

UNIVERSIDADE FEDERAL DO PARÁ NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

VITOR DA SILVA JORGE

APRENDIZADO EM CONJUNTO APLICADO À CLASSIFICAÇÃO DA IMAGÉTICA MOTORA

Tucuruí 2024



UNIVERSIDADE FEDERAL DO PARÁ NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

VITOR DA SILVA JORGE

APRENDIZADO EM CONJUNTO APLICADO À CLASSIFICAÇÃO DA IMAGÉTICA MOTORA

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Núcleo de desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para a obtenção do título de Mestre em Computação Aplicada.

Área de Concentração: Computação Aplicada Orientador: Dr. Cleison Daniel Silva

Tucuruí 2024

BANCA EXAMINADORA



Prof. Dr. Cleison Daniel Silva Orientador (PPCA/NDAE/UFPA)



Prof. Dr. Edson Koiti Kudo Yasojima *Examinador Externo (UFRA)*



Prof. Dr. Caio Carvalho Moreira *Examinador Interno (PPCA/NDAE/UFPA)*



Prof. Dr. Otávio Noura Teixeira *Examinador Interno (PPCA/NDAE/UFPA)*

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)

 D111a da Silva Jorge, Vitor. Aprendizado em Conjunto Aplicado à Classificação da Imagética Motora / Vitor da Silva Jorge. — 2025. 78 f. : il. color.

> Orientador(a): Prof. Dr. Cleison Daniel Silva Dissertação (Mestrado) - Universidade Federal do Pará, Campus Universitário de Tucuruí, Programa de Pós-Graduação em Engenharia de Barragem e Gestão Ambiental, Tucuruí, 2025.

1. Interface Cérebro Máquina. 2. Imagética Motora. 3. Aprendizado em Conjunto. 4. Otimização. 5. Eletroencefalografia. I. Título.

CDD 006.31

LISTA DE ILUSTRAÇÕES

Figura 1 –	Cadeia de processamento que define uma interface cérebro-máquina tradicional.	10
Figura 2 –	Aquisição dos sinais de EEG	11
Figura 3 –	Sistema 10-20 de posicionamento de eletrodos para aquisição dos sinais de	
	EEG. Vistas lateral e superior.	11
Figura 4 –	Córtices Sensório-Motores.	12
Figura 5 –	Dessincronização e Sincronização Relacionada a Evento observadas no espectro do	
	EEG filtrado nas bandas que compreendem os ritmos mu (8-12Hz) e beta (14-30Hz)	
	ao longo da execução do movimento do braço direito. O sinal é aquisitado pelos	
	eletrodos C3 e C4 posicionados no escalpo do indivíduo	13
Figura 6 –	Extração de épocas realizada com o auxílio do protocolo experimental apli-	
	cado à coleta de sinais de EEG. $t_{min} = 0,5$ e $t_{max} = 2,5$ denotam o início e o	
	final da janela, respectivamente	14
Figura 7 –	Exemplo de Classificação da Imagética Motora das Mãos.	19
Figura 8 –	Um hiperplano separando as classes "cruzes" e "círculos"	20
Figura 9 –	Padrões linearmente separáveis e hiperplanos capazes de efetuar a separação	
	entre as classes. O hiperplano g_2 em b) proporciona maior margem de separação.	22
Figura 10 -	- Representação do hiperplano de separação para padrões não linearmente	
	separáveis	23
Figura 11 -	- Exemplo de mapeamento dos padrões para um espaço de maior dimensão	
	que proporciona separação linear.	24
Figura 12 -	- Modelo básico de um Perceptron	24
Figura 13 -	- Perceptron Multicamadas	25
Figura 14 -	- Fluxo da otimização bayesiana.	28
Figura 15 -	- Exemplo de votação majoritária.	29
Figura 16 -	- Exemplo de soft voting	30
Figura 17 -	- Etapas do processo de stacking.	31
Figura 18 -	- Etapas do processo de bagging.	32
Figura 19 -	- Matriz de confusão.	33
Figura 20 -	- Etapas do desenvolvimento da metodologia.	38
Figura 21 -	- Disposição dos Canais de Eletrodos do Banco de dados 2a da IV competição de ICM.	39
Figura 22 -	- Ajuste de uma interface cérebro-máquina.	40
Figura 23 -	- Diagrama funcional do processo de ajuste do modelo de classificação a partir da	
	plataforma <i>AutoBCI</i>	42
Figura 24 -	- Os três modelos de ICM utilizados selecionados para a classificação das classes ME	
	e MD para o sujeito 1	43
Figura 25 -	- Esquema de aplicação do algoritmo <i>Bagging</i> para a classificação binária	45
Figura 26 -	- Esquema de aplicação do algoritmo <i>bagging</i> para a classificação multi-classe	45

Figura 27 – Esquema de aplicação do algoritmo stacking para a classificação multi-classe. . . . 46

LISTA DE TABELAS

Tabela 1 Hiperparâmetros configuráveis em cada etapa da ICM.	41
Tabela 2 – Hiperparâmetros configuráveis em cada classificador. . <t< td=""><td>41</td></t<>	41
Tabela 3 – Resultados da aplicação do Bagging para as classes ME x MD.	49
Tabela 4 – Resultados da aplicação do Bagging para as classes ME x PE.	50
Tabela 5 – Resultados da aplicação do Bagging para as classes ME x LG.	51
Tabela 6 – Resultados da aplicação do <i>Bagging</i> para as classes MD x PE.	52
Tabela 7 – Resultados da aplicação do Bagging para as classes MD x LG.	53
Tabela 8 – Resultados da aplicação do Bagging para as classes PE x LG.	54
Tabela 9 – Acurácia de classificação e índice kappa dos modelos Stacking e Bagging	
Multi-classe	55
Tabela 10 – Desempenho de precisão do modelo bagging multi-classe.	56
Tabela 11 – Desempenho de precisão do modelo Stacking multi-classe.	56
Tabela 12 – Comparação do desempenho de acurácia (%) entre os modelos de aprendi-	
zado em conjunto propostos e os resultados dos artigos selecionados	57
Tabela 13 – Configuração de hiperparâmetros de pré-processamento para cada modelo de	
ICM por sujeito para o par ME x MD.	65
Tabela 14 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par ME x MD.	66
Tabela 15 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par ME x PE	67
Tabela 16 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par ME x PE	68
Tabela 17 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par ME x LG	69
Tabela 18 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par ME x LG	70
Tabela 19 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par MD x PE	71
Tabela 20 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par MD x PE	72
Tabela 21 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par MD x LG.	73
Tabela 22 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par MD x LG.	74
Tabela 23 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par PE x LG	75
Tabela 24 – Configuração de hiperparâmetros do classificador para cada modelo de ICM	
por sujeito para o par PE x LG	76

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	9
1.1.1	Objetivo Geral	9
1.1.2	Objetivos Específicos	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Interface Cérebro-Máquina (ICM)	10
2.1.1	Aquisição de Sinais de EEG	10
2.1.2	Imagética Motora	11
2.1.3	Extração de Épocas dos Sinais de EEG	13
2.1.4	Pré-processamento dos Sinais de EEG	14
2.1.4.0.1	Filtragem Temporal	14
2.1.4.0.2	Filtragem Espacial	15
2.1.5	Extração e Seleção de Características	16
2.1.6	Algoritmos de Classificação	17
2.1.6.0.1	Análise Discriminante Linear (LDA)	19
2.1.6.0.2	Regressão Logística (LR)	21
2.1.6.0.3	Máquina de Vetores de Suporte (SVM)	21
2.1.6.0.4	Perceptron Multicamadas (MLP)	24
2.1.6.0.5	K-Vizinhos mais Próximos (KNN)	26
2.2	Otimização Bayesiana	26
2.3	Aprendizado em Conjunto	28
2.3.1	Votação por Maioria e Soft Voting	29
2.3.2	Stacking	30
2.3.3	Bagging	31
2.4	Métricas de Avaliação	32
2.5	Trabalhos Relacionados	35
3	METODOLOGIA	38
3.1	Ambiente de Execução dos Experimentos e Descrição da Bases de Dados	38
3.2	Ajuste dos Modelos de Interface Cérebro-Máquina	40
3.3	Aplicação dos Algoritmos de Aprendizado em Conjunto	43
3.3.1	Aplicação do <i>Bagging</i> - Abordagem binária	44
3.3.2	Aplicação do <i>Bagging</i> - Abordagem multi-classe	44
3.3.3	Aplicação do Stacking - Abordagem multi-classe	46
3.4	Avaliação dos modelos propostos	47
4	RESULTADOS E DISCUSSÕES	48
4.1	Resultados da aplicação do <i>Bagging -</i> Abordagem Binária	48
4.1.1	Bagging para as classes ME x MD	48

