EEG) (TAN; NIJHOLT, 2010).

A eletroencefalografia é um exame não invasivo, realizado por meio de eletrodos posicionados sobre o couro cabeludo, os quais captam os sinais gerados pela atividade bioelétrica do cérebro. Esses sinais são gerados pela interação eletroquímica das sinapses, o que gera sinais de baixíssima amplitude contendo as respostas do cérebro a diferentes ações do indivíduo. A interpretação desses sinais permite inferir estados mentais ou intenções do usuário, a depender do paradigma experimental adotado (VILAS-BOAS; SILVA, 2020).

Dentre os paradigmas utilizados em sistemas ICM, destaca-se o uso do fenômeno denominado Potencial Evocado Visual em Estado Estacionário (SSVEP, do inglês *Steady-State Visually Evoked Potential*) pela sua praticidade, facilidade de aplicação e simplicidade de interpretação dos dados. O SSVEP consiste em uma resposta oscilatória do córtex visual a estímulos visuais periódicos, geralmente de frequência constante, manifestando-se no EEG na mesma frequência do estímulo visual observado pelo indivíduo, bem como em seus harmônicos (GARCIA-MOLINA; ZHU, 2011).

No desenvolvimento de ICMs, esses dados são coletados, pré-processados, tratados e, posteriormente, separados em classes a fim de transformá-los em informações úteis para a aplicação de interesse. A etapa de separação dos dados em classes é realizada pelos chamados algoritmos de classificação.

Algoritmos de classificação são algoritmos que buscam estimar de forma automática a classe de um dado representado por um vetor de características (DUDA; HART; STORK, 2000). Ao identificar padrões complexos nos dados, esses algoritmos permitem a categorização automática de informações, o que viabiliza soluções inovadoras em áreas como medicina, indústria, defesa e aplicações civis.

No entanto, o ajuste dos algoritmos de classificação representa um desafio significativo, uma vez que alguns de seus parâmetros, os chamados hiperparâmetros, precisam ser definidos

manualmente antes da execução. Esses hiperparâmetros exercem influência direta tanto no desempenho quanto no tempo de processamento do modelo. Encontrar a configuração ideal desses valores é uma das etapas mais críticas no desenvolvimento de sistemas ICM. Nesse contexto, destaca-se a Otimização Bayesiana (OB), uma técnica eficaz na busca automática por combinações de hiperparâmetros que maximizem o desempenho do classificador.

A OB é uma abordagem iterativa utilizada para otimizar funções objetivo que possuam um alto custo computacional para serem avaliadas, especialmente quando não se conhece uma formulação explícita da função, mas é possível observá-la pontualmente com ruído (BROCHU; CORA; FREITAS, 2010). Em contextos como o presente estudo, ela permite selecionar combinações de parâmetros que maximizem o desempenho do classificador com base em um histórico de tentativas anteriores.

Assim, este trabalho pretende avaliar o desempenho de diferentes classificadores multi-classe aplicados a um banco de dados público de EEG, disponibilizado pelo RIKEN-LABSP em colaboração com o Dr. Hovagim Bakardjian. O conjunto de dados foi obtido por meio de experimentos realizados com cinco participantes, submetidos a estímulos visuais nas frequências de 8 Hz, 14 Hz e 28 Hz, com amostragem de 250 Hz.

Além disso, é analisado o impacto de diferentes técnicas de tratamento dos sinais, como a separação do sinal em diferentes épocas (125 ms, 250 ms, 500 ms, 1 s e 2 s) e o tratamento dos dados, avaliando técnicas como normalização e a normalização *Z-score*.

Os tópicos abordados estão organizados da seguinte forma: a Seção 2 apresenta a fundamentação teórica necessária; a Seção 3 descreve a metodologia aplicada; a Seção 4 discute os resultados experimentais; e a Seção 5 apresenta as conclusões.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Avaliar o desempenho de múltiplos classificadores multiclasse aplicados à análise de sinais de EEG baseados no paradigma SSVEP, investigando sua capacidade de distinguir diferentes frequências de estímulo com base nas respostas cerebrais, considerando distintos métodos de tratamento dos dados e variações nas janelas temporais de análise.

### 1.1.2 Objetivos Específicos

- Investigar a aplicação dos classificadores multiclasse: *Linear Discriminant Analysis* (LDA), *K Nearest Neighbors* (KNN) e *Random Forest* na análise de sinais de EEG com estímulo visual do tipo SSVEP.

- 
- Avaliar o impacto de diferentes tratamentos de dados e janelas temporais na qualidade da classificação.
  - Otimizar os hiperparâmetros dos modelos por meio de técnicas de Otimização Bayesiana.
  - Comparar o desempenho dos classificadores com base nas métricas de acurácia e precisão.

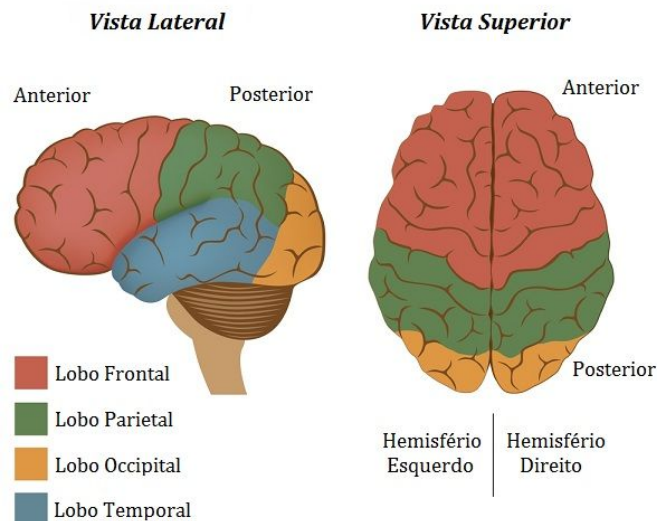
## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos fundamentais para a adequada compreensão do trabalho desenvolvido.

### 2.1 Introdução à Anatomia Cerebral

O cérebro é um dos órgãos mais complexos do corpo humano, formado por aproximadamente cem bilhões de neurônios e uma quantidade ainda maior de células gliais (MARTIN, 2013). Essas células se organizam em redes e estruturas especializadas, responsáveis por funções essenciais à vida, como controle motor, percepção sensorial, linguagem, cognição, regulação hormonal e manutenção da homeostase. Além de coordenar os sentidos e a movimentação voluntária, o cérebro também desempenha papel central no processamento de emoções, aprendizado e memória, sendo o principal integrador das funções fisiológicas do organismo humano (SILVERTHORN, 2017; KANDEL et al., 2014).

A estrutura do cérebro é dividida em dois hemisférios: esquerdo e direito, interligados por uma densa faixa de fibras nervosas denominada corpo caloso (ROCHA, 2019). Cada hemisfério, por sua vez, é composto por quatro lobos com funções distintas: frontal, parietal, occipital e temporal (SILVERTHORN, 2017). Essas divisões anatômicas estão ilustradas na Figura 1.



**Figura 1 – Divisão dos lobos cerebrais. Fonte: (JORGE, 2022)**

O lobo occipital, localizado na região posterior do cérebro, é o menor entre os quatro lobos cerebrais e exerce um papel fundamental no processamento da informação visual, sendo responsável pela percepção de atributos como cor, forma e movimento. A análise da atividade dessa região permite inferir o estado visual do indivíduo, possibilitando, por exemplo, identificar se ele está com os olhos abertos ou fechados, concentrado ou relaxado (BUENO, 2019).

Essa atividade elétrica cerebral pode ser monitorada, a fim de extrair informações úteis sobre a resposta cerebral à estímulos visuais recebidos por um indivíduo. A obtenção dessas respostas é feita por meio de um exame chamado eletroencefalografia, comumente abreviado como EEG.

## 2.2 Eletroencefalografia (EEG)

A eletroencefalografia (EEG) é uma técnica de captação dos potenciais elétricos gerados pela atividade neuronal no cérebro. Trata-se de um exame não invasivo, realizado por meio de eletrodos posicionados sobre o couro cabeludo (TAN; NIJHOLT, 2010). Esses eletrodos captam os sinais gerados pela atividade bioelétrica do cérebro. Os sinais captados pelo EEG apresentam baixíssima amplitude, geralmente na ordem de microvolts, e refletem o estado funcional do cérebro (LOPES, 2019).

Devido à sua natureza sensível, os sinais de EEG são altamente suscetíveis a interferências externas e artefatos fisiológicos, como movimentos oculares, musculares e atividade cardíaca. Apesar dessas limitações, o EEG destaca-se por sua alta resolução temporal, portabilidade e custo relativamente baixo em comparação a outras técnicas, como fMRI e MEG (NIEDERMEYER; SILVA, 2005).

Outra vantagem do uso de EEG é sua natureza não invasiva, dispensando a necessidade de intervenções cirúrgicas e implantação de sensores diretamente no cérebro do indivíduo cujos sinais são adquiridos. Para a captação desses sinais cerebrais, os eletrodos são posicionados conforme um sistema padronizado internacional, garantindo cobertura adequada dos hemisférios cerebrais e permitindo a aquisição de sinais com boa representatividade funcional. O modelo de captação de sinais do modelo 10-10 é apresentado na Figura 2.

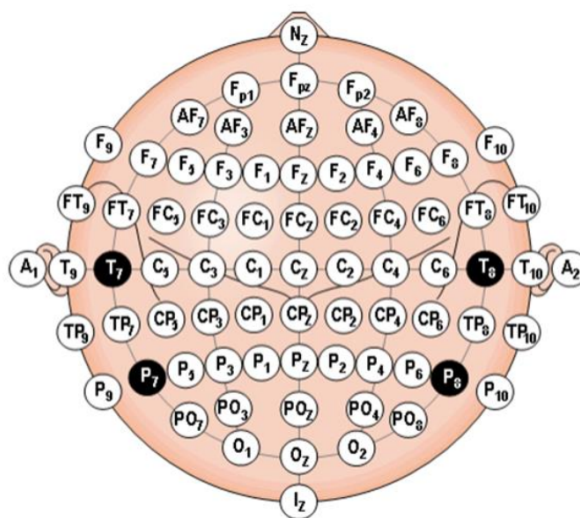


Figura 2 – Mapeamento do couro cabeludo com a localização e nomenclatura dos eletrodos de acordo com o sistema 10-10. Fonte: (DIAS, 2023)

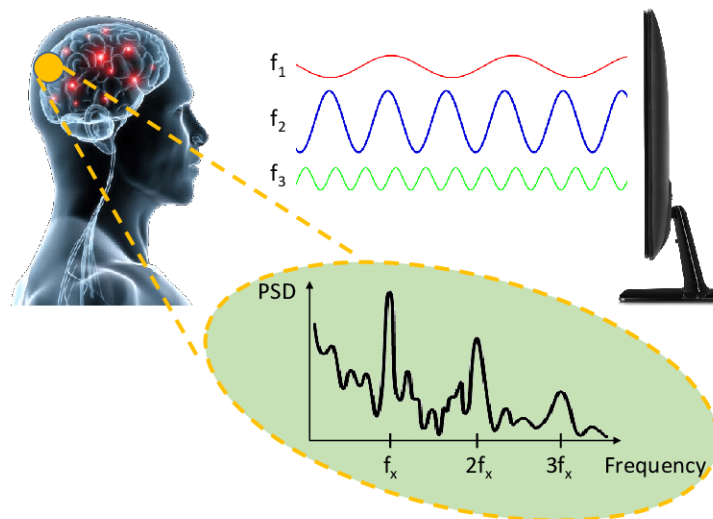
A Figura 2 apresenta o mapeamento aplicado a cada região do cérebro, a fim de triangular a posição onde cada resposta é captada. Esta técnica permite isolar sinais emitidos por regiões de interesse para cada estudo individual. Neste estudo, serão de maior relevância os eletrodos com nomenclatura iniciada na letra "O", pois indicam a região occipital, onde são captadas as respostas visuais, em especial, os sinais de SSVEP.

## 2.3 Potencial Evocado Visual em Estado Estacionário

O Potencial Evocado Visual em Estado Estacionário (SSVEP, do inglês *Steady-State Visually Evoked Potential*) é uma resposta oscilatória gerada pelo cérebro em reação a estímulos visuais periódicos de frequência constante (DIAS, 2023). Quando um indivíduo direciona sua atenção a um estímulo visual intermitente, como um LED ou imagem piscando em determinada frequência, ocorre uma sincronização da atividade elétrica no córtex visual com a frequência do estímulo, produzindo padrões facilmente detectáveis por meio do EEG.

Essas respostas são mais pronunciadas na região occipital do cérebro e manifestam-se no espectro de frequência do EEG tanto na frequência fundamental do estímulo quanto em seus harmônicos (GARCIA-MOLINA; ZHU, 2011). Essa característica tem permitido o desenvolvimento de sistemas ICM robustos, frequentemente atingindo taxas de acerto em torno de 90% em experimentos baseados em SSVEP (LEITE et al., 2015).

Um exemplo visual do processo de coleta de dados de EEG baseados em SSVEP é apresentado na Figura 3



**Figura 3 – Resposta SSVEP a estímulos codificados por frequência na região occipital do cérebro.**  
Fonte: (DEMIR; ARSLAN; UYSAL, 2019)

A utilização do SSVEP favorece a criação de interfaces cérebro-máquina, uma vez que proporciona altas taxas de transferência de informação (ITR, do inglês *Information Transfer Rate*) e requer menor tempo de treinamento quando comparado a outros paradigmas ICM (LEITE

et al., 2015). Através da eletroencefalografia, essas respostas podem ser captadas e interpretadas para controlar dispositivos externos, estabelecendo uma ponte entre a atividade cerebral do usuário e as ações executadas por um sistema computacional (DIAS, 2023).

## 2.4 Pré-processamento de Sinais de EEG

Os sinais de EEG são altamente suscetíveis a ruídos, devido à presença de outras atividades bioelétricas no corpo humano. Entre as principais fontes de interferência estão os movimentos oculares (como piscadas e sacadas), a atividade cardíaca, contrações musculares e outros artefatos fisiológicos (KANG; AL., 2014). Esses ruídos podem mascarar ou distorcer os sinais cerebrais de interesse, tornando essencial a aplicação de técnicas de pré-processamento para sua atenuação ou remoção.

A fim de reduzir as distorções causadas por esses ruídos, diversas técnicas de pré-processamento, como filtragem e extração de épocas, são empregadas para remover artefatos e isolar as faixas de frequência de interesse. Entre as técnicas mais comuns estão a filtragem de ruídos e a normalização dos sinais. Este processo visa melhorar a relação sinal-ruído (SNR) e garantir que os dados reflitam, com maior fidelidade, a atividade cerebral relacionada às tarefas propostas. (ACUNZO; MACKENZIE; van Rossum, 2012). O objetivo final desta etapa é destacar as características únicas dos sinais antes de entregá-los aos algoritmos de classificação.

### 2.4.1 Filtro Passa-Faixa

No caso de sistemas baseados em SSVEP, os estímulos visuais modulam a resposta bioelétrica cerebral em torno de uma frequência específica de estimulação, denotada por  $f_0$ . Assim, uma estratégia eficaz para realçar as informações relevantes do sinal é a aplicação de um filtro centrado em  $f_0$ , de modo a captar com maior precisão os componentes espectrais induzidos pelo estímulo visual (DIAS, 2023).

Uma forma eficaz de aplicar essa técnica é por meio de um filtro passa-faixa, que permite isolar uma faixa específica de frequências ao atenuar as demais componentes do sinal. Suas vantagens envolvem a facilidade de implementação e sua baixa complexidade computacional, sendo uma excelente solução para problemas de processamento digital de sinais.

Um aspecto fundamental na implementação de filtros passa-faixa é o impacto da ordem do filtro sobre a resposta obtida. A ordem define a capacidade do filtro em atenuar frequências fora da faixa de interesse: filtros de ordem mais alta promovem uma transição mais abrupta entre a banda passante e as bandas de rejeição, o que resulta em uma maior eliminação de componentes indesejáveis. No entanto, esse ganho em seletividade vem acompanhado de um aumento na complexidade computacional e, em alguns casos, na possibilidade de instabilidade numérica, especialmente em implementações com filtros IIR.

### 2.4.2 Extração de Épocas

A extração de épocas é uma etapa no processamento de sinais eletroencefalográficos (EEG) onde o sinal é segmentado em intervalos menores e bem definidos, chamados de "épocas" (ou *epochs*, em inglês). Essas épocas representam janelas de tempo fixas que podem estar associadas a estímulos ou eventos de interesse, permitindo uma análise mais detalhada e localizada do comportamento do sinal.

A segmentação é geralmente realizada com base em marcadores temporais inseridos durante a aquisição do sinal, que indicam o início de um estímulo ou ação específica. A partir desses marcadores, são extraídos trechos do sinal com duração previamente definida, que podem incluir períodos anteriores (pré-estímulo) e posteriores (pós-estímulo) ao evento. Essa abordagem facilita a comparação entre diferentes tentativas e a aplicação de técnicas de extração de características, uma vez que garante uma padronização na análise dos dados.

Além disso, ao reduzir o sinal contínuo a janelas menores, a extração de épocas contribui para uma diminuição da complexidade computacional e possibilita o uso de algoritmos de aprendizado de máquina e estatística de forma mais eficiente.

## 2.5 Extração de *Features*

Os sinais de EEG apresentam uma ampla variedade de características, como frequência, potência e localização espacial, entre outras (LOTTE; CONGEDO, 2016). Essa diversidade, aliada à natureza complexa dos sinais, torna inviável a utilização direta dos dados brutos, tanto pela alta dimensionalidade quanto pelo volume de informações envolvidas. Além disso, trabalhar com todas as variáveis disponíveis pode comprometer o desempenho dos algoritmos de aprendizado de máquina, devido à "maldição da dimensionalidade", fenômeno em que o aumento excessivo de atributos demanda mais dados para uma boa generalização e eleva o risco de *overfitting*, além de aumentar significativamente o custo computacional.

Diante disso, torna-se essencial selecionar e extrair as características mais relevantes, de modo a representar o sinal de forma mais compacta e informativa, preservando os padrões fundamentais para a tarefa de classificação (LOTTE; CONGEDO, 2016). Essa etapa, conhecida como extração de características (ou *features*), é um dos pilares na construção de sistemas baseados em EEG.

A extração de *features* consiste em transformar os sinais brutos em representações de menor dimensão, porém capazes de capturar informações discriminativas relacionadas à intenção do usuário. Tais representações facilitam o uso de modelos estatísticos ou algoritmos de aprendizado de máquina, tornando o sistema mais eficiente e preciso. Dentre os métodos mais utilizados para essa finalidade, destaca-se o periodograma, amplamente empregado na análise espectral de sinais.

### 2.5.1 Periodograma

O periodograma é uma técnica amplamente utilizada para estimar a densidade espectral de potência (PSD, do inglês *Power Spectral Density*) de um sinal. Em termos gerais, ele fornece uma representação da distribuição de energia do sinal ao longo do espectro de frequências, permitindo identificar quais componentes espectrais são mais importantes em um determinado intervalo temporal (DIAS, 2023).

No contexto dos sinais SSVEP, o uso do periodograma é particularmente eficaz para quantificar a atividade cerebral nas proximidades da frequência de estímulo  $f_0$ . Essa estimativa permite mensurar a energia presente nessas bandas específicas e identificar padrões relevantes que podem servir como características discriminativas para os classificadores utilizados no sistema.

A estimativa da PSD é obtida através da Transformada Rápida de Fourier (FFT, do inglês *Fast Fourier Transform*) sobre janelas temporais do sinal (também chamadas de épocas), convertendo-o do domínio do tempo para o domínio da frequência. Esse processo resulta em um espectro contínuo de frequências, que evidencia as frequências predominantes e sua respectiva energia (DIAS, 2023). A densidade espectral gerada serve, então, como vetor de características a ser utilizado na etapa de classificação.

Para sinais discretos, como os sinais de EEG utilizados neste trabalho, o periodograma é computado por meio da Equação 2.1:

$$S\left(\frac{k}{NT}\right) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk} \right|^2, \quad \forall k = 0, 1, \dots, N-1, \quad (2.1)$$

onde  $S$  é o periodograma,  $k$  representa o ordinal no domínio da frequência,  $n$  representa o ordinal no domínio do tempo,  $N$  é a sequência a ser transformada e  $T$  é o período de amostragem. (CHOWDHURY et al., 2020).

## 2.6 Tratamento das *Features*

Os dados obtidos por meio da aplicação do periodograma foram posteriormente submetidos a cinco diferentes estratégias de processamento antes de serem utilizados nos classificadores, com o objetivo de avaliar possíveis melhorias no desempenho de classificação.

Para ilustrar os métodos aplicados, considere uma matriz de características (*features*) com dimensões  $a \times b$ , na qual cada linha representa uma amostra e cada coluna corresponde a uma característica extraída. Essa estrutura é exemplificada na Equação 2.2.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,b-1} & x_{1,b} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,b-1} & x_{2,b} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{a,1} & x_{a,2} & \cdots & x_{a,b-1} & x_{a,b} \end{bmatrix} \quad (2.2)$$

### 2.6.1 Tratamento 1: Normalização de *features*

Para a primeira forma de tratamento, os dados foram normalizados através da divisão de cada elemento das colunas da matriz de *features* pela soma dos elementos daquela coluna, denominado peso. O cálculo realizado nesta etapa é apresentado na Equação 2.3.

$$\text{peso}_j = \sum_{i=1}^a x_{i,j}, \quad (2.3)$$

onde  $k$  representa a coluna de interesse a qual será somada e  $a$  é o número total de linhas da matriz de *features*.

Assim, a nova matriz de *features* é apresentada na Equação 2.4.

$$X^{(1)} = \begin{bmatrix} \frac{x_{1,1}}{\text{peso}_1} & \frac{x_{1,2}}{\text{peso}_2} & \cdots & \frac{x_{1,b-1}}{\text{peso}_{b-1}} & \frac{x_{1,b}}{\text{peso}_b} \\ \frac{x_{2,1}}{\text{peso}_1} & \frac{x_{2,2}}{\text{peso}_2} & \cdots & \frac{x_{2,b-1}}{\text{peso}_{b-1}} & \frac{x_{2,b}}{\text{peso}_b} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{x_{a,1}}{\text{peso}_1} & \frac{x_{a,2}}{\text{peso}_2} & \cdots & \frac{x_{a,b-1}}{\text{peso}_{b-1}} & \frac{x_{a,b}}{\text{peso}_b} \end{bmatrix} \quad (2.4)$$

### 2.6.2 Tratamento 2: Normalização Z-score Global

A segunda forma de tratamento adotada baseou-se na normalização do tipo Z-score, também conhecida como padronização. Esse método foi aplicado utilizando a média global dos dados e o desvio padrão global da matriz de características.

O cálculo da média global da matriz de *features* é apresentada na Equação 2.5:

$$\mu = \frac{1}{a \cdot b} \sum_{i=1}^a \sum_{j=1}^b x_{i,j}. \quad (2.5)$$

O desvio-padrão global da matriz de *features* é calculado com base na Equação 2.6:

$$\sigma = \sqrt{\frac{1}{a \cdot b} \sum_{i=1}^a \sum_{j=1}^b (x_{i,j} - \mu)^2}. \quad (2.6)$$

Após o tratamento 2, a nova matriz de *features* é apresentada na Equação 2.7

$$X^{(2)} = \begin{bmatrix} \frac{x_{1,1}-\mu}{\sigma} & \frac{x_{1,2}-\mu}{\sigma} & \dots & \frac{x_{1,b-1}-\mu}{\sigma} & \frac{x_{1,b}-\mu}{\sigma} \\ \frac{x_{2,1}-\mu}{\sigma} & \frac{x_{2,2}-\mu}{\sigma} & \dots & \frac{x_{2,b-1}-\mu}{\sigma} & \frac{x_{2,b}-\mu}{\sigma} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{x_{a,1}-\mu}{\sigma} & \frac{x_{a,2}-\mu}{\sigma} & \dots & \frac{x_{a,b-1}-\mu}{\sigma} & \frac{x_{a,b}-\mu}{\sigma} \end{bmatrix} \quad (2.7)$$

### 2.6.3 Tratamento 3: Normalização *Z-score* de *features*

A terceira forma de tratamento consiste em uma variação da normalização *Z-score*, na qual a média e o desvio padrão globais são substituídos pela média e desvio padrão calculados individualmente para cada coluna da matriz de *features*.

O cálculo da média por coluna da matriz de *features* é apresentada na Equação 2.8:

$$\mu_j = \frac{1}{a} \sum_{i=1}^a x_{i,j} \quad (2.8)$$

O desvio-padrão por coluna da matriz de *features* é calculado com base na Equação 2.9:

$$\sigma_j = \sqrt{\frac{1}{a} \sum_{i=1}^a (x_{i,j} - \mu_j)^2} \quad (2.9)$$

Após o tratamento 3, a nova matriz de *features* é apresentada na Equação 2.10

$$X^{(3)} = \begin{bmatrix} \frac{x_{1,1}-\mu_1}{\sigma_1} & \frac{x_{1,2}-\mu_2}{\sigma_2} & \dots & \frac{x_{1,b-1}-\mu_{b-1}}{\sigma_{b-1}} & \frac{x_{1,b}-\mu_b}{\sigma_b} \\ \frac{x_{2,1}-\mu_1}{\sigma_1} & \frac{x_{2,2}-\mu_2}{\sigma_2} & \dots & \frac{x_{2,b-1}-\mu_{b-1}}{\sigma_{b-1}} & \frac{x_{2,b}-\mu_b}{\sigma_b} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{x_{a,1}-\mu_1}{\sigma_1} & \frac{x_{a,2}-\mu_2}{\sigma_2} & \dots & \frac{x_{a,b-1}-\mu_{b-1}}{\sigma_{b-1}} & \frac{x_{a,b}-\mu_b}{\sigma_b} \end{bmatrix} \quad (2.10)$$

### 2.6.4 Tratamento 4: Normalização *Z-score* de amostras

A quarta forma de tratamento consiste em uma variação da normalização *Z-score*, na qual a média e o desvio padrão globais são substituídos pela média e desvio padrão calculados individualmente para cada linha da matriz de *features*, assim normalizando cada característica e não cada amostra.