Α	MODELOS DE ICMS-IM GERADOS	64
	REFERÊNCIAS	60
5	CONCLUSÕES E PERSPECTIVAS FUTURAS	58
4.2	Resultados da Aplicação do Bagging e Stacking - Abordagem Multi-classe	55
4.1.6	Bagging para as classes PE x LG	54
4.1.5	Bagging para as classes MD x LG	52
4.1.4	Bagging para as classes MD x PE	51
4.1.3	Bagging para as classes ME x LG	50
4.1.2	Bagging para as classes ME x PE	49

1 INTRODUÇÃO

As interfaces cérebro máquina (ICM's) baseadas em imagética motora (IM) são cadeias de processamento capazes de traduzir a intenção motora do usuário em comando para um dispositivo externo ou virtual (LOTTE, 2014). Essas cadeias são ajustadas com pequenos trechos do sinal de eletroencefalografia (EEG), exame capaz de quantificar a atividade cerebral de um indivíduo. Uma ICM-IM busca padrões no sinal de EEG que possam ser relacionados à imaginação de movimentos de membros como mãos, pés e língua. Após ajustada, a interface deve ser capaz de detectar os padrões de cada IM e classificá-los quando presentes no sinal. O sinal de EEG é tênue e sujeito a ruído, além disso, os padrões que definem a IM podem ocorrer a qualquer momento do sinal aquisitado. Portanto, classificar a intenção motora não é uma tarefa

Os trabalhos que visam a melhoria da classificação da IM, em sua maioria, têm se concentrado em otimizar os algoritmos que constituem a estrutura da ICM como filtros, extratores de características e classificadores. Contudo, o ajuste dos algoritmos utilizados que produzem melhores resultados pode variar de sujeito para sujeito (VILAS-BOAS; JORGE; SILVA, 2020). Na pesquisa desenvolvida por (VILAS-BOAS; SILVA, 2020) as ICM's são ajustadas de forma personalizada para cada sujeito, utilizando uma otimização bayesiana. A otimização busca as melhores combinações de parâmetros para diferentes algoritmos da interface. O trabalho evidenciou que combinações distintas podem produzir um desempenho de classificação robusto para um mesmo sujeito. Portanto, é pertinente investigar se é possível combinar essas estruturas ajustadas individualmente para produzir um melhor desempenho de classificação.

A combinação de classificadores é uma vertente do aprendizado de máquina conhecida como aprendizado em conjunto. O objetivo desta técnica é explorar as características de diferentes modelos diante de um problema de classificação. No contexto de ICM's, essa estratégia já foi amplamente explorada para o desenvolvimento de interfaces robustas e independentes de sujeito, como no trabalho de (DOLZHIKOVA et al., 2021). O objetivo desse segmento é combinar os dados de sujeitos distintos e produzir uma interface que seja capaz de classificar com precisão independente do sujeito que a utilizar. Os resultados sugerem que a combinação de classificadores é capaz de produzir modelos com desempenho superior aos modelos tradicionais.

O desenvolvimento de uma ICM personalizada busca explorar as características particulares que a IM de cada sujeito possui para produzir resultados de classificação mais precisos. Enquanto a combinação de modelos ajustados tem como objetivo produzir uma interface com desempenho superior aos modelos individuais. Assim, este trabalho tem como objetivo investigar se a combinação de ICM's ajustadas individualmente é capaz de melhorar o desempenho de classificação da imagética motora, uma vez que tenta explorar o melhor das duas abordagens descritas. O trabalho ainda investiga qual a melhor estratégia de combinar as ICM's ajustadas.

A metodologia proposta nesta pesquisa é aplicada em duas abordagens de classificação:

binária e multi-classe. Na abordagem binária os experimentos estudam a combinação dos modelos de ICM ajustados com Otimização Bayesiana para a distinção de duas classes de imagética motora e duas abordagens são avaliadas para a classificação final: Votação por Maioria e *Soft Voting*. Já a abordagem multi-classe investiga a combinação dos modelos otimizados para a distinção de quatro classes de imagética motora, a estratégia de classificação final utilizada é a *Soft Voting*.

Este trabalho está organizado da seguinte maneira: A seção 2 apresenta uma breve fundamentação teórica a respeito das interfaces cérebro-máquina. Na Seção 3, é apresentada a metodologia do trabalho. Na seção 4 estão apresentados os resultados dos experimentos conduzidos na pesquisa. Na seção 5 são apresentadas as conclusões a respeito dos resultados obtidos no estudo.

1.1 Objetivos

1.1.1 Objetivo Geral

Investigar se a combinação de interfaces cérebro-máquina personalizadas com otimização é capaz de melhorar o desempenho de classificação da imagética motora.

1.1.2 Objetivos Específicos

- Estudar como os algoritmos de aprendizado em conjunto podem ser aplicados à classificação da imagética motora.
- Investigar a aplicação dos algoritmos de aprendizado em conjunto para a classificação binária.
- Investigar a aplicação dos algoritmos de aprendizado em conjunto para a classificação multi-classe.
- Avaliar o desempenho de classificação dos modelos combinados frente aos modelos individuais.
- Realizar uma análise estatística da relevância dos modelos combinados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Interface Cérebro-Máquina (ICM)

Nesta seção são apresentados conceitos básicos para a compreensão e desenvolvi-mento de uma Interface Cérebro-Máquina baseada em Imagética Motora (ICM-IM) tradicional. Após a aquisição dos sinais de eletroencefalografia, pequenos trechos do sinal são entregues para a ICM, que geralmente é composta pelas etapas de extração de épocas, pré-processamento, extração de características e classificação, como mostra a Figura 1.

Figura 1 – Cadeia de processamento que define uma interface cérebro-máquina tradicional.



Fonte: O Autor.

2.1.1 Aquisição de Sinais de EEG

Os eventos que definem a atividade cerebral podem ser quantificados por meio do eletroencefalograma (EEG), procedimento que monitora os campos elétricos ou magnéticos produzidos pela atividade neuronal usando sensores no couro cabeludo, na superfície do cérebro ou dentro do cérebro (LOPES, 2019). As ICM's mais comuns aquisitam os sinais de EEG através de sensores posicionados no escalpo do usuário, como mostrado na Figura 2.

O posicionamento dos eletrodos no escalpo segue o padrão internacional mostrado na Figura 3, conhecido como sistema 10-20. A padronização do posicionamento permite que a aquisição dos sinais de EEG seja melhor registrada e documentada. Segundo o padrão, cada eletrodo recebe uma identificação de acordo com a região do cérebro na qual está posicionado (KUMAR; BHUVANESWARI, 2012). Os índices O, Fp, F, C, T e P representam, respectivamente, as regiões Occipital, Frontal-polar, Frontal, Central, Temporal e Parietal do cérebro. E os números apontam sobre qual hemisfério do cérebro os eletrodos estão posicionados, sendo os números ímpares indicativos do hemisfério esquerdo e os pares para o direito.