O cálculo da média por linha da matriz de *features* é apresentada na Equação 2.11:

$$\mu_i = \frac{1}{b} \sum_{j=1}^b x_{i,j} \quad (2.11)$$

O desvio-padrão por linha da matriz de *features* é calculado com base na Equação 2.12:

$$\sigma_i = \sqrt{\frac{1}{b} \sum_{j=1}^b (x_{i,j} - \mu_i)^2} \quad (2.12)$$

Após o tratamento 4, a nova matriz de *features* é apresentada na Equação 2.13

$$X^{(4)} = \begin{bmatrix} \frac{x_{1,1}-\mu_1}{\sigma_1} & \frac{x_{1,2}-\mu_1}{\sigma_1} & \dots & \frac{x_{1,b-1}-\mu_1}{\sigma_1} & \frac{x_{1,b}-\mu_1}{\sigma_1} \\ \frac{x_{2,1}-\mu_2}{\sigma_2} & \frac{x_{2,2}-\mu_2}{\sigma_2} & \dots & \frac{x_{2,b-1}-\mu_2}{\sigma_2} & \frac{x_{2,b}-\mu_2}{\sigma_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{x_{a,1}-\mu_a}{\sigma_a} & \frac{x_{a,2}-\mu_a}{\sigma_a} & \dots & \frac{x_{a,b-1}-\mu_a}{\sigma_a} & \frac{x_{a,b}-\mu_a}{\sigma_a} \end{bmatrix} \quad (2.13)$$

## 2.7 Algoritmos de Classificação

Algoritmos de classificação são técnicas de aprendizado supervisionado que buscam estimar, de forma automática, a classe à qual um determinado dado pertence, com base em seu vetor de características extraídas (LOTTE et al., 2007). No contexto da análise de sinais de EEG, os classificadores desempenham o papel de interpretar a atividade elétrica cerebral registrada e traduzi-la em comandos ou intenções do usuário (JORGE, 2024). Foram escolhidos para este trabalho 3 classificadores multiclasse distintos: LDA, KNN e *Random Forest*. A descrição teórica e a modelagem matemática de cada um é descrita a seguir.

### 2.7.1 Análise Discriminante Linear (LDA)

A Análise Discriminante Linear (LDA, do inglês *Linear Discriminant Analysis*) é um algoritmo de classificação amplamente utilizado, cuja principal função é encontrar um hiperplano que melhor separa os dados de diferentes classes no espaço de características (LOTTE et al., 2007). No contexto da classificação de sinais de EEG, o LDA está entre os classificadores mais populares, destacando-se por sua simplicidade, eficiência computacional e bom desempenho em tarefas com alta dimensionalidade e número limitado de amostras (CALVINHO, 2023).

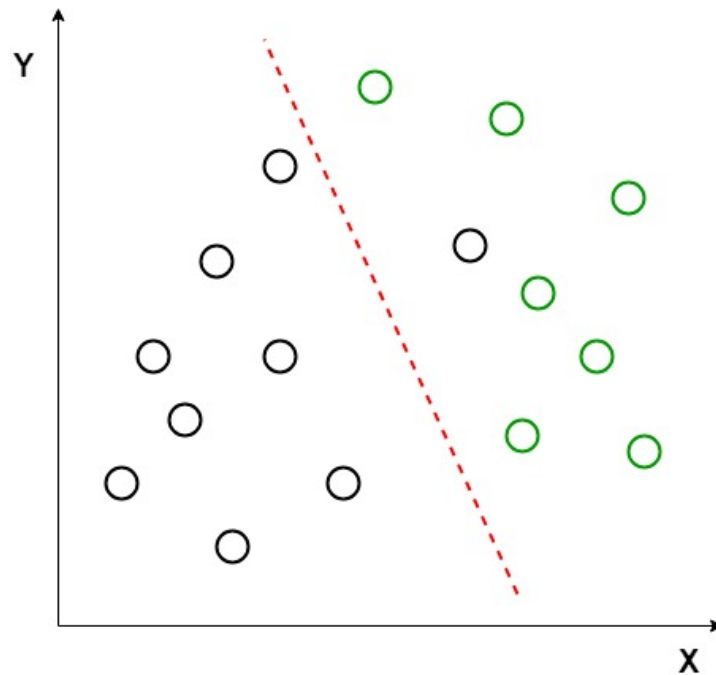
A Figura 4 exemplifica o funcionamento do LDA graficamente. A linha tracejada em vermelho representa o hiperplano de separação dos dados.

Em sistemas de Interface Cérebro-Máquina (ICM) baseados em SSVEP, classificadores lineares como o LDA são eficazes na distinção de padrões de ativação cerebral relacionados a diferentes frequências de estímulo visual.

A separação das classes é realizada por meio da Equação 2.14:

$$Y = w^T X, \quad (2.14)$$

em que  $Y$  representa a saída do classificador,  $X$  é o vetor de características extraídas do sinal e  $w$  é o vetor de pesos. O expoente  $T$  indica a transposição do vetor, permitindo o produto interno



**Figura 4 – Exemplo visual do funcionamento do LDA. Fonte: GEEKSFORGEEKS (2020).**

entre  $w$  e  $X$ , cuja finalidade é projetar os dados em um novo espaço onde a separação entre as classes seja maximizada (SENA, 2018).

Para determinar o vetor  $w$  utiliza-se uma técnica de treinamento do classificador que busca maximizar o chamado Critério de Fisher.

O Critério de Fisher é uma medida estatística usada para encontrar a matriz de projeção,  $w$ , que garante a máxima separabilidade maximizando a razão entre a variância entre classes,  $S_B$ , para a variância dentro da classe,  $S_W$  (JORGE, 2022).

O critério de Fisher é descrito na Equação 2.15:

$$J(w) = \frac{|w^T S_B w|}{|w^T S_W w|}, \quad (2.15)$$

onde  $S_B$  e  $S_W$  são definidos, respectivamente, nas Equações 2.16 e 2.17 como:

$$S_W = \sum_{i=1}^C S_i = \sum_{i=1}^C \sum_{x_k \in \omega_i} (x_k - \mu'_i)(x_k - \mu'_i)^T \quad (2.16)$$

$$S_B = \sum_{i=1}^C n_i (\mu'_i - \mu)(\mu'_i - \mu)^T, \quad (2.17)$$

onde  $C$  é o número de classes distintas,  $S_i$  é a matriz de espalhamento de cada uma das classes,  $\omega_i$  é a matriz com todos os elementos de uma mesma classe,  $\mu'_i$  é a média de uma classe e  $\mu$  é a média global entre todas as classes (LEVADA, 2021).

A definição de  $\mu$  é apresentada na equação 2.5, e a definição de  $\mu'_i$  é apresentada na Equação 2.18 como:

$$\mu'_i = \frac{1}{n_i} \sum_{x_j \in \omega_i} x_j, \quad (2.18)$$

onde  $n_i$  é o número total de elementos da  $i$ -ésima classe.

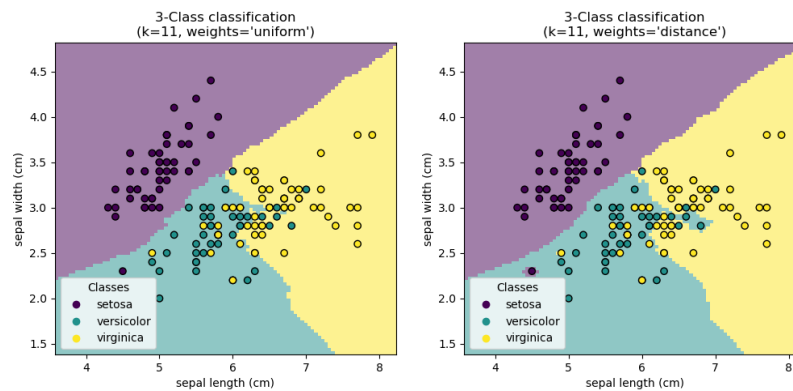
## 2.7.2 KNN

O KNN (do inglês *K Nearest Neighbors*) é um dos algoritmos de classificação mais fáceis de serem utilizados em vários problemas. Seu funcionamento baseia-se na ideia de classificar uma nova amostra com base nas amostras mais próximas a ela em um conjunto de treinamento (MLADENOVA; VALOVA, 2022)

Assim como outros classificadores, o KNN depende da definição de hiperparâmetros para sua correta aplicação. O principal deles é o valor de  $K$ , que determina o número de vizinhos mais próximos considerados na decisão de classe para um novo dado (JORGE, 2024). A escolha adequada desse valor é essencial para o desempenho do classificador: valores muito baixos de  $K$  podem levar ao *overfitting*, (do inglês, sobreajuste) enquanto valores muito altos podem resultar em *underfitting* (do inglês, subajuste), comprometendo a capacidade de generalização do modelo e, conseqüentemente, sua acurácia (JORGE, 2024).

Uma vez escolhido o parâmetro  $K$ , o classificador irá então calcular as distâncias entre os dados de entrada e os dados do conjunto de treinamento. As classes das  $K$  amostras mais próximas são então consideradas, e a classe mais frequente entre elas é atribuída à nova amostra.

A Figura 5 exemplifica graficamente o funcionamento do KNN.



**Figura 5 – Exemplo visual do funcionamento do KNN. Fonte: SCIKIT-LEARN (2024).**

No contexto computacional da presente aplicação, o cálculo dessas distâncias pode ser feito de duas formas diferentes: a primeira é a distância Euclidiana, a qual busca calcular a distância entre dois pontos como a raiz quadrada da soma das diferenças ao quadrado de todas as dimensões de um vetor, como apresentado na Equação 2.19:

$$D_{euclidian} = \sqrt{\sum_{i=1}^N (\hat{X}_i - X_i)^2}, \quad (2.19)$$

onde  $\hat{X}$  e  $X$  representam respectivamente o vetor de teste e o vetor de treinamento com  $N$  características e o índice  $i$  representa o número dessa característica (JORGE, 2024).

Já outra forma utilizada para calcular a distância entre dois pontos é a distância de Manhattan, a qual calcula a distância entre dois pontos como a soma do módulo das diferenças em todas dimensões de um vetor, como apresentado na Equação 2.20:

$$D_{manhattan} = \sum_{i=1}^N |\hat{X}_i - X_i| \quad (2.20)$$

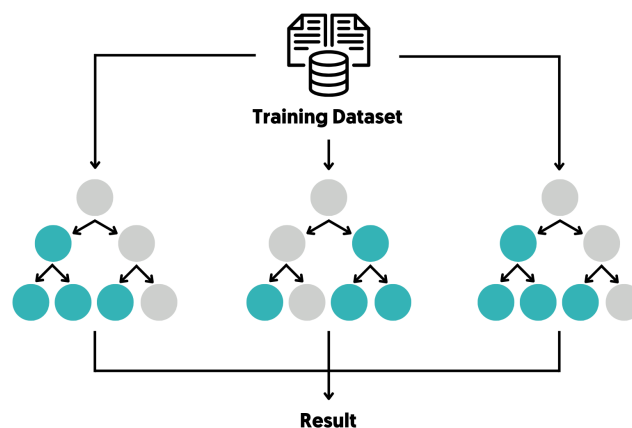
A escolha da métrica de distância também pode influenciar significativamente o desempenho do KNN, especialmente em contextos com alta dimensionalidade ou quando os dados possuem escalas diferentes.

### 2.7.3 Random Forest

O algoritmo *Random Forest* é um método de classificação baseado em um conjunto de Árvores de Decisão (DT, do inglês *Decision Trees*). A ideia central por trás desse algoritmo é combinar diversas árvores, cada uma com relativa independência, a fim de melhorar a robustez e a acurácia do processo de classificação (BREIMAN, 2001).

As Árvores de Decisão são algoritmos de aprendizado supervisionado que particionam o espaço de dados em regiões homogêneas com base em regras hierárquicas. Cada divisão ocorre em um nó de decisão, onde uma regra é aplicada a uma das *features* do conjunto de dados. O objetivo é que, ao final da árvore, os dados estejam agrupados em folhas (ou *leaves* em inglês) compostas, idealmente, apenas por amostras de uma única classe (MITCHELL, 1997).

A Figura 6 apresenta um fluxograma simplificado do funcionamento do algoritmo *Random Forest*.



**Figura 6 – Fluxograma simplificado do funcionamento do *Random Forest*. Fonte: DIDA (2024).**

A escolha das regras de divisão nos nós é feita com base em critérios como o ganho de informação, o qual visa selecionar a característica que mais reduz a entropia do sistema. A

entropia, nesse contexto, é uma medida da incerteza ou desordem do conjunto de dados. Quanto maior a entropia, maior é o grau de aleatoriedade, dificultando a classificação. Ela pode ser calculada pela Equação 2.21:

$$\text{Entropia} = - \sum_{i=1}^N p_i \log_2(p_i), \quad (2.21)$$

em que  $p_i$  representa a proporção de elementos da classe  $i$  no nó em análise.

Apesar de sua simplicidade e interpretabilidade, Árvores de Decisão individuais tendem a ser sensíveis a pequenas variações nos dados de treinamento, o que pode resultar em sobreajuste (*overfitting*) e variações significativas no desempenho de classificação.

Para mitigar essa limitação, foi proposto o classificador *Random Forest*, que consiste em um conjunto de Árvores de Decisão treinadas de forma independente. Cada árvore recebe uma amostra diferente dos dados, gerada por meio do processo de *Bootstrapping* (amostragem com reposição). A classificação final é definida com base em um esquema de votação por maioria entre as árvores. Essa abordagem é conhecida como *Bagging* (junção das palavras *Bootstrap* e *Aggregating*), e contribui para a redução da variância do modelo sem aumentar significativamente o viés, resultando em um classificador mais estável e com melhor capacidade de generalização (GÉRON, 2019).

## 2.8 Otimização Bayesiana

Durante a construção de modelos de classificação, é necessário ajustar os chamados hiperparâmetros, que são parâmetros externos ao modelo em si. Ao contrário dos parâmetros internos, que são aprendidos automaticamente durante o treinamento (como os pesos em um classificador linear), os hiperparâmetros devem ser definidos manualmente pelo usuário (JORGE, 2024).

A escolha adequada dos hiperparâmetros é considerada uma das etapas mais críticas na implementação de classificadores, pois impacta diretamente o comportamento do modelo, como sua capacidade de generalização, tempo de processamento e desempenho em métricas como acurácia e precisão (MITRA; BIESSMANN, 2024). Técnicas de otimização são frequentemente empregadas para auxiliar nessa seleção.

Uma das técnicas mais populares utilizadas para a solução do problema de otimização de parâmetros é a chamada Otimização Bayesiana (OB), a qual busca encontrar os melhores hiperparâmetros para um modelo de aprendizado supervisionado com base no Teorema Bayesiano. A OB incorpora o conhecimento prévio sobre a função objetivo e atualiza o conhecimento posterior, aumentando a precisão e reduzindo a perda do modelo (JORGE, 2024).

O método de Otimização Bayesiana é empregado para ajustar os hiperparâmetros de cada modelo, com o objetivo de maximizar seu desempenho nas métricas de avaliação e, assim,

obter os melhores resultados possíveis.

## 2.9 Métricas de Avaliação

Para avaliar o desempenho de um classificador, é fundamental definir métricas que forneçam uma visão clara, objetiva e quantitativa sobre sua qualidade e confiabilidade. Neste trabalho, são utilizadas duas métricas: acurácia e precisão.

Essas métricas são calculadas com base na chamada matriz de confusão, uma ferramenta que resume graficamente o desempenho de um modelo de classificação ao comparar os rótulos reais das amostras com os rótulos previstos. A Figura 7 ilustra um exemplo de matriz de confusão utilizada como base para os cálculos das métricas descritas.

X		Dado de Referência		
		A	B	C
Dado Classificado	A	VA	FP	FP
	B	FN	VB	FP
	C	FN	FN	VC

Figura 7 – Exemplo de Matriz de Confusão. Fonte: (AMBIENTAL, 2020).

### 2.9.1 Acurácia

A acurácia é uma métrica amplamente utilizada para avaliar o desempenho geral de um classificador. Ela representa a proporção de previsões corretas (tanto positivas quanto negativas) em relação ao total de amostras avaliadas.

A acurácia é definida na Equação 2.22:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}, \quad (2.22)$$

onde VP são os verdadeiros positivos, VN os verdadeiros negativos, FP os falsos positivos e FN os falsos negativos. Apesar de ser uma métrica intuitiva, a acurácia pode ser enganosa em casos de classes desbalanceadas, motivo pelo qual é recomendável analisá-la em conjunto com outras métricas.

### 2.9.2 Precisão

A precisão é uma métrica que indica a proporção de previsões corretas entre todas as previsões positivas realizadas pelo classificador. Em outras palavras, ela mede o quanto das classificações positivas feitas pelo modelo são, de fato, verdadeiras. Essa métrica é especialmente útil em cenários onde os falsos positivos são críticos.

A precisão é definida na Equação 2.23:

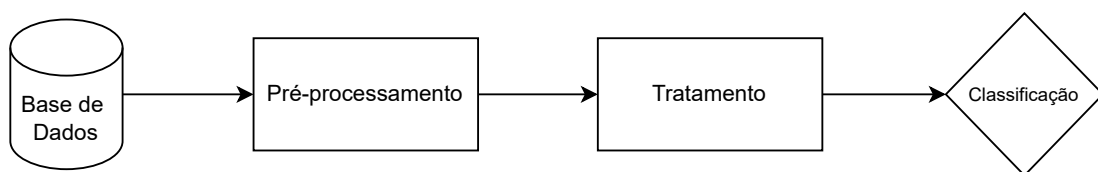
$$\text{Precisão} = \frac{VP}{VP + FP}, \quad (2.23)$$

onde  $VP$  representa os verdadeiros positivos e  $FP$  os falsos positivos. Um valor alto de precisão indica que o classificador comete poucos erros ao prever a classe positiva.

### 3 METODOLOGIA

O presente trabalho busca avaliar a aplicação de algoritmos de classificação para identificação de sinais de SSVEP em diferentes características de filtragem, organização dos dados e aleatoriedade, a fim de analisar os parâmetros e determinar as variáveis ideais para a análise e classificação dos dados.

Nesta etapa, é feita uma descrição da base de dados assim como das etapas de pré-processamento, tratamento e classificação dos dados. A Figura 8 exemplifica visualmente o fluxograma completo desenvolvido nas etapas anteriores e nesta.



**Figura 8 – Diagrama Simplificado do Processo Classificação. Fonte: O autor.**

#### 3.1 Descrição da Base de dados

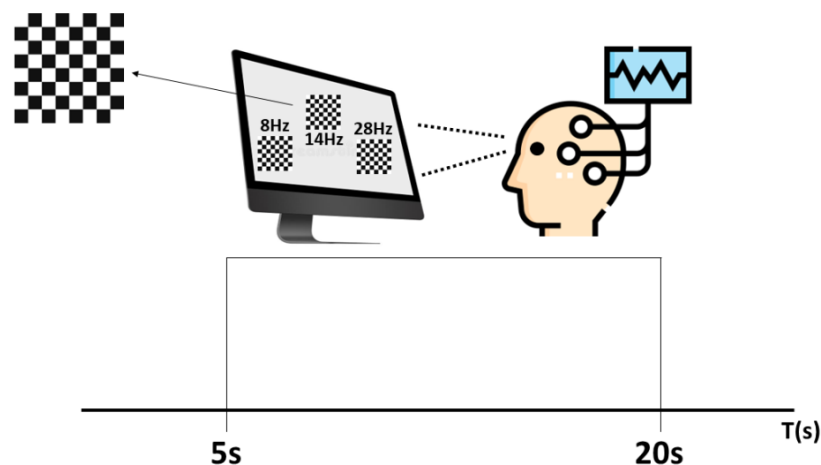
Para o desenvolvimento deste trabalho, é utilizado um conjunto de dados público sob os direitos do RIKEN-LABSP e do Dr. Hovagim Bakardjian. O conjunto é composto por sinais de EEG de quatro sujeitos, registrados enquanto cada um era exposto a três diferentes frequências de estímulo visual: 8 Hz, 14 Hz e 28 Hz. Cada participante realizou cinco tentativas para cada frequência de estímulo.

Os dados são adquiridos por meio de 128 canais, com os eletrodos posicionados sobre o couro cabeludo de acordo com o sistema internacional 10-10, como apresentado na Figura 2. A frequência de amostragem do experimento é de 256 Hz, ou seja, 256 amostras por segundo em cada eletrodo.

Cada tentativa teve duração total de 24 segundos, sendo que os estímulos visuais foram aplicados no intervalo entre 5 e 20 segundos, conforme ilustrado na Figura 9.

#### 3.2 Ambiente Computacional e Tecnologias Utilizadas

As análises deste trabalho foram desenvolvidas no ambiente de programação *Visual Studio Code*, utilizado tanto para a escrita quanto para a execução dos códigos. A linguagem adotada é o *Python*, escolhida por sua sintaxe simples, alto nível de abstração e ampla comunidade de usuários. Além disso, a linguagem oferece uma vasta coleção de bibliotecas específicas para análise de sinais, aprendizado de máquina e visualização de dados, o que a torna especialmente adequada para o desenvolvimento de aplicações em interfaces cérebro-máquina (ICM).



**Figura 9 – Exemplo do experimento e da duração. Fonte:(DIAS, 2023).**

As principais bibliotecas utilizadas no desenvolvimento das análises foram: *SciPy*, empregada na implementação de filtros e na análise da resposta em frequência dos sinais; *scikit-learn*, responsável pela construção dos algoritmos de aprendizado de máquina, bem como pelas ferramentas de análise de desempenho e cálculo de métricas estatísticas; *NumPy*, utilizada para manipulação e processamento eficiente de *arrays* numéricos; e *Matplotlib*, adotada para a geração de gráficos e visualização dos resultados experimentais. Além disso, a biblioteca *Pandas* foi utilizada para organizar, estruturar e manipular os dados de saída dos experimentos, facilitando a análise tabular dos resultados obtidos.

A Tabela 1 apresenta um resumo das principais bibliotecas utilizadas neste trabalho.

**Tabela 1 – Principais bibliotecas utilizadas**

Biblioteca	Função	Descrição
Numpy	mean()	Função para calcular a média de um vetor de dados
	std()	Função para calcular o desvio-padrão de um vetor de dados
Scipy	butter()	Função que implementa o filtro Butterworth
	freqz()	Função que calcula a resposta em frequência de um filtro digital
Scikit-learn	LinearDiscriminantAnalysis()	Função que implementa o Classificador LDA
	KNeighborsClassifier()	Função que implementa o Classificador KNN
	RandomForestClassifier()	Função que implementa o Classificador Random Forest
Scikit-optimize	BayesSearchCV()	Função que implementa a Otimização Bayesiana

**Fonte: O autor.**

### 3.3 Pré-processamento dos dados

O pré-processamento representa uma das etapas mais importantes no fluxo de classificação de dados, pois é nesse momento que os sinais são filtrados e tratados com o objetivo de realçar as características mais relevantes para a distinção entre as classes.

#### 3.3.1 Remoção do sinal DC, filtragem e seleção do canal de interesse

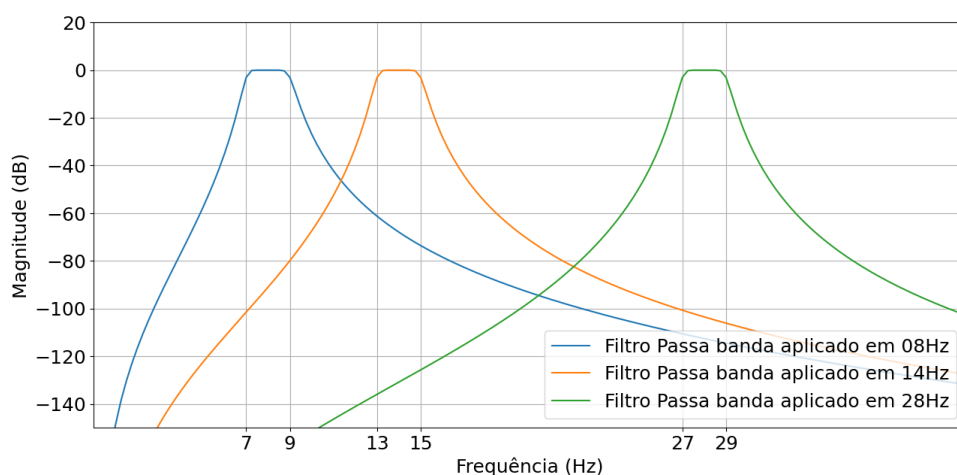
A etapa de filtragem dentro do pré-processamento tem como objetivo eliminar possíveis ruídos e preservar o máximo de informações relevantes do sinal. Para isso, foram desenvolvidas operações matemáticas no ambiente computacional.

A primeira operação realizada foi a remoção da componente DC de cada sinal. Essa etapa foi implementada com o uso da função *mean()* da biblioteca *NumPy*, que calcula o valor médio de cada sinal, o qual foi posteriormente subtraído do respectivo sinal bruto.

Para a etapa de filtragem dos sinais foram definidos os coeficientes para 3 filtros distintos, cada um configurado para filtrar os sinais ao redor das 3 frequências de interesse: 8 Hz, 14 Hz e 28 Hz. Esses coeficientes foram calculados pela função *butter()*, a qual pertence ao módulo *signal* da biblioteca *Scipy*.

Os filtros implementados foram então aplicados aos dados de EEG, de modo que cada frequência de estímulo fosse filtrada nas 3 frequências de interesse, resultando assim em 9 conjuntos de dados.

A Figura 10 apresenta a representação visual dos filtros. Esses resultados foram obtidos através do uso da função *freqz()*, também pertencente ao módulo *signal*.



**Figura 10 – Resposta em Frequência dos Filtros. Fonte: O autor.**

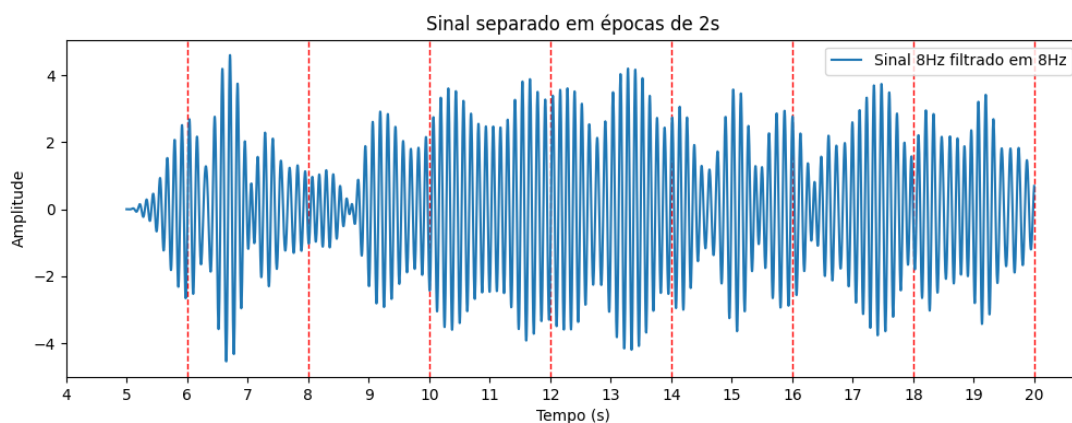
Após a etapa de filtragem, foram selecionados os sinais provenientes do eletrodo posicionado na região occipital do cérebro (Oz), área responsável pelo processamento da informação visual.

### 3.3.2 Aplicação da Extração de Épocas

A fim de simplificar e padronizar a análise dos dados, foi aplicada uma técnica chamada extração de épocas, a qual consiste em segmentar os dados em trechos de menor duração com período definido.

São escolhidas para essa metodologia 5 épocas distintas: 125 ms, 250 ms, 500 ms, 1 s e 2 s. A escolha dessas épocas é inspirada na análise desenvolvida pelo trabalho de (DIAS, 2023), onde esses períodos são utilizados a fim de destacar eventos relacionados à tarefa ou estímulo que o usuário está realizando.