Figura 2 – Aquisição dos sinais de EEG.

(TEPLAN, 2002).

Figura 3 – Sistema 10-20 de posicionamento de eletrodos para aquisição dos sinais de EEG. Vistas lateral e superior.



Fonte: (PAPANASTASIOU et al., 2020)

2.1.2 Imagética Motora

O desenvolvimento de uma ICM baseia-se no princípio de que determinadas atividades mentais produzem padrões característicos que podem ser detectados e distinguidos através dos sinais de EEG. Essas atividades mentais são chamadas de paradigmas experimentais (ABIRI et al., 2018). Após detectados, os padrões observados são traduzidos em sinal de controle para dispositivos. Dentre os paradigmas de controle disponíveis para as ICM's, destaca-se a Imagética Motora (IM), que utiliza como informação as mudanças na atividade rítmica cerebral, provocadas pela imaginação do movimento de membros como mãos, pés e língua (LINDIG; BOUGRAIN, 2015).

Os ritmos cerebrais ou ritmos sensório-motores (RSM) são oscilações nos campos elétricos ou magnéticos, registradas por EEG sobre os córtices sensório-motores, ilustrados na Figura 4. Os RSM normalmente se enquadram em três bandas de frequência principais: mu (8–12 Hz), beta (18–30 Hz) e gamma (30–200 Hz). O registro de EEG é amplamente limitado

aos ritmos mu, beta e atividade gamma de baixa frequência (CHEN et al., 2021).

As características de amplitude e frequência de cada ritmo correspondem à realização de diferentes atividades cerebrais. O ritmo gmama, por exemplo, está associado a tarefas como consciência e percepções sensoriais sonoras e visuais. Enquanto que os ritmos mu e beta estão associados às áreas corticais mais diretamente conectadas aos canais de saída motora normal do cérebro (LOTZE; HALSBAND, 2006).





A atividade motora ou preparação para o movimento é tipicamente acompanhada por uma diminuição nos ritmos mu e beta (TANG et al., 2019). Esta diminuição é chamada de dessincronização relacionada ao evento (Event-Related Desynchronization – ERD). O aumento do ritmo ou sincronização relacionada ao evento (Event-Related Synchronization – ERS) ocorre após a execução da atividade motora. Ambos os fenômenos estão ilustrados na Figura 5, para o movimento do braço direito.

As ICM's baseadas em imagética motora se beneficiam da ERD e ERS também ocorrerem apenas ao imaginar o movimento (LLANOS et al., 2013). Além disso, outra peculiaridade desses fenômenos é o fato de serem melhor observados no hemisfério contralateral ao movimento imaginado. Utilizando a imagética motora das mãos como exemplo, isso significa que a ERD e ERS relacionadas ao movimento da mão direita é mais perceptível no hemisfério esquerdo do escalpo e vice-versa. A Figura 5 mostra que a ERS e ERD produzidas pelo movimento do braço direito são mais acentuadas e perceptíveis no eletrodo C3, localizado no hemisfério esquerdo do escalpo e o mesmo ocorre analogamente para o movimento do braço esquerdo.

Portanto, ao observar os fenômenos de ERD e ERS, as ICM's baseadas em Imagética Motora se concentram em distinguir o sinal de EEG associado à imaginação de diferentes ações motoras simples, como o movimento da mão direita ou esquerda, do pé ou da língua (LOTTE, 2014). Para isso, os sinais de EEG passam por uma cadeia de processamento que visam tratá-los de forma que essa distinção ocorra de forma eficiente. Figura 5 – Dessincronização e Sincronização Relacionada a Evento observadas no espectro do EEG filtrado nas bandas que compreendem os ritmos mu (8-12Hz) e beta (14-30Hz) ao longo da execução do movimento do braço direito. O sinal é aquisitado pelos eletrodos C3 e C4 posicionados no escalpo do indivíduo.



Fonte: Adaptado de (TANG et al., 2019).

2.1.3 Extração de Épocas dos Sinais de EEG

O desenvolvimento de uma ICM pode ser dividido em duas fases: ajuste dos algoritmos e validação. Na etapa de ajuste ou treinamento, segmentos de EEG são entregues à cadeia de processamento que deve extrair informações que auxiliem a distinguir as intenções motoras do usuário. Esses segmentos são delimitados por um instante de tempo conhecido como janela de EEG, que possui largura e posição fixas durante a extração de épocas. Na etapa de validação, a ICM treinada deve classificar as intenções motoras do usuário quando presentes nos sinais de EEG. Portanto, o ajuste da ICM inicia ao escolher os segmentos de EEG utilizados para calibrar os algoritmos.

Para facilitar a manipulação dos dados, as cadeias de processamento baseadas em imagética motora utilizam um protocolo experimental no momento da aquisição dos sinais. O protocolo, através de dicas visuais, rotula o período e a intenção motora que o indivíduo realiza durante a coleta dos dados. Isso permite que os dados possam ser gravados e disponibilizados publicamente, para que possam ser utilizados em diferentes estudos.

A Figura 6 ilustra o processo de extração de épocas realizado com o auxílio de um protocolo experimental. O protocolo supraescrito inicia com a apresentação de uma cruz durante 2 segundos. Em seguida, é apresentada uma dica visual que indica qual imagética motora o usuário deve realizar. O usuário deve realizar a imagética motora indicada durante 3 segundos. Por fim, há uma pausa e o protocolo se repete até que sejam gravadas tentativas suficientes para cada imagética motora do usuário. Geralmente, as épocas de EEG são extraídas logo após o início da dica visual, como mostrado na Figura 6, que apresenta uma janela de extração com início em $t_{min} = 0.5$ segundos e fim em $t_{máx} = 2.5$ segundos.

Figura 6 – Extração de épocas realizada com o auxílio do protocolo experimental aplicado à coleta de sinais de EEG. $t_{min} = 0,5$ e $t_{máx} = 2,5$ denotam o início e o final da janela, respectivamente.



2.1.4 Pré-processamento dos Sinais de EEG

Por se tratar de um sinal de baixa amplitude, o EEG está sujeito a muitos ruídos que podem comprometer a eficiência da ICM (KANG; KAVURI; LEE, 2014). O pré-processamento tem como objetivo isolar o máximo de informação útil possível do sinal bruto e remover os ruídos que o prejudicam, através da aplicação de filtros temporais e filtros espaciais.

Uma estratégia bastante utilizada na literatura atual, na etapa de pré-processamento, é a aplicação de um banco de filtros na configuração de sub-bandas (SBCSP) para realizar a filtragem temporal e espacial dos sinais. A escolha por essa configuração baseia-se em estudos que apontam a utilização de sub-bandas como melhor estratégia para observar os fenômenos da ERD e ERS, que podem ocorrer de forma diferente no espectro do sinal de cada indivíduo (NOVI et al., 2007).

O banco de filtros tem como finalidade filtrar o sinal em diferentes sub-bandas, que juntas abrangem a faixa de frequência dos ritmos mu e beta. Um filtro espacial é aplicado em cada sub-banda e, posteriormente, é feita uma seleção dos dados que melhor representam cada uma das intenções motoras. A decisão final é derivada da fusão desta pontuação de cada sub-banda. Cada uma das etapas é explicada nas sessões a seguir.

2.1.4.0.1 Filtragem Temporal

A filtragem temporal visa a remoção de frequências indesejadas dos sinais de EEG aquisitados. O filtro passa-faixas restringe a análise dos dados a uma determinada faixa de

frequência onde se espera possuir toda a informação útil do sinal que, para o caso das ICM-IM's, trata-se da frequência que compreende os ritmos mu (8-12Hz) e beta (12-30Hz) (LOPES, 2019). Portanto, uma estratégia muito utilizada para a filtragem temporal dos sinais é representa-los no domínio da frequência.