A Figura 11 apresenta um exemplo visual da extração de épocas, ao apresentar o sinal de 8 Hz segmentado em trechos de 2 s.



**Figura 11 – Exemplo de extração de épocas do sinal. Fonte: O autor.**

Ao final da extração de épocas, cada conjunto de dados apresentou um número de elementos diferente. Esse processo gerou 35 exemplos para a época de 2 s, 75 exemplos para a época de 1 s, 150 exemplos para a época de 500 ms, 300 exemplos para a época de 250 ms e 600 exemplos para a época de 125 ms.

### 3.3.3 Aplicação do Periodograma e Extração de *Features*

A etapa final do pré-processamento foi a aplicação da técnica de periodograma. O periodograma é uma técnica capaz de extrair a densidade espectral de potência de um sinal, retornando assim a potência para cada componente de frequência do sinal.

A implementação prática desta técnica foi realizada por meio da função *periodogram()* da biblioteca *SciPy*, a qual calculou automaticamente a PSD para todas as durações de épocas definidas neste estudo: 125 ms, 250 ms, 500 ms, 1 s e 2 s. Os resultados da aplicação do periodograma nas diferentes janelas temporais estão ilustrados na Figura 12.

Esses valores de potência foram então somados, resultando em uma matriz que representa as características (*features*) extraídas do sinal. Essas *features* serão então tratadas, a fim de avaliar possíveis melhorias nos desempenhos de classificação dos classificadores.

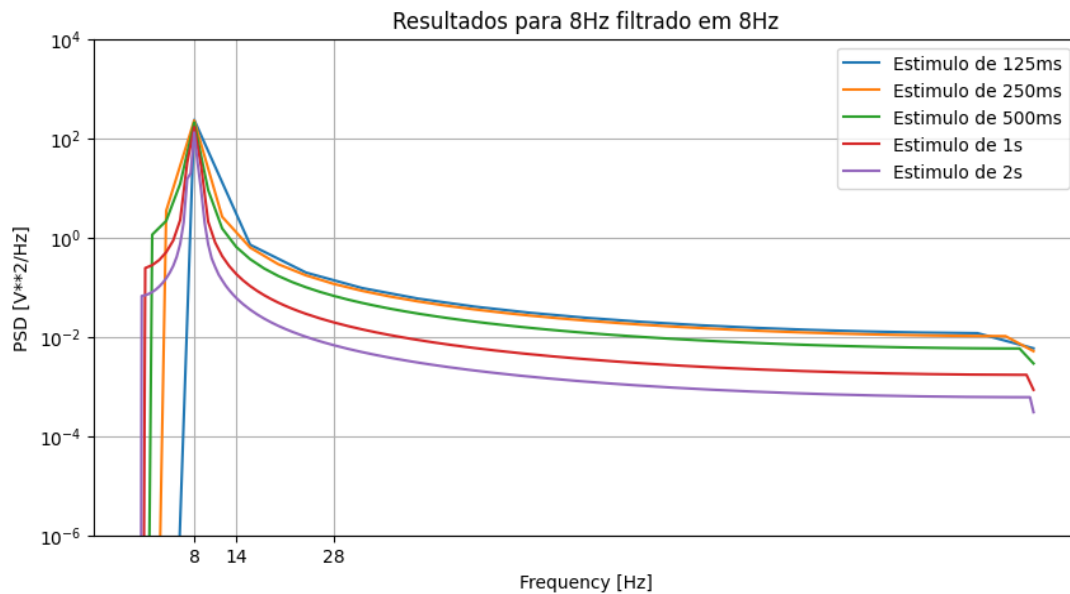


Figura 12 – Gráfico Periodogramas sinal 8Hz filtrado em 8Hz. Fonte: O autor.

### 3.4 Aplicação das Técnicas de Tratamento

São aplicadas cinco técnicas distintas de tratamento dos dados, com o objetivo de aprimorar a qualidade das *features* extraídas e, conseqüentemente, melhorar o desempenho dos algoritmos de classificação. Cada abordagem buscou realçar características relevantes dos sinais, reduzir variações indesejadas e padronizar os dados, permitindo uma comparação mais justa entre os modelos avaliados. As equações que descrevem os tratamentos aplicados estão descritas na seção 2.6.

A aplicação dessas técnicas foi implementada através do uso das funções *mean()* e *std()* da biblioteca *Numpy*, as quais realizaram os cálculos de média e desvio padrão para cada um dos tratamentos. Ao final desse processo, foram gerados 25 conjuntos de dados distintos, os quais foram posteriormente entregues aos classificadores.

### 3.5 Classificação dos Sinais de EEG.

Após a extração das *features* e o respectivo tratamento dos dados, a etapa final consistiu no treinamento e teste dos algoritmos de classificação, com o objetivo de avaliar sua capacidade de identificar corretamente a frequência de estímulo associada a cada amostra de EEG.

Essa etapa foi realizada utilizando três algoritmos distintos: LDA, KNN e *Random Forest*. Cada classificador recebeu 70% dos dados para treinamento, permitindo o ajuste automático de seus parâmetros internos a partir dos exemplos fornecidos. Os 30% restantes foram utilizados para validação, possibilitando a verificação do desempenho na classificação de dados inéditos.

As amostras utilizadas consistiam em matrizes de características correspondentes às

frequências de estímulo de 8 Hz, 14 Hz e 28 Hz. Cada matriz foi gerada a partir da filtragem do sinal original nas três bandas, sendo posteriormente submetida aos tratamentos descritos na Seção 2.6.

Cada classificador recebeu, portanto, nove conjuntos de dados compostos por amostras destinadas ao treinamento e outros nove conjuntos compostos por amostras destinadas ao teste. Cada conjunto continha amostras de uma única frequência de estímulo, processadas nas três bandas de filtragem e submetidas ao método de tratamento correspondente.

Por fim, os dados foram submetidos aos três classificadores, cujos hiperparâmetros foram otimizados via algoritmo de Otimização Bayesiana. Os resultados foram registrados por meio de matrizes de confusão, bem como pelas métricas de acurácia e precisão.

### 3.6 Otimização de Hiperparâmetros

A etapa de classificação constitui o ponto central do fluxograma apresentado, sendo fundamental a seleção criteriosa dos hiperparâmetros de cada classificador para garantir sua eficácia.

Para resolver esse problema, foi implementado um algoritmo de Otimização Bayesiana (OB) através do uso da função *BayesSearchCV()*, pertencente à biblioteca *scikit-optimize*. Essa função aplica a OB com base em um espaço de busca pré-definido e parâmetros de validação, a fim de otimizar os hiperparâmetros dos classificadores.

Os hiperparâmetros utilizados para configurar a função de Otimização Bayesiana (OB) determinam características como o número de iterações (*n\_iter*), número de validações cruzadas (*cv*), métrica de comparação entre resultados (*scoring*) e o estado aleatório das variáveis do classificador (*random\_state*). Os valores das variáveis adotadas neste estudo estão apresentados na Tabela 2.

**Tabela 2 – Hiperparâmetros da Função de OB**

Hiperparâmetro	Valor
<b>n_iter</b>	20
<b>cv</b>	5
<b>scoring</b>	"accuracy"
<b>random_state</b>	42

Fonte: O autor.

Para cada um dos classificadores, foi definido um espaço de busca para o qual o algoritmo de otimização realizou as simulações, a fim de definir os melhores valores para cada hiperparâmetro. A Tabela 3 apresenta os espaços de busca para cada um dos hiperparâmetros otimizados.

**Tabela 3 – Espaços de busca para os hiperparâmetros ajustáveis**

<b>Classificador</b>	<b>Hiperparâmetros</b>	<b>Valores</b>
<b>LDA</b>	n_components	[1, 2]
	shrinkage	[None, 'auto']
	solver	['lsqr', 'eigen']
<b>KNN</b>	n_neighbors	[1, 2, ..., 50]
	p	[1, 2]
	weights	['uniform', 'distance']
<b>RF</b>	bootstrap	['True', 'False']
	max_depth	[5, 6, ..., 50]
	max_features	['sqrt', 'log2', None]
	min_samples_leaf	[1, 2, ..., 20]
	min_samples_split	[2, 3, ..., 20]
	n_estimators	[50, 51, ..., 300]

**Fonte: O autor.**

## 4 RESULTADOS E DISCUSSÕES

Na Seção 4, são apresentados os resultados obtidos a partir da aplicação dos dados aos diferentes algoritmos de classificação avaliados neste trabalho. Para cada classificador, são exibidas a matriz de confusão, bem como as métricas de acurácia e precisão, considerando as distintas janelas temporais (épocas) e frequências de estímulo analisadas.

### 4.1 Resultados do Tratamento dos dados

#### 4.1.1 *Features* Geradas sem Tratamento

Como descrito na Seção 2.6, foram empregadas diferentes técnicas de tratamento das matrizes de *features* a fim de realizar uma análise comparativa dos resultados dos diferentes classificadores quando treinados com diferentes formas dos dados, assim, esta seção apresentará os resultados visuais da aplicação das técnicas nos dados utilizados neste trabalho.

Os plots de dados apresentados nas seções seguintes são todos resultados de diferentes técnicas de tratamento aplicados aos mesmos dados utilizados nas matrizes base de *features*, o plot dessas matrizes é apresentado na Figura 13:

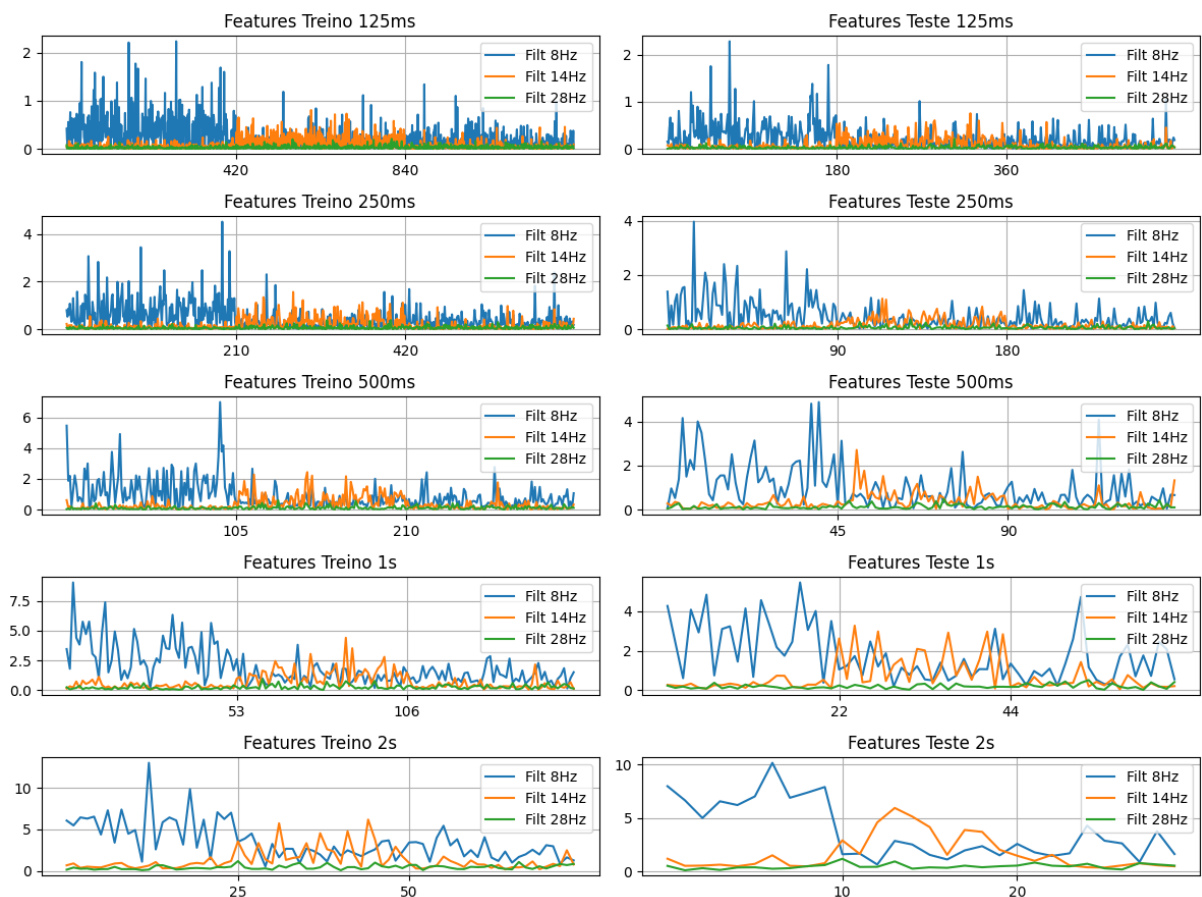
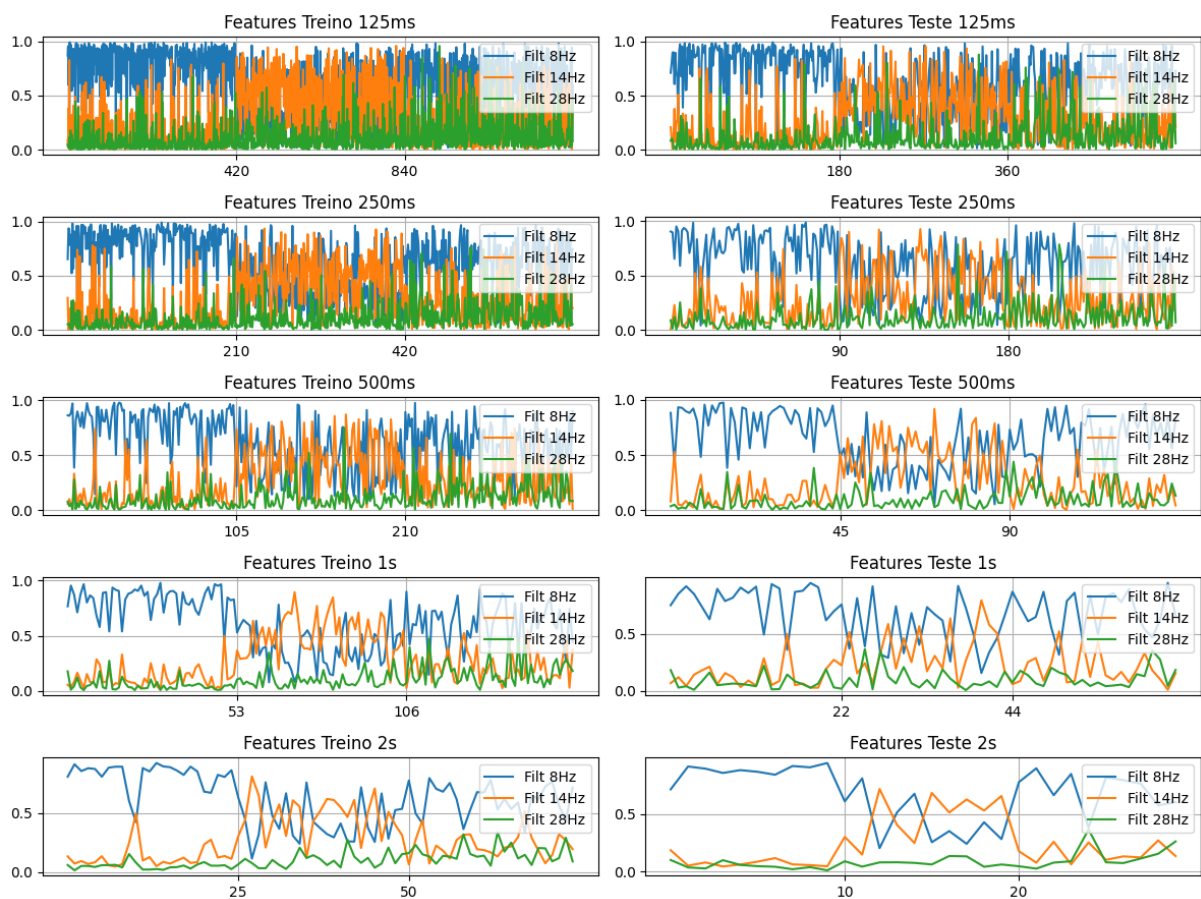