A representação dos sinais no domínio da frequência é feita através da Transformada Discreta de Fourier (DFT – Discret Fourier Transform), que pode ser obtida através do algoritmo da transformada rápida de Fourier (FFT, da sigla em inglês para Fast Fourier Transform). O algoritmo obtém, de forma eficiente, a representação do sinal ao decompô-lo nas bases senoidais a partir de suas harmônicas, transformando-o em um conjunto de coeficientes associados à diferentes frequências que, nesse caso, correspondem à faixa de frequência associada à imagética motora.

Portanto, seja $\mathbf{Z} \in \mathbb{R}^{P \times Q}$ uma única época de EEG com P canais e Q amostras por canal, ao aplicar o algoritmo da FFT aos sinais em cada um dos P canais representados, obtém-se a informação espectral contida em Z na forma da matriz $\mathbf{X} \in \mathbb{C}^{P \times Q}$ cujos elementos da p-ésima linha são definidos por:

$$\mathbf{x}_{pc} = \sum_{q=1}^{Q-1} z_{pq} e^{-\frac{j2\pi qc}{Q}}$$
(2.1)

em que $p \in \{1, 2...P\}$ e z_{pq} representa a q-ésima amostra do p-ésimo canal de Z os termos nos c-ésimos coeficientes complexos correspondem à decomposição do sinal na base de Fourier. Cada coeficiente x_{pc} está relacionado a um bin de frequência discreta ω_c no intervalo $[0, \frac{2\pi(Q-1)}{O}]$.

2.1.4.0.2 Filtragem Espacial

A filtragem espacial busca remover informações desnecessárias ou comuns a eletrodos diferentes, ao passo que acentua as informações dos eletrodos com dados mais úteis para o experimento. Assim a filtragem espacial reduz a quantidade de eletrodos analisados e realça um conjunto específico de eletrodos. Além disso, os filtros espaciais ajudam a recuperar informações da atividade cerebral que foram espalhadas pelos eletrodos devido a interferência do crânio e da própria fisiologia do cérebro (LOTTE, 2014). Para tal, o algoritmo Padrões Espaciais Comuns (CSP) é um dos algoritmos populares em sistemas ICM-IM.

O CSP busca filtros espaciais que maximizam a variância de energia de uma classe e minimizam a de outra classe de forma a ressaltar as características que descrevem a IM (TANG et al., 2019). Ao considerar os sinais filtrados, X, para o conjunto de treinamento associadas à classe, um filtro espacial, que maximiza a separabilidade entre as classes 1 (mão esquerda) e 2 (mão direita) pode ser obtido extremando:

$$arg_{w}max = \frac{\mathbf{w}^{T}C_{1}\mathbf{w}}{\mathbf{w}^{T}(C_{1}+C_{2})\mathbf{w}}$$
(2.2)

em que o sobrescrito $(.)^T$ denota transposição matricial, C_1 e C_2 representam as médias entre as matrizes de covariância espacial das duas classes, respectivamente.

Uma possível solução para a Eq.3.2 é obtida ao resolver $C_1 \mathbf{w} = \lambda (C_1 + C_2) \mathbf{w}$, em que λ e \mathbf{w} correspondem aos autovalores e autovetores generalizados, respectivamente. O problema de autovetores generalizados gera na equação P filtros \mathbf{w}^{Nr} para Nr = 1, 2,... P, cada qual, ao ser aplicado na matriz de entrada, produz o sinal espacialmente filtrado:

$$\mathbf{R} = \mathbf{W}^T \mathbf{X} \tag{2.3}$$

em que $\mathbf{W} \in \mathbb{R}^{Nr \times Q}$. Uma vez que os filtros espaciais correspondentes aos maiores e menores autovalores são os que efetivamente maximizam, de modo recíproco, a variância de uma classe em detrimento de outra, enquanto que autovalores medianos tendem a contribuir menos para a separação, é comum realizar a seleção de Nr < P filtros para compor um vetor de características de dimensão reduzida, sendo Nr necessariamente, natural, positivo e par. Assim, ao receber os sinais filtrados pelas sub-bandas na frequência, os dados são filtrados espacialmente e um novo conjunto de dados.

2.1.5 Extração e Seleção de Características

Após o tratamento dos sinais, a extração de características busca informações que ajudam a traduzir a atividade cerebral do usuário em comando. Os dados extraídos do sinal aquisitado do usuário são analisados e transformados em um novo conjunto que representa de forma mais eficiente as informações relevantes para o experimento (WOLPAW; WOLPAW, 2012).

Na configuração de sub-bandas o vetor de características é extraído de cada sub-banda filtrada e, em seguida, é calculado o score de cada sub-banda, de forma que seja possível identificar quais faixas de frequência possuem as informações mais relevantes para a análise.

Ao considerar os P filtros espaciais (Nr=P), um vetor de características da k-ésima banda de frequência pode ser definido por:

$$f_p^k = log\left(\frac{var(R_p^k)}{\sum_{p=N_r} var(R_p^k)}\right), p = (1, 2...Nr).$$
(2.4)

Em seguida, a Análise Linear Discriminante (LDA) é aplicada como meta-classificador ao vetor de características para realizar a pontuação das sub-bandas. O LDA encontra a matriz de projeção, W_{LDA} , que garante a máxima separabilidade maximizando a razão entre a variância entre classes, S_B para a variância dentro da classe, S_W . A função de custo na k-ésima banda seria:

$$G^{k} = \frac{W_{LDA}^{(k)T} S_{B}^{(k)} W_{LDA}^{(k)}}{W_{LDA}^{(k)T} S_{W}^{(k)} W_{LDA}^{(k)}}$$
(2.5)

onde S_B e S_W são definidos como:

$$S_B^{(k)} = \left(m_2^{(k)} - m_1^{(k)}\right) \left(m_2^{(k)} - m_1^{(k)}\right)^T$$
(2.6)

$$S_W^{(k)} = \sum_{f_p \in C_1} \left(f_p^{(k)} - m_1^{(k)} \right)^2 + \sum_{f_p \in C_2} \left(f_p^{(k)} - m_2^{(k)} \right)^2$$
(2.7)

onde C_1 e C_2 denotam classe 1 e classe 2, respectivamente. O valor de $m_1^{(k)}$ e $m_2^{(k)}$ são as médias de classe empíricas das características do SBCSP calculadas a partir do conjunto de treinamento. Para um problema de 2 classes, por exemplo, o LDA projetará os dados para uma representação unidimensional. Portanto, definimos a pontuação na k-ésima sub-banda como:

$$s_k = W_{LDA}^{(k)T} f_p^{(k)} (2.8)$$

Como dito anteriormente, os sinais de EEG são muito suscetíveis a ruído e, portanto, apenas algumas bandas são úteis. Dessa forma, a seleção de bandas daria intuitivamente uma classificação mais precisa. A fusão de sub-bandas gera um novo vetor que une as características extraídas das sub-bandas para entregar ao classificador final. Existem alguns métodos para realizar a fusão de recursos por pontuação, um deles utiliza classificadores Bayesianos. Assumindo que as distribuições condicionais de classe de pontuações são distribuições normais iguais, ou seja:

$$p(s_k|\boldsymbol{\omega}_i) = \left(2\pi\sigma_i^{(k)2}\right)^2 exp\left(-\frac{\left(s_k - \mu_i^{(k)}\right)^2}{2\sigma_i^{(k)2}}\right),$$
(2.9)

onde $\mu_i^{(k)}$ e $\sigma_i^{(k)}$ são a média e o desvio padrão de características de pontuação, respectivamente, que foram estimadas a partir do conjunto de treinamento. Normalmente, para um classificador Bayesiano, o valor da razão de verossimilhança é usado. Assim, definimos a saída dos classificadores Bayesianos como uma meta-escore e é expresso como k-vetores $[V_1, V_2, V_3, \dots, V_k]^T$ com:

$$V_k = log\left(\frac{p(s_k|\boldsymbol{\omega}_1)}{p(s_k|\boldsymbol{\omega}_2)}\right)$$
(2.10)

2.1.6 Algoritmos de Classificação

A classificação consiste na aplicação de classificadores ou outros algoritmos de aprendizagem de máquina para que os dados recebidos da extração de características sejam interpretados e, então, separadas em classes distintas (HALTAŞ; ERGÜZEN; ERDAL, 2019). No caso da ICM baseada em IM, as classes representam as diferentes intenções do usuário que devem ser observadas em sua atividade cerebral monitorada pelo EEG, como observado na Figura 7 para a imagética motora das mãos. O desempenho de classificação depende das características extraídas do sinal, o que reforça novamente a importância das etapas anteriores.