Figura 13 – *Features* 8Hz, 14Hz e 28Hz. Fonte: O autor.

### 4.1.2 Features Geradas com Tratamento 1

O Tratamento 1 consistiu na normalização dos dados por coluna, utilizando como base a soma dos elementos de cada coluna da matriz de *features*, conforme descrito na Seção 2.6.1. Essa abordagem visa igualar a influência de cada *feature* ao dividir os valores de cada coluna pelo seu respectivo "peso" (soma dos valores da coluna), de modo a normalizar os dados em uma escala entre 0 e 1.

A Figura 14 apresenta os gráficos das *features* extraídas após a aplicação desse tratamento. Observa-se que a amplitude das *features* foi normalizada, promovendo uma maior uniformização entre as colunas. Essa regularização por *feature* visa trazer resultados mais precisos em algoritmos de classificação como o KNN, onde variáveis com maiores magnitudes acabam por "dominar" o cálculo.



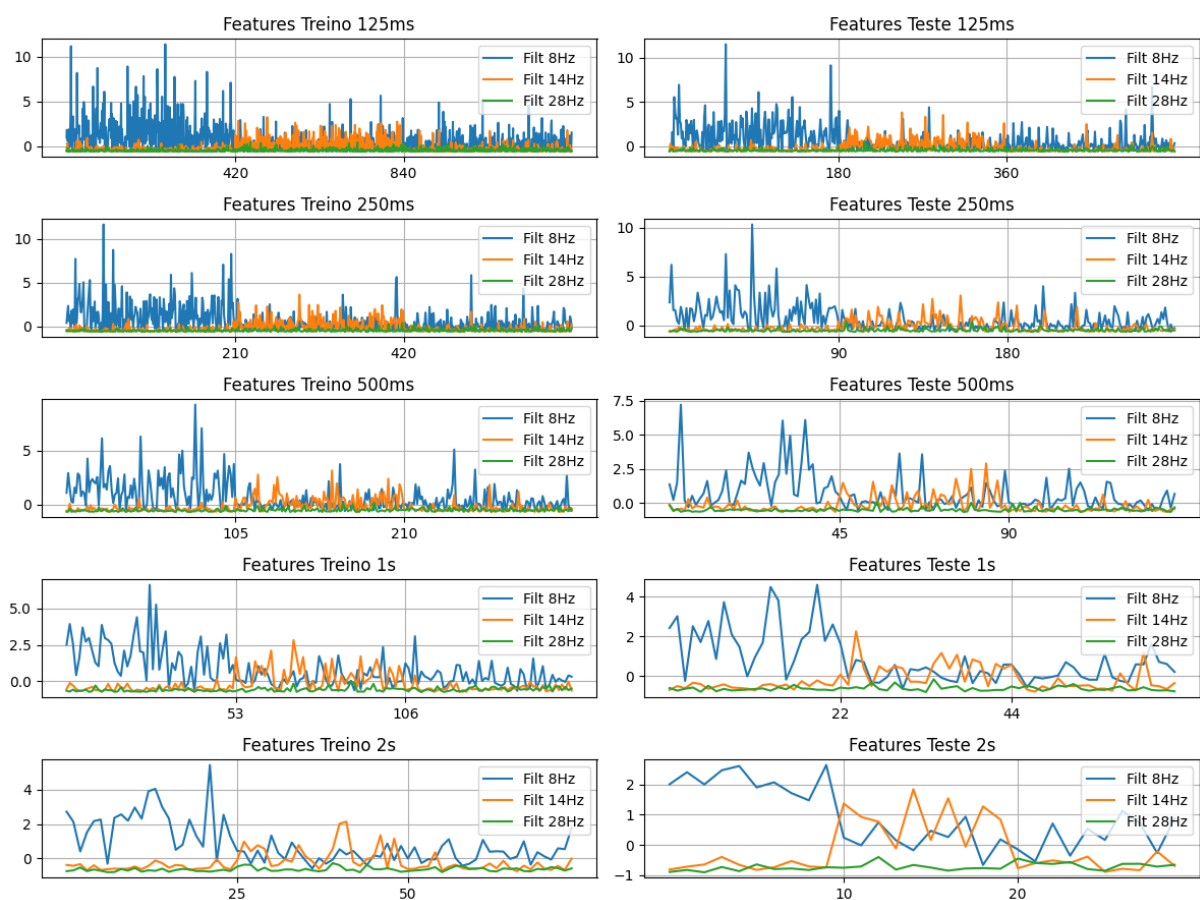
**Figura 14 – Features 8Hz, 14Hz e 28Hz após Tratamento 1. Fonte: O autor.**

Observa-se uma redução na dispersão e no contraste entre as classes, especialmente nas janelas mais curtas (125 ms e 250 ms), o que pode ter limitado a capacidade de separação entre os grupos durante o treinamento. Já nas janelas maiores (1 s e 2 s), os padrões por frequência permanecem mais evidentes, o que pode explicar a recuperação de desempenho observada nas métricas de classificação para essas janelas.

### 4.1.3 Features Geradas com Tratamento 2

O Tratamento 2 foi baseado na padronização global dos dados utilizando o método *Z-score*, conforme apresentado na Seção 2.6.2. Nesse método, a normalização é feita através da média e desvio padrão globais de toda a matriz de *features*, aplicando a mesma transformação para todos os elementos. O objetivo dessa técnica é centralizar a média dos dados em zero e uniformizar a escala das variáveis, sem remover a variação entre elas.

A Figura 15 mostra os resultados após a aplicação desse tratamento. Diferente do Tratamento 1, a padronização *Z-score* preserva melhor as distribuições relativas das *features*, ao mesmo tempo em que reduz possíveis assimetrias causadas por variáveis com maior magnitude.



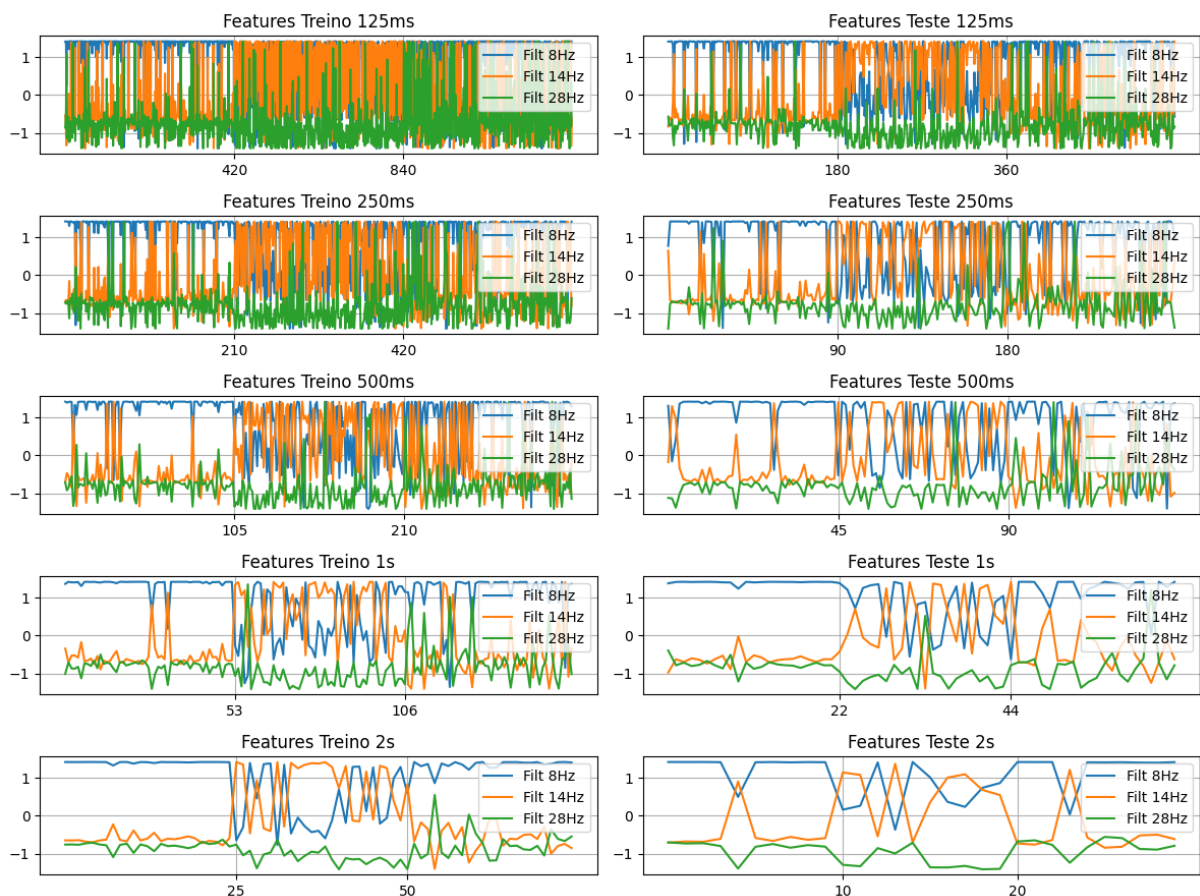
**Figura 15 – Features 8Hz, 14Hz e 28Hz após Tratamento 2. Fonte: O autor.**

Semelhante ao tratamento anterior, os dados das janelas de 1 s e 2 s possuem uma maior separação entre os agrupamentos de diferentes frequências, tanto nos dados de treino quanto de teste. O que reforça uma possível melhora nos resultados para essas classes, em contraste com classes de janelas mais curtas.

#### 4.1.4 Features Geradas com Tratamento 3

O Tratamento 3 foi aplicado por meio de uma variação do método *Z-score*, em que a média e o desvio padrão são calculados individualmente para cada coluna da matriz de *features*, conforme descrito na Seção 2.6.3. O objetivo desse método é realizar a normalização independente de cada *feature*, de forma semelhante à abordagem global adotada no Tratamento 2, mas com foco em preservar as particularidades de cada variável.

A Figura 16 apresenta os resultados obtidos após a aplicação deste tratamento. Observe-se uma redução significativa da variabilidade geral, com forte homogeneização das *features*, o que pode indicar uma menor capacidade dos algoritmos de classificação em distinguir entre as diferentes classes.



**Figura 16 – Features 8Hz, 14Hz e 28Hz após Tratamento 3. Fonte: O autor.**

De modo geral, o tratamento 3 se apresenta como o pior dos métodos propostos, no que diz respeito a representação dos dados, o que será observado posteriormente no resultado da sua entrega aos algoritmos de classificação que serão aplicados na seção seguinte.

#### 4.1.5 Features Geradas com Tratamento 4

O Tratamento 4 consistiu na aplicação da normalização do tipo *Z-score* por linha na matriz de *features*, conforme descrito na Seção 2.6.4. Diferentemente das abordagens anteriores, neste caso a média e o desvio padrão são calculados individualmente para cada linha da matriz, ou seja, para cada amostra. Com isso, normaliza-se cada vetor de características de forma independente, permitindo uma padronização da distribuição de valores ao longo das *features* dentro de cada amostra.

A Figura 17 ilustra os gráficos das *features* extraídas após a aplicação do Tratamento 4, organizadas por diferentes janelas temporais: 125 ms, 250 ms, 500 ms, 1 s e 2 s. Para cada uma dessas janelas, são apresentados os valores das *features* nos conjuntos de treinamento e teste, para as três frequências de estímulo (8 Hz, 14 Hz e 28 Hz).



**Figura 17 – Features 8Hz, 14Hz e 28Hz após Tratamento 4. Fonte: O autor.**

Após a padronização interna das amostras, os padrões temporais característicos de cada frequência tornam-se evidentes por meio das amplitudes associadas a cada janela, especialmente nas janelas mais longas (1 s e 2 s). Esse comportamento sugere uma possível melhora no desempenho dos classificadores na distinção entre as diferentes classes de sinais.

O impacto efetivo desse tratamento será analisado nas seções seguintes, quando os dados normalizados forem aplicados aos algoritmos de classificação.

## 4.2 Desempenho de Classificação sem Tratamento

A seguir são apresentados os resultados obtidos através do uso dos dados sem tratamento para classificação dos dados. Cada resultado foi obtido através da aplicação dos dados aos classificadores, seguido do processo de otimização bayesiana. Os melhores resultados de acurácia e precisão foram então compilados na Tabela 4.

**Tabela 4 – Métricas de Classificação sem Tratamento**

Acurácia						
Época	125 ms	250 ms	500 ms	1s	2s	Média
<b>LDA</b>	67,04 %	68,52 %	74,81 %	80,30 %	100,00 %	78,13 %
<b>KNN</b>	66,30 %	69,26 %	73,33 %	80,30 %	93,33 %	76,50 %
<b>RF</b>	66,67 %	69,26 %	74,07 %	78,79 %	93,33 %	76,42 %
Precisão						
Época	125 ms	250 ms	500 ms	1s	2s	Média
<b>LDA</b>	71,29 %	73,27 %	77,54 %	81,33 %	100,00 %	80,69 %
<b>KNN</b>	68,88 %	69,88 %	75,38 %	81,17 %	93,33 %	77,73 %
<b>RF</b>	67,36 %	69,50 %	75,16 %	79,55 %	94,44 %	77,20 %

Fonte: O autor.

Com base nos dados apresentados, o classificador LDA se destaca, em média, sobre os demais nas duas métricas analisadas, com uma média de 78,13% contra 76,50% e 76,42% dos classificadores KNN e *Random Forest* (RF). Em especial, o LDA se destaca para a janela de 2 s, apresentando 100% para as duas métricas.

A Tabela 5 apresenta os hiperparâmetros otimizados para cada um dos classificadores.

**Tabela 5 – Hiperparâmetros otimizados sem Tratamento**

Classificador	LDA			KNN		
	n_components	shrinkage	solver	n_neighbors	p	weights
125 ms	2	None	eigen	50	1	uniform
250 ms	1	None	eigen	36	1	uniform
500 ms	1	auto	eigen	26	2	uniform
1 s	2	auto	eigen	14	1	distance
2s	2	None	eigen	11	1	distance
Classificador	RANDOM FOREST					
Época	bootstrap	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
125 ms	True	46	'sqrt'	9	5	163
250 ms	True	46	'sqrt'	9	5	163
500 ms	True	46	'sqrt'	9	5	163
1 s	True	46	'sqrt'	9	5	163
2s	True	38	None	7	14	154

Fonte: O autor.

### 4.3 Desempenho de Classificação com Tratamento 1

Nessa seção são apresentados os resultados obtidos através do uso dos dados com tratamento 1 para classificação dos dados. Os melhores resultados de acurácia e precisão foram então compilados na Tabela 6.

**Tabela 6 – Métricas de Classificação com Tratamento 1**

Acurácia						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	62,41 %	64,81 %	70,37 %	74,24 %	93,33 %	73,03 %
KNN	61,85 %	63,70 %	70,37 %	77,27 %	86,67 %	71,97 %
RF	61,30 %	63,33 %	70,37 %	74,24 %	90,00 %	71,85 %
Precisão						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	62,49 %	65,79 %	69,86 %	76,00 %	94,44 %	73,72 %
KNN	61,60 %	63,25 %	71,01 %	77,79 %	87,96 %	72,32 %
RF	60,81 %	63,25 %	70,33 %	74,07 %	91,11 %	71,91 %

Fonte: O autor.

De forma semelhante aos dados sem tratamento, o LDA se destaca como o melhor classificador para as duas métricas. É observável também, uma piora nos resultados para as duas métricas em todos os classificadores, com uma redução média de 4,73% na acurácia e 5,89% na precisão para os 3 classificadores.

A Tabela 7 apresenta os hiperparâmetros otimizados para cada um dos classificadores.

**Tabela 7 – Hiperparâmetros otimizados com Tratamento 1**

Classificador	LDA			KNN		
	n_components	shrinkage	solver	n_neighbors	p	weights
125 ms	1	auto	eigen	50	1	uniform
250 ms	1	auto	eigen	33	2	uniform
500 ms	1	auto	eigen	19	1	uniform
1 s	1	auto	eigen	15	2	uniform
2s	1	auto	eigen	14	2	distance
Classificador	RANDOM FOREST					
Época	bootstrap	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
125 ms	True	46	'sqrt'	9	5	163
250 ms	False	46	'sqrt'	19	18	66
500 ms	False	45	'sqrt'	19	18	66
1 s	True	45	'sqrt'	9	5	163
2s	True	38	None	7	14	154

Fonte: O autor.

## 4.4 Desempenho de Classificação com Tratamento 2

No tópico 4.4 são apresentados os resultados obtidos ao entregar os dados com o tratamento 2 para os classificadores. Os melhores resultados são compilados na Tabela 8.

**Tabela 8 – Métricas de Classificação com Tratamento 2**

Acurácia						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	67,96 %	71,48 %	73,33 %	81,82 %	93,33 %	77,58 %
KNN	66,11 %	69,63 %	73,33 %	78,79 %	90,00 %	75,57 %
RF	65,37 %	70,00 %	74,81 %	78,79 %	93,33 %	76,46 %
Precisão						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	71,50 %	75,96 %	74,03 %	85,54 %	93,33 %	80,07 %
KNN	68,18 %	70,70 %	74,76 %	82,40 %	92,31 %	77,67 %
RF	66,94 %	70,77 %	75,81 %	81,15 %	94,44 %	77,82 %

Fonte: O autor.

Os resultados apresentados para o tratamento 2 se mostraram melhores do que os obtidos para os dados com o tratamento 1. No entanto, quando comparados aos dados do sem tratamento, os resultados foram 0,48% menores para acurácia e 0,14% maiores para precisão, o que indica uma pequena melhoria média para essa métrica com este método de tratamento.

A Tabela 9 apresenta os hiperparâmetros otimizados para cada um dos classificadores.

**Tabela 9 – Hiperparâmetros otimizados com Tratamento 2**

Classificador	LDA			KNN		
	n_components	shrinkage	solver	n_neighbors	p	weights
125 ms	1	auto	eigen	37	2	uniform
250 ms	1	auto	eigen	36	1	uniform
500 ms	1	auto	eigen	48	2	distance
1 s	2	None	eigen	38	1	uniform
2s	1	auto	eigen	23	2	uniform
Classificador	RANDOM FOREST					
Época	bootstrap	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
125 ms	False	40	'log2'	18	13	179
250 ms	True	46	'sqrt'	9	5	137
500 ms	False	46	'log2'	17	8	16
1 s	False	46	'log2'	17	8	16
2s	False	25	'log2'	15	18	215

Fonte: O autor.

## 4.5 Desempenho de Classificação com Tratamento 3

No tópico 4.5 são apresentados os resultados obtidos ao entregar os dados com o tratamento 3 para os classificadores. Os melhores resultados são compilados na Tabela 10.

Tabela 10 – Métricas de Classificação com Tratamento 3

Acurácia						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	60,37 %	58,89 %	60,00 %	65,15 %	73,33 %	63,55 %
KNN	60,74 %	60,74 %	68,15 %	77,27 %	86,67 %	70,71 %
RF	58,70 %	60,37 %	66,67 %	77,27 %	86,67 %	69,94 %
Precisão						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	64,52 %	64,61 %	64,49 %	77,42 %	76,39 %	69,49 %
KNN	60,12 %	60,21 %	67,93 %	78,82 %	90,48 %	71,51 %
RF	58,17 %	59,81 %	66,98 %	78,97 %	87,27 %	70,24 %

Fonte: O autor.

Os dados submetidos ao Tratamento 3 apresentaram os piores desempenhos entre todos os conjuntos analisados nesta pesquisa, resultando nas menores médias de acurácia e precisão para os três classificadores avaliados. Em comparação aos dados sem tratamento, a acurácia foi, em média, 13,47% inferior, enquanto a precisão apresentou uma redução média de 8,13%.

A Tabela 11 apresenta os hiperparâmetros otimizados para cada um dos classificadores.

Tabela 11 – Hiperparâmetros otimizados com Tratamento 3

Classificador	LDA			KNN		
	n_components	shrinkage	solver	n_neighbors	p	weights
125 ms	1	auto	eigen	47	2	uniform
250 ms	1	auto	eigen	42	2	uniform
500 ms	1	auto	eigen	40	1	uniform
1 s	1	auto	eigen	16	2	uniform
2s	1	auto	eigen	8	2	distance
Classificador	RANDOM FOREST					
Época	bootstrap	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
125 ms	False	45	'sqrt'	19	18	20
250 ms	False	45	'sqrt'	19	18	20
500 ms	False	45	'sqrt'	19	18	20
1 s	False	45	'sqrt'	19	18	20
2s	False	47	'sqrt'	5	16	112

Fonte: O autor.

## 4.6 Desempenho de Classificação com Tratamento 4

No tópico 4.6 são apresentados os resultados obtidos ao entregar os dados com o tratamento 4 para os classificadores. Os melhores resultados são compilados na Tabela 12.

Tabela 12 – Métricas de Classificação com Tratamento 4

Acurácia						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	66,48 %	70,37 %	74,07 %	83,33 %	93,33 %	77,52 %
KNN	67,04 %	68,52 %	71,85 %	78,79 %	93,33 %	75,91 %
RF	67,22 %	71,11 %	74,81 %	78,79 %	93,33 %	77,05 %
Precisão						
Época	125 ms	250 ms	500 ms	1s	2s	Média
LDA	69,82 %	74,09 %	77,35 %	83,95 %	93,64 %	79,77 %
KNN	68,82 %	70,24 %	74,95 %	81,25 %	94,44 %	77,94 %
RF	67,91 %	71,22 %	76,30 %	81,54 %	94,44 %	78,28 %

Fonte: O autor.

Os resultados do tratamento 4 apresentaram o segundo melhor desempenho médio de acurácia e o melhor desempenho de precisão entre todos os dados. A acurácia média ficou 0,19% abaixo da apresentada pelos dados sem tratamento, enquanto a precisão ficou 0,12% acima. Os resultados sugerem que a normalização *z-score* das amostras pode influenciar positivamente a precisão dos modelos de classificação.

A Tabela 13 apresenta os hiperparâmetros otimizados para cada um dos classificadores.

Tabela 13 – Hiperparâmetros otimizados com Tratamento 4

Classificador	LDA			KNN		
	n_components	shrinkage	solver	n_neighbors	p	weights
125 ms	2	None	eigen	42	2	uniform
250 ms	2	None	eigen	50	1	distance
500 ms	1	auto	eigen	46	1	uniform
1 s	1	auto	eigen	27	1	uniform
2s	1	auto	eigen	15	1	uniform
Classificador	RANDOM FOREST					
Época	bootstrap	max_depth	max_features	min_samples_leaf	min_samples_split	n_estimators
125 ms	True	46	'sqrt'	9	5	163
250 ms	False	46	'sqrt'	19	18	66
500 ms	False	45	'sqrt'	19	18	66
1 s	True	45	'sqrt'	9	5	163
2s	True	38	None	7	14	154

Fonte: O autor

## 4.7 Comparação dos desempenhos

A Seção 4.7 apresenta uma visão consolidada do desempenho dos classificadores, com o objetivo de viabilizar uma comparação abrangente entre os resultados obtidos ao longo do estudo.