Figura 7 – Exemplo de Classificação da Imagética Motora das Mãos.

Fonte: Adaptado de (LOTTE, 2014).

Para selecionar o classificador mais apropriado para uma determinada ICM, é essencial entender claramente quais características são usadas, quais são suas propriedades e como são usadas (LOTTE, 2014). Para a classificação binária de IM, ou seja, limitada a duas classes, os classificadores lineares como a Análise Linear Descriminante (LDA) e a Máquina de Vetores de Suporte (SVM) apresentam um bom desempenho. Porém, as características individuais da IM podem dificultar a separação entre as classes de forma linear, o que justifica a utilização de diferentes tipos de algoritmos de classificação nas ICM's. O Perceptron de Multi-Camadas (MLP), K-vizinhos mais próximos (KNN), Regressão Logística (LR) e árvore de decisão (DT) são exemplos de algoritmos comumente aplicados à problemas de classificação.

2.1.6.0.1 Análise Discriminante Linear (LDA)

O LDA (silga em inglês para *Linear Discriminant Analysis*) é um algoritmo de classificação que visa encontrar uma combinação linear de características para distinguir entre duas ou mais classes. Ele é baseado em métodos estatísticos e é amplamente utilizado quando as classes têm distribuições Gaussianas (ou seja, as distribuições das variáveis de cada classe seguem uma distribuição normal) e variâncias iguais. O LDA é tanto um método de classificação quanto uma técnica de redução de dimensionalidade, como apresentado na Seção 2.1.5.

Segundo (LOTTE et al., 2018), o LDA utiliza um hiperplano de separação para realizar a distinção entre classes. Ao considerar um problema de classificação binário, O algoritmo busca por uma direção ótima w^* que maximiza a distância entre as médias das duas classes e minimiza a variância interclasses. Essa solução é obtida extremando a função custo:

$$\arg\max_{\bar{w}} = \frac{\bar{w}^T S_b \bar{w}}{\bar{w}^T S_w \bar{w}}$$
(2.11)

em que S_b é a matriz de dispersão entre as diferentes classes, ou seja, quanto maior essa dispersão, melhor será a separação entre as classes. S_w é a matriz de dispersão dentro de cada classe, ou seja, quanto menor essa dispersão, mais semelhantes são os dados dentro da classe. E w é o vetor de pesos que representa a combinação linear das características.

A classificação de um padrão x_n é obtida em um hiperplano de separação determinado com base no vetor resultante da Equação 2.11 e um limiar de decisão $b \in \Re$ definido por:

$$b = \frac{1}{2}\bar{w}^{*T}(\mu_1 + \mu_2) \tag{2.12}$$

em que μ_1 e μ_2 são os afastamentos dos pontos médios entre os vetores de cada classe $i \in \{1,2\}$, os quais são denotados genericamente por μ_i e definidos conforme a Equação 2.13, onde N_i denota as N amostras para a classe i e x_{in} é o n-ésimo vetor de características da classe i:

$$\mu_i = \frac{1}{N_i} \sum_{n=1}^{N_i} x_{in} \tag{2.13}$$

Portanto, a saída contínua do LDA para um dado vetor de entrada x_n é dada por:

$$f(x_n) = \bar{w}^{*T} x_n - b \tag{2.14}$$

A equação 2.14 define o hiperplano de separação entre as classes. A classe de um vetor de características depende de qual lado do hiperplano o vetor está (veja a figura 8). Para resolver um problema com mais de duas classes, vários hiperplanos são usados em combinação com estratégias como OVR (sigla em inglês para *one versus rest*) e OVO (sigla em inglês para *one versus one*).

Figura 8 – Um hiperplano separando as classes "cruzes" e "círculos".

Fonte: Adaptado de (LOTTE et al., 2018).

2.1.6.0.2 Regressão Logística (LR)

A regressão logística é um algoritmo de aprendizado supervisionado usado para classificação, ou seja, prever a probabilidade de um evento ocorrer, dado um conjunto de variáveis de entrada. Embora seu nome inclua "regressão", ela é amplamente utilizada para problemas de classificação, especialmente quando a variável dependente é categórica, tipicamente binária (BHUVANA; MAHESHWARI; SASIKALA, 2023).

O modelo da regressão logística usa uma função chamada função logística ou função sigmoide, que transforma um valor real em uma probabilidade entre 0 e 1. O modelo é representado como:

$$h_w(\mathbf{x}) = \frac{e^{\mathbf{w}^T \mathbf{x}}}{1 + e^{\mathbf{w}^T \mathbf{x}}} \tag{2.15}$$

onde $h_w(\mathbf{x})$ é uma probabilidade de entre 0 e 1 e representa a hipótese ou a previsão do modelo de regressão logística. O termo *e* é a base do logaritmo natural. E $\mathbf{w}^T \mathbf{x}$ é o produto escalar entre o vetor dos parâmetros \mathbf{w} e o vetor das características \mathbf{x} .

O modelo da regressão logística é treinado usando um conjunto de dados rotulado, ou seja, um conjunto de exemplos nos quais as classes são conhecidas. O objetivo do treinamento é encontrar os pesos **w** que minimizem a diferença entre as previsões do modelo e os valores reais das classes (rótulos) dos dados. Isso é feito utilizando uma técnica chamada gradiente descendente ou máxima verossimilhança para otimizar a função de custo. A função de custo usada na regressão logística é a log-verossimilhança, dada por:

$$J(w) = -\left[\frac{1}{m}\sum_{i=1}^{m} (y_i \log(h_{\mathbf{w}}(x_i)) + (1 - y_i)\log(1 - h_{\mathbf{w}}(x_i)))\right]$$
(2.16)

onde $h_{\mathbf{W}}(x_i)$ é a saída do modelo, y_i é o valor real do rótulo (0 ou 1) e *m* é o número total de exemplos no conjunto de treinamento.

Uma vez treinado o modelo, calcula-se o valor de z e aplica-se a função sigmoide para obter o resultado de P(y = 1|X). Se a probabilidade for maior que 0,5, a classe prevista será 1; caso contrário, será 0.

2.1.6.0.3 Máquina de Vetores de Suporte (SVM)

Assim como a LDA, o SVM também usa um hiperplano discriminante para identificar classes (LOTTE et al., 2018). No entanto, a função objetivo do SVM é maximizar a margem entre os dados de duas classes (ver Figura 9). A margem é a distância entre o hiperplano e os pontos de suporte. Quanto maior a margem, mais confiante o SVM está na sua classificação.

A Figura 9 apresenta dois exemplos de hiperplanos, $g_1 e g_2$. Embora ambos os hiperplanos sejam capazes de separar adequadamente os conjuntos de padrões, o hiperplano g_2 apresenta maior margem de separação. Assim, tal hiperplano possui maiores condições de manter a separabilidade entre as classes caso novos padrões sejam considerados, isto é, sua capacidade de generalização é maior em comparação a g_1 .





Os pontos mais próximos seriam suficientes para determinar um hiperplano de separação ideal, e portanto, são comumente chamados de vetores de suporte. O hiperplano ideal é o que mantém as maiores margens para os vetores suporte de ambas as classes e os pontos situados no hiperplano de separação são os que satisfazem a relação $\bar{w}^T x + b = 0$, em que *w* e *b* são os parâmetros que descrevem o hiperplano e *x* é um vetor de características que atende à condição. O desenvolvimento apresentado é fundamentado na suposição de padrões linearmente separáveis. No entanto, é natural a existência de problemas envolvendo padrões que não são linearmente separáveis. A fim de contornar este tipo de problema, são inseridas variáveis de folga $\xi \ge 0$ (NEGRI, 2021).

Geometricamente, as variáveis de folga reapresentam o deslocamento em que os padrões classificados erroneamente encontram-se do limite da margem de separação referente a sua respectiva classe, conforme ilustra a Figura 10.

Com intuito de incorporar um custo adicional devido à não separabilidade dos padrões, é introduzido o termo $C\sum_{n=1}^{N} \xi_i$ na função objetivo do problema. Esse termo é responsável por contabilizar e penalizar a ocorrência de classificações incorretas segundo o hiperplano de separação definido. Neste contexto, *C* é um termo de regularização usado para penalizar a minimização dos erros nos dados de treinamento em função da minimização da complexidade do modelo (balanço viés-variâncias). A definição de *C* representa um hiperparâmetro do modelo. Assim, é proporcionada a seguinte reformulação sobre o problema:



Figura 10 – Representação do hiperplano de separação para padrões não linearmente separáveis.

Fonte: O autor.

$$\min_{\bar{w},b,\xi} \frac{1}{2} ||\bar{w}||^2 + C(\sum_{i=1}^N \xi_i)$$
sujeito $a: \begin{cases} y_i(\mathbf{w}^T \mathbf{x}_i + b) \ge 1 - \xi_i, i = 1, ..., N \\ \xi_i > 0; i = 1, ...N \end{cases}$
(2.17)

A maximização das margens é feita por uma função de classificação. A função de classificação do SVM é uma função linear, mas pode ser estendida para problemas de classificação não lineares usando o truque do *kernel* (MANKAR; KHOBRAGADE; RAGHUWANSHI, 2016). A ideia básica é que, em vez de calcular diretamente os produtos internos no espaço original (onde os dados podem não ser separáveis), o *kernel* permite que o SVM calcule esses produtos internos em um espaço de maior dimensão de maneira implícita e mais eficiente. O *kernel* é uma função que mapeia os dados de entrada para um espaço de características de maior dimensão e, ao fazer isso, torna os dados linearmente separáveis. A Figura 11 apresenta um exemplo de mapeamento dos padrões para um espaço de maior dimensionalidade.

A escolha do *kernel* do SVM é um importante hiperparâmetro de ajuste do algoritmo pois, dependendo da distribuição dos padrões, uma função *kernel* desempenha melhor o papel de proporcionar a separabilidade entre as classes. Exemplos de funções *kernel* disponíveis para o SVM são: linear, polinomial, RBF (sigla em inglês para *Radian Basis Function*) e sigmóide.

¹ Disponível em: https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d. Acesso em 21/11/2024.





2.1.6.0.4 Perceptron Multicamadas (MLP)

O perceptron Multicamadas (MLP, sigla em inglês para *Multilayer Perceptron*) é composto por uma sequência de camadas de perceptrons, que são unidades de processamento inspirados nos neurônios que podem aprender a realizar operações matemáticas simples (NEGRI, 2021). No modelo do perceptron apresentado na Figura 12, é possível perceber 5 elementos básicos: (i) à esquerda, uma camada composta por *n* receptores, responsáveis por receber as características $x_1, ..., x_n$; (ii) um conjunto de conexões sinápticas, que pondera as informações de entrada segundo os pesos $w_1, ..., w_n$; (iii) uma entrada constante e igual a +1, ponderada por w_0 ; (iv) no núcleo do neurônio, um concentrador (i.e., uma soma Σ) de sinais de entrada que induz um campo sináptico e repassa para; (v) a função de ativação *g*, a qual é responsável por gerar uma resposta.





Um MLP, apresentado na Figura 13, consiste em perceptrons interconectados também chamados de neurônios, organizados em camadas. Cada neurônio recebe sinais de entrada,

executa um cálculo sobre eles usando uma função de ativação e produz um sinal de saída que pode ser transmitido a outros neurônios da rede. Uma função de ativação determina a saída de um neurônio com base em sua entrada. Essas funções introduzem a não linearidade na rede, permitindo que ela aprenda padrões complexos nos dados.



Camada de Entrada



Fonte: (SOBREIRO et al., 2008).

A camada de entrada de um MLP recebe dados de entrada, que podem ser características extraídas das amostras de entrada em um conjunto de dados. Cada neurônio na camada de entrada representa uma característica. Os neurônios da camada de entrada não realizam nenhum cálculo, eles simplesmente passam os valores de entrada para os neurônios da primeira camada oculta.

Entre as camadas de entrada e saída, pode haver uma ou mais camadas de neurônios. Cada neurônio em uma camada oculta recebe entradas de todos os neurônios da camada anterior (seja a camada de entrada ou outra camada oculta) e produz uma saída que é passada para a próxima camada. O número de camadas ocultas e o número de neurônios em cada camada oculta são hiperparâmetros que precisam ser determinados durante a fase de projeto do modelo.

Cada neurônio em uma camada oculta recebe entrada de todos os neurônios da camada anterior. As entradas são multiplicadas pelos pesos correspondentes, denotados como *w*. Os pesos determinam o grau de influência que a entrada de um neurônio tem sobre a saída de outro.

Normalmente, cada neurônio nas camadas ocultas e na camada de saída aplica uma função de ativação à sua soma ponderada de entradas. As funções de ativação comuns incluem sigmoide, tangente hiperbólica (tanh), ReLU (Unidade Linear Retificada) e *softmax* (ESQUI-VEL; VARGAS; LOPEZ-MEYER, 2021). Essas funções introduzem a não linearidade na rede, permitindo que ela aprenda padrões complexos nos dados.

A camada de saída de um MLP produz as previsões finais ou os resultados da rede. O número de neurônios na camada de saída depende da tarefa que está sendo executada (por exemplo, classificação binária, classificação multiclasse, regressão). Cada neurônio na camada de saída recebe entrada dos neurônios na última camada oculta e aplica uma função de ativação. Essa função de ativação geralmente é diferente das usadas nas camadas ocultas e produz o valor final de saída ou a previsão.

2.1.6.0.5 K-Vizinhos mais Próximos (KNN)

O classificador KNN funciona comparando a distância entre uma nova amostra e um conjunto de amostras de treinamento (MLADENOVA; VALOVA, 2022). Um conjunto de amostras de treinamento é fornecido ao algoritmo. Cada amostra é representada por um vetor de características. O algoritmo calcula a distância entre a nova amostra e cada amostra de treinamento. Os K vizinhos mais próximos da nova amostra são selecionados. Então, a classe da nova amostra é determinada com base na classe da maioria de seus vizinhos mais próximos.

O valor de K define o número de vizinhos mais próximos a serem considerados ao fazer a previsão. O valor de K é o hiperparâmetro mais importante do algoritmo e precisa ser escolhido cuidadosamente. K pequeno (por exemplo, K = 1) pode ser muito sensível ao ruído nos dados, levando ao *overfitting*. K grande pode suavizar as decisões e ajudar a reduzir o impacto de dados discrepantes, mas também pode levar a um *underfitting* se K for muito grande, ignorando padrões importantes.