Tabela 14 – Tabela Comparativo Geral Classificadores

Acurácia							
Época	Tratamento	125 ms	250ms	500ms	1s	2s	Média
LDA	-	67,04 %	68,52 %	74,81 %	80,30 %	100,00 %	78,13 %
	1	62,41 %	64,81 %	70,37 %	74,24 %	93,33 %	73,03 %
	2	67,96 %	71,48 %	73,33 %	81,82 %	93,33 %	77,58 %
	3	60,37 %	58,89 %	60,00 %	65,15 %	73,33 %	63,55 %
	4	66,48 %	70,37 %	74,07 %	83,33 %	93,33 %	77,52 %
KNN	-	66,30 %	69,26 %	73,33 %	80,30 %	93,33 %	76,50 %
	1	61,85 %	63,70 %	70,37 %	77,27 %	86,67 %	71,97 %
	2	66,11 %	69,63 %	73,33 %	78,79 %	90,00 %	75,57 %
	3	60,74 %	60,74 %	68,15 %	77,27 %	86,67 %	70,71 %
	4	67,04 %	68,52 %	71,85 %	78,79 %	93,33 %	75,91 %
RF	-	66,67 %	69,26 %	74,07 %	78,79 %	93,33 %	76,42 %
	1	61,30 %	63,33 %	70,37 %	74,24 %	90,00 %	71,85 %
	2	65,37 %	70,00 %	74,81 %	78,79 %	93,33 %	76,46 %
	3	58,70 %	60,37 %	66,67 %	77,27 %	86,67 %	69,94 %
	4	67,22 %	71,11 %	74,81 %	78,79 %	93,33 %	77,05 %
Precisão							
Época	Tratamento	125 ms	250ms	500ms	1s	2s	Média
LDA	-	71,29 %	73,27 %	77,54 %	81,33 %	100,00 %	80,69 %
	1	62,49 %	65,79 %	69,86 %	76,00 %	94,44 %	73,72 %
	2	71,50 %	75,96 %	74,03 %	85,54 %	93,33 %	80,07 %
	3	64,52 %	64,61 %	64,49 %	77,42 %	76,39 %	69,49 %
	4	69,82 %	74,09 %	77,35 %	83,95 %	93,64 %	79,77 %
KNN	-	68,88 %	69,88 %	75,38 %	81,17 %	93,33 %	77,73 %
	1	61,60 %	63,25 %	71,01 %	77,79 %	87,96 %	72,32 %
	2	68,18 %	70,70 %	74,76 %	82,40 %	92,31 %	77,67 %
	3	60,12 %	60,21 %	67,93 %	78,82 %	90,48 %	71,51 %
	4	68,82 %	70,24 %	74,95 %	81,25 %	94,44 %	77,94 %
RF	-	67,36 %	69,50 %	75,16 %	79,55 %	94,44 %	77,20 %
	1	60,81 %	63,25 %	70,33 %	74,07 %	91,11 %	71,91 %
	2	66,94 %	70,77 %	75,81 %	81,15 %	94,44 %	77,82 %
	3	58,17 %	59,81 %	66,98 %	78,97 %	87,27 %	70,24 %
	4	67,91 %	71,22 %	76,30 %	81,54 %	94,44 %	78,28 %

Fonte: O autor.

## 5 CONCLUSÕES

Este trabalho teve como objetivo avaliar o desempenho de classificadores multiclasse aplicados a um problema de análise de sinais de EEG, utilizando o paradigma de estímulo visual SSVEP. Foram analisadas três frequências de estímulo (8 Hz, 14 Hz e 28 Hz), extraídas de um banco de dados público fornecido pelo RIKEN-LABSP, com diferentes janelas temporais e técnicas de pré-processamento. Os classificadores utilizados foram LDA, KNN e *Random Forest*, combinados a quatro estratégias distintas de tratamento das *features* extraídas por periodogramas.

Verificou-se, através dos resultados aqui obtidos, que janelas temporais maiores (1s e 2s) proporcionaram, em média, melhores resultados de acurácia e precisão em comparação às janelas mais curtas (125 ms, 250 ms e 500 ms). Em destaque, o classificador LDA, utilizando os dados brutos segmentados em janelas de 2 segundos, atingiu uma acurácia de 100%, valor expressivo frente às demais configurações avaliadas.

Observou-se também que o classificador LDA superou os classificadores KNN e *Random Forest*, apresentando acurácia média máxima de 78,13%, frente a 76,50% e 77,05%, respectivamente. O desempenho superior do LDA pode ser atribuído à natureza linearmente separável das características extraídas dos sinais SSVEP, bem como à robustez do algoritmo frente a conjuntos com número limitado de amostras por classe. Diferente de métodos como KNN, que sofrem com alta dimensionalidade, e *Random Forest*, que pode superajustar em dados escassos, o LDA mostrou-se mais adequado às condições do experimento, alcançando alta acurácia com baixa complexidade computacional.

Quanto ao tratamento dos dados, observou-se que as técnicas de normalização por *Z-score* não proporcionaram melhorias significativas no desempenho dos classificadores. Os melhores resultados, tanto para o LDA quanto para o KNN, foram obtidos com dados não normalizados, sugerindo que os métodos de pré-processamento avaliados podem não ser os mais adequados neste contexto. Em particular, no que se refere ao "Tratamento 3", os dados apresentaram menor variabilidade entre amostras, conforme ilustrado nos gráficos da Seção 4.1.

A análise visual dos resultados revela uma perda considerável na separabilidade dos dados submetidos aos tratamentos de normalização e de normalização por *Z-score*, o que pode justificar a redução no desempenho de classificação para os dados tratados quando comparados aos dados sem tratamento. Esse comportamento sugere que as técnicas aplicadas atenuaram ou eliminaram características distintivas dos dados, comprometendo a capacidade dos classificadores de diferenciá-los de forma eficaz.

Além disso, considerando a maior separabilidade dos dados e as características inerentes ao sinal de SSVEP, pode-se deduzir que o desempenho superior do classificador LDA está relacionado ao seu aproveitamento da separabilidade linear presente nos dados para realizar a classificação.

Conclui-se, portanto, que a abordagem multiclasse aplicada a sinais de EEG do tipo SSVEP demonstrou-se eficaz, especialmente quando utilizadas janelas temporais mais longas. Os melhores desempenhos registrados foram de 100% com janelas de 2 segundos, 83,33% com janelas de 1 segundo e 74,81% com janelas de 500 ms, evidenciando o impacto da segmentação temporal na acurácia do sistema. Os resultados obtidos confirmam a viabilidade do uso de classificadores multiclasse para o problema proposto, sendo essa abordagem particularmente promissora para janelas maiores.

## 6 SUGESTÕES PARA TRABALHOS FUTUROS

Com base no estudo desenvolvido nesta monografia, sugere-se o desenvolvimento de trabalhos futuros, a fim de avaliar possíveis melhorias na eficiência dos sistemas e nas métricas de avaliação.

- Análise de sobreposição de épocas: Técnica que pode ser utilizada em trabalhos futuros para criar uma maior variabilidade nos conjuntos de *features*;
- Variação nos algoritmos de classificação: Outros algoritmos de classificação multiclasse como o Perceptron de Múltiplas Camadas e o SVM podem ser estudados e empregados para análise dos dados utilizados neste estudo;
- Conjuntos de Dados: Os métodos e algoritmos utilizados nesse trabalho podem ser aplicados a novos conjuntos de dados, a fim de avaliar sua eficiência em outros cenários;
- Novas Métricas de Avaliação: O uso de outras métricas de avaliação e comparação podem ser utilizadas tanto no desenvolvimento do trabalho como na etapa de Otimização Bayesiana.

## REFERÊNCIAS

- ACUNZO, D. J.; MACKENZIE, G.; van Rossum, M. C. Systematic biases in early erp and erf components as a result of high-pass filtering. **Journal of Neuroscience Methods**, v. 209, n. 1, p. 212–218, 2012. ISSN 0165-0270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0165027012002361>>.
- AMBIENTAL, S. E. e C. **Avaliação de Acurácia ou Concordância**. 2020. Acessado em 5 ago. 2025. Disponível em: <<https://solved.eco.br/avaliacao-de-acuracia-ou-concordancia>>.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.
- BROCHU, E.; CORA, V. M.; FREITAS, N. de. **A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning**. 2010. <<https://arxiv.org/pdf/1012.2599.pdf>>. ArXiv:1012.2599.
- BUENO, L. **Classificação de sinais de EEG para interfaces cérebro-computador utilizando aprendizagem de máquina**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Espírito Santo, Vitória, ES, 2019. Disponível em: <[https://sappg.ufes.br/tese\\_drupal/tese\\_11033\\_Leandro%20Bueno.pdf](https://sappg.ufes.br/tese_drupal/tese_11033_Leandro%20Bueno.pdf)>.
- CALVINHO, J. V. F. **ICM Space Game: uma interface baseada na imaginação de movimentos**. 64 p. Dissertação (Dissertação (Mestrado em Computação Aplicada)) — Universidade Federal do Pará, Campus Tucuruí, Tucuruí, PA, Brasil, mar. 2023. Disponível em: Repositório Institucional UFPA.
- CHOWDHURY, M. S. N. et al. Deep neural network for visual stimulus-based reaction time estimation using the periodogram of single-trial eeg. **Sensors**, v. 20, n. 21, p. 6090, 2020.
- DEMIR, A. F.; ARSLAN, H.; UYSAL, I. Bio-inspired filter banks for frequency recognition of ssvep-based brain–computer interfaces. **IEEE Access**, v. 7, p. 160295–160303, 11 2019.
- DIAS, F. K. N. **Interface Cérebro-Máquina Baseada em Potenciais Visualmente Evocados: Análise de Extração de Épocas**. 2023. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Universidade Federal do Pará, Campus Universitário de Tucuruí. Orientador: Prof. Dr. Cleison Daniel Silva.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2nd. ed. New York: Wiley-Interscience, 2000. ISBN 978-0-471-05669-0.
- GARCIA-MOLINA, G.; ZHU, D. Optimal spatial filtering for the steady state visual evoked potential: Bci application. In: **Proceedings of the 5th International IEEE EMBS Conference on Neural Engineering**. Cancun, Mexico: IEEE, 2011. p. 156–160.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. [S.l.]: O'Reilly Media, 2019.
- JORGE, V. da S. **ANÁLISE DA UTILIZAÇÃO DE INTERFACES-CÉREBRO MÁQUINA EM PARALELO PARA A CLASSIFICAÇÃO DA IMAGÉTICA MOTORA**. 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Federal do Pará, Campus Universitário de Tucuruí. Orientador: Prof. Dr. Cleison Daniel Silva.

- JORGE, V. da S. **Aprendizado em Conjunto Aplicado à Classificação da Imagética Motora**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Pará, Programa de Pós-Graduação em Computação Aplicada, Tucuruí, PA, Brasil, 2024. Orientador: Dr. Cleison Daniel Silva.
- KANDEL, E. R. et al. **Princípios de Neurociência**. 5. ed. Porto Alegre: AMGH Editora, 2014.
- KANG, J.; AL. et. Adaptive eeg noise filtering for coherence analysis. In: **Proceedings of the 2014 International Winter Workshop on Brain–Computer Interface (BCI)**. Gangwon, Republic of Korea: [s.n.], 2014. p. 1–4.
- LEITE, S. N. d. C. et al. Comparative analysis of strategies for feature extraction and classification in ssvep bcis. **Biomedical Signal Processing and Control**, Elsevier, v. 21, p. 34–42, 2015. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1746809415000877>>.
- LEVADA, P. A. **Análise Discriminante Múltipla (MDA)**. 2021. <<https://www.youtube.com/watch?v=Qb1-ra5zsdY>>. Vídeo publicado em 26 de março de 2021. Acessado em 1 ago. 2025.
- LOPES, T. de J. **Estudo, Implementação e Comparação de Diferentes Tipos de Pré-Processamento Aplicados a um Sistema Interface Cérebro-Computador**. Dissertação (Trabalho de Conclusão de Curso (Engenharia Elétrica)) — Universidade Federal do Pará, Tucuruí, PA, Brasil, 2019. Orientador: Prof. Dr. Cleison Daniel Silva.
- LOTTE, F.; CONGEDO, M. Eeg feature extraction. **Brain–Computer Interfaces 1: Foundations and Methods**, Wiley Online Library, p. 127–143, 2016.
- LOTTE, F. et al. A review of classification algorithms for eeg-based brain–computer interfaces. **Journal of Neural Engineering**, v. 4, n. 2, p. R1, jan 2007. Disponível em: <<https://dx.doi.org/10.1088/1741-2560/4/2/R01>>.
- MARTIN, J. H. **Neuroanatomia: Texto e Atlas**. 4. ed. Porto Alegre: McGraw Hill Education, 2013.
- MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw Hill, 1997.
- MITRA, P.; BIESSMANN, F. **Automated Computational Energy Minimization of ML Algorithms using Constrained Bayesian Optimization**. 2024. Disponível em: <<https://arxiv.org/abs/2407.05788>>.
- MLADENOVA, T.; VALOVA, I. Comparative analysis between the traditional k nearest neighbor and modifications with weight-calculation. In: **2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)**. [S.l.: s.n.], 2022. p. 961–965.
- NIEDERMEYER, E.; SILVA, F. H. Lopes da. **Electroencephalography: Basic Principles, Clinical Applications, and Related Fields**. 5. ed. Philadelphia: Lippincott Williams & Wilkins, 2005.
- ROCHA, J. de Caldas Simões Neto e Francisco Marcelo de Oliveira e Pergentina Catunda e Lucielton Martins e A. Processos de plasticidades na aprendizagem do cérebro humano e sua relação com atividade física. **Revista MotriSaúde**, v. 1, n. 1, 2019. ISSN 2674-7782. Disponível em: <[http://revista.fundacaojau.edu.br:8078/journal%20-%20Copia/index.php/revista\\_motrisaude/article/view/51](http://revista.fundacaojau.edu.br:8078/journal%20-%20Copia/index.php/revista_motrisaude/article/view/51)>.

SENA, M. L. d. R. **Desenvolvimento de um sistema BCI-SSVEP e testes preliminares no controle de um protótipo de cadeira de rodas**. João Monlevade, MG, Brasil, 2018. 40 p. Disponível no repositório digital da UFOP.

SILVERTHORN, D. U. **Fisiologia Humana: Uma Abordagem Integrada**. 7. ed. Porto Alegre: Artmed, 2017. ISBN 9788582714161.

TAN, D. S.; NIJHOLT, A. Brain–computer interfaces and human–computer interaction. In: **Brain–Computer Interfaces: Applying our Minds to Human–Computer Interaction**. London: Springer, 2010, (Human–Computer Interaction Series). p. 3–19.

VILAS-BOAS, V.; SILVA, C. D. Configuração automática de hiperparâmetros em sistemas bci-mi de sub-bandas. In: **Biblioteca do XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2020)**. Florianópolis, SC, Brasil: SBRT, 2020. p. 5. Disponível em: <<https://biblioteca.sbrt.org.br/articles/2245>>.

## **Anexos**

## **ANEXO A – LINK DE ACESSO AO REPOSITÓRIO DO GITHUB**

**Repositório:** <https://github.com/JoaoLucas976/TCC>