Após a escolha de K, é necessário calcular a distância entre o ponto de teste e os pontos de treinamento. O primeiro passo é medir a distância entre o novo ponto de teste (o ponto a ser classificado ou previsto) e todos os pontos do conjunto de treinamento. A métricas de distância mais comuns é a Distância Euclidiana (para dados contínuos). Dadas as N_r características nos vetores de teste e treinamento, o calculo da distância entre eles na forma euclidiana é definido por:

$$d(\hat{x}, x) = \sqrt{\sum_{j=1}^{N_r} (\hat{x}^{(j)} - x_n^{(j)})^2}$$
(2.18)

em que, $\hat{x}^{(j)}$ e $x_n^{(j)}$ representam o j-ésimo elemento (característica) do vetor de teste e do n-ésimo vetor de treinamento, respectivamente.

2.2 Otimização Bayesiana

No campo do aprendizado de máquina, hiperparâmetros são parâmetros externos ao modelo que influenciam seu comportamento e desempenho durante o processo de treinamento. Diferente dos parâmetros do modelo, que são aprendidos automaticamente a partir dos dados durante o treinamento (como os pesos no perceptron de multicamadas), os hiperparâmetros são definidos antes do treinamento e não são ajustados diretamente pela aprendizagem do modelo (MITRA; BIESSMANN, 2024).

A busca por hiperparâmetros ideais em modelos de aprendizado de máquina (machine learning) é uma das etapas mais cruciais e desafiadoras no desenvolvimento de soluções eficazes. No caso do desenvolvimento das Interfaces Cérebro-Máquina, hiperparâmetros, como a banda de frequência em que os sinais são filtrados, número de sub-bandas, o classificador e os parâmetros do classificador, têm um impacto direto no desempenho e generalização de um modelo. No entanto, a busca por esses valores ótimos não é trivial e pode consumir vastos recursos computacionais e tempo, especialmente quando uma estrutura de ICM é ajustada de forma personalizada para cada sujeito. É nesse contexto que a utilização de algoritmos de otimização se torna essencial (BASHASHATI; WARD; BASHASHATI, 2016).

O ajuste de hiperparâmetros é um problema de otimização com uma função objetivo desconhecida. Como pode ser visto na Equação 2.19, onde *h* denota uma configuração de hiperparâmetro, h_{opt} significa os parâmetros mais promissores dentro de um espaço de busca *H* e g(h) a função objetivo.

$$h_{opt} = \arg\max_{h \in H} g(h) \tag{2.19}$$

Uma das estratégias existentes para solucionar o problema descrito é a Otimização Bayesina (BO). A BO incorpora o conhecimento prévio sobre a função objetivo e atualiza o conhecimento posterior, aumentando a precisão e reduzindo a perda do modelo. A otimização é realizada de acordo com o Teorema Bayesiano, que está representado na Equação 2.20, onde *M* denota o modelo e *O* denota a observação.

$$P(M|O) = \frac{P(O|M)P(M)}{P(O)}$$
(2.20)

onde P(M|O) descreve é a probabilidade *a posteriori* do modelo *M* dado as observações *O*. P(O|M) é a probabilidade de ver as observações *O* dado o modelo *M*, também conhecida como a verossimilhança. P(M) é a probabilidade *a priori* de *M*. P(D) é a probabilidade marginal das observações, que serve como normalizador.

O primeiro passo crítico no processo de BO é a modelagem da função objetivo, onde um modelo probabilístico é utilizado para aproximar g(h). Uma abordagem de modelagem bastante robusta é a Estimadores de *Parzen* com Estrutura em Árvore (TPE, sigla em inglês para *Tree Parzen Estimators*). A TPE opera modelando a distribuição da função objetivo g(h)e selecionando hiperparâmetros com base na inferência probabilística. A abordagem constrói dois modelos separados: um para configurações promissoras e outro para configurações menos promissoras, permitindo uma exploração eficiente do espaço de hiperparâmetros (WATANABE, 2023). O próximo passo é usar uma função de aquisição para escolher o ponto ótimo para *g*. Em seguida, a BO identifica os pontos de amostragem sugeridos conforme determinado pela função de aquisição. Uma função de aquisição bastante utilizada na literatura é a Esperança de Melhoria (EI, sigla em inglês para *Expected Improvement*), ela mede o aumento esperado na função objetivo em relação à melhor observação atual (DAULTON et al., 2024).

A representação matemática da função EI é mostrada na Equação 2.21, onde MSP denota a média dos pontos da amostra, OFH significa a Função de Otimização no ponto mais alto. A função de otimização tem uma limitação de espaço de busca representada como L. A Distribuição Cumulativa, denotada por CD, leva em consideração todos os pontos até a amostra atual sob distribuição Gaussiana (KAMHI et al., 2022).

$$EI = ((MSP - OFH) - L)(CD)$$
(2.21)

O critério de EI proporciona uma maior eficiência na otimização em relação ao número de iterações realizadas e menor custo para selecionar a próxima observação. Após cada avaliação, o modelo probabilístico é atualizado para refletir novas informações, melhorando sua precisão. Várias iterações desse processo são concluídas para ajustar o conjunto de validação e encontrar os hiperparâmetros otimizados que funcionam melhor para o problema de classificação. O fluxograma do algoritmo BO é resumido na Figura 14.



Figura 14 - Fluxo da otimização bayesiana.

2.3 Aprendizado em Conjunto

A aplicação dos algoritmos de classificação depende diretamente da características dos dados e seu contexto e, embora existam muitos tipos de classificadores, a escolha de um só não é suficiente para distinguir as classes dos dados de forma eficiente e robusta. O aprendizado em conjunto é uma vertente que busca ampliar essas possibilidades dos métodos de classificação, ao combinar classificadores conhecidos para resolver os problemas de distinção de classes.

A combinação dos classificadores tem como objetivo explorar as melhores características que diferentes classificadores desempenham diante de uma aplicação. Ao unir esses modelos ajustados é esperado que se obtenha um desempenho de classificação mais robusto e menores taxas de erro (NEGRI, 2021).

No caso da classificação da imagética motora, os modelos de ICM devem extrair informações dos sinais de EEG do usuário na etapa de ajuste e classificá-los corretamente na etapa de teste ou validação, de forma que façam a distinção correta de cada classe de imagética motora do usuário. Na etapa de ajuste, a escolha de diferentes hiperparâmetros dos algoritmos de pré-processamento e de classificação afetam significativamente o desempenho de classificação, de forma que modelos de ICM ajustados de maneira distinta classificam corretamente diferentes exemplos de imagética motora. Portanto, é pertinente avaliar a aplicação dos algoritmos de aprendizado em conjunto para a combinação de ICMs.

As técnicas de combinação são diversas e podem ser aplicadas tanto na saída dos classificadores quanto nas etapas de ajuste dos algoritmos. Além disso, os modelos podem dispor ou não de técnicas de otimização, o que aumenta ainda mais as possibilidades dos arranjos. A seguir, são apresentados os algoritmos de aprendizado em conjunto mais comumente utilizados em problemas de classificação.

2.3.1 Votação por Maioria e Soft Voting

Segundo (NEGRI, 2021), a votação por maioria é uma estratégia intuitiva e de simples implementação. A decisão de classificação é tomada em favor da classe com maior frequência de predição pelos classificadores que constituem a combinação, como mostrado na Figura 15.







Se o número de modelos for ímpar, sempre haverá uma classe vencedora (sem empates). Se o número de modelos for par, pode haver empate, mas diferentes estratégias podem ser adotadas para resolver isso, como escolher a classe de maior probabilidade ou resolver com base no modelo de maior precisão. Assim como a votação por maioria, o *soft voting* é uma técnica de aprendizado de máquina que combina as previsões de vários modelos para gerar uma única previsão. No *soft voting*, cada modelo atribui uma probabilidade a cada classe. Essas probabilidades são então combinadas para gerar uma nova probabilidade para cada classe. A classe com a probabilidade mais alta é então escolhida como a previsão final. Outra maneira de implementar o soft voting é usar uma função de ponderação. A função de ponderação atribui pesos a cada modelo, de modo que as previsões dos modelos mais confiáveis tenham mais peso. A Figura 16 ilustra a estratégia.



Figura 16 – Exemplo de soft voting.

Fonte: O autor.

A votação por maioria e o *soft voting* são utilizados como estratégia de decisão de outras técnicas de aprendizado em conjunto como o *Bagging*, que será apresentado a seguir.

2.3.2 Stacking

A estratégia conhecida como *stacking* consiste na obtenção de um classificador que toma decisões baseadas em decisões anteriores, obtidas por um conjunto de classificadores. Portanto, o processo combina classificadores através de um outro classificador. A Figura 17 demonstra a estratégia descrita.

O conjunto de amostras D é dividido em dois subconjuntos, D1 e D2. Em seguida, o conjunto D1 é utilizado para ajustar N classificadores, que são os modelos base. Posteriormente, um modelo final é treinado usando o conjunto D2 e as previsões dos modelos base como dados de entrada. O classificador final aprende a combinar as previsões dos modelos base para gerar uma previsão mais precisa.

Os modelos base podem ser diferentes tipos de classificadores como regressão logística, ávore de decisão, SVM e etc. Possui custo computacional elevado, porém é considerada uma





estratégia flexível e robusta. O stacking pode ser aplicado tanto em problemas de classificação, quanto em problemas de regressão.

Para a classificação da imagética motora, o algoritmo *Stacking* pode utilizar como parâmetros de ajuste as previsões que cada modelo base produz. No ajuste dos modelos base, pode-se manipular o conjunto de treinamento de forma que cada um deles se torne um especialista na distinção de determinadas classes de imagética motora. E, embora os modelos base não apresentem bom desempenho de classificação para todas as classes, na etapa de ajuste do classificador final do *Stacking*, o mesmo pode aprender como utilizar o comportamento dos modelos bases para realizar a previsão final.

2.3.3 Bagging

A estratégia de *bagging* começa por amostrar os dados originais com (*bootstraping*) ou sem reposição, gerando conjuntos de dados variados. Cada conjunto de dados é então usado para treinar um modelo de aprendizado de máquina, como uma árvore de decisão, uma rede neural ou um SVM. Os resultados dos modelos são então agregados para produzir uma previsão final (RAHIMI et al., 2016). A predição final é obtida por meio da votação por maioria ou pelo *soft voting*. A Figura 18 apresenta a estratégia.

A estratégia de *bagging* funciona melhor para algoritmos de aprendizado de máquina que são propensos ao *overfitting*. O *overfitting* ocorre quando um modelo se ajusta muito bem aos dados de treinamento, mas não é capaz de generalizar para novos dados. A estratégia de bagging ajuda a reduzir o overfitting, gerando vários modelos diferentes que são menos propensos a se ajustarem aos dados de treinamento de forma excessiva.

No caso da classificação da imagética motora, os classificadores combinados são modelos de ICMs ajustados com diferente conjuntos de dados. O objetivo é que cada modelo consiga identificar de formas diferentes as intenções motoras do usuário quando presentes nos sinais de

² Disponível em: https://www.geeksforgeeks.org/ml-bagging-classifier/. Acesso em 21/11/2024.



Figura 18 – Etapas do processo de bagging.

Fonte: Adaptado de geeksforgeeks.org²

EEG, e que a combinação das classificações dos modelos produza uma classificação final mais precisa e robusta.

2.4 Métricas de Avaliação

A combinação dos modelos proposta nesta pesquisa deve classificar a imagética motora de cada sujeito de forma eficiente e robusta. Além disso, é preciso avaliar estatisticamente se o desempenho dos modelos combinados é igual, superior ou até mesmo inferior aos modelos individuais personalizados. Portanto, para avaliar os resultados obtidos nos experimentos, serão utilizadas como métricas de avaliação a acurácia de classificação e o índice *kappa*.

As métricas supracitadas podem ser determinadas a partir da matriz de confusão. A matriz de confusão é uma ferramenta fundamental para avaliar a performance de modelos de classificação, particularmente em problemas de aprendizado supervisionado. Ela é uma tabela que resume o desempenho do modelo, comparando as previsões feitas pelo modelo com os valores reais (verdadeiros) de uma base de dados. A Figura 19 exemplifica uma matriz de confusão. Seja o espaço de classes $\Omega = \{\omega_1...\omega_c\}$ e um conjunto de referência *A*, cujo os indicadores de classe são admitidos como corretos. Nesta matriz, a_{ij} representa o número de exemplos em *A* que foram associados à classe ω_i , sendo tais exemplos pertencentes à classe ω_j . Ainda, a_{i+} e a_{+i} representam as as quantidades marginais em relação à classe ω_i em termos do classificado pelo modelo e da classe esperada, respectivamente. As somas das quantidades marginais devem resultar em *m*, ou seja, no número de exemplos classificados pelo modelo.

A acurácia de classificação ou acerto global é a métrica mais utilizada e a mais simples para avaliar o desempenho de modelos de classificação. Ela avalia o percentual de acertos geral da classificação ao dividir o número de acertos pelo número total de classificações, como mostrado na Equação 2.22.

		Classe Esperada				
		ω_1	ω_2		ω_c	
ssificação	ω_1	<i>a</i> ₁₁	<i>a</i> ₁₂		a_{1c}	<i>a</i> ₁₊
	ω_2	<i>a</i> ₂₁	a_{22}		a_{2c}	a ₂₊
	•		•	•		•
		•				
Cla	ω_c	a_{c1}	a_{c2}	•	a_{cc}	a_{c+}
		a ₊₁	a ₊₂	-	a_{+c}	m

Figura 19 – Matriz de confusão.

Fonte: O autor.

$$P_o = \frac{1}{m} \sum_{i=1}^{c} a_{ii}$$
(2.22)

O Índice *kappa* verifica a concordância entre a classificação efetuada e a realidade, assim como pode ser utilizado para determinar se os acertos alcançados são significativamente melhores que uma classificação esperada por acaso (CHICCO; WARRENS; JURMAN, 2021). O índice *kappa* é dado por:

$$\kappa = \frac{P_o - P_e}{1 - P_e},\tag{2.23}$$

onde P_o é a proporção de concordância entre os classificadores, ou seja, a frequência em que os classificadores concordam em suas classificações. O indicador P_e é a proporção de concordância esperada, assumindo que eles estão classificando e concordando por acaso dado por:

$$P_e = \frac{\sum_{i=1}^{c} a_{i+}a_{+1}}{m^2}.$$
(2.24)

O valor do índice *kappa* varia de -1 a 1, onde $\kappa = 1$ significa a concordância perfeita entre a classificação e o esperado. O indicador $\kappa = 0$ reflete concordância igual à que seria esperada por acaso (sem concordância além da chance). E $\kappa < 0$ indica desacordo maior do que o esperado por acaso, o que é raro e indicaria uma discrepância significativa entre o classificador e as predições reais.

A variância do índice *kappa* (σ_{κ}^2) também é uma importante ferramente utilizada para a análise dos resultados de classificação de modedlos. Além da noção da variabilidade dos dados, esta medida auxilia na verificação da significância entre os resultados de classificação quando expressos em termos do índice *kappa*. A variância do índice *kappa* é dada por:

$$\sigma_{\kappa}^{2} = \frac{1}{m} \left\{ \frac{\theta_{q}(1-\theta_{1})}{(1-\theta_{2})^{2}} + \frac{2(1-\theta_{1})(2\theta_{1}\theta_{2}-\theta_{3})}{(1-\theta_{2})^{3}} + \frac{(1-\theta_{1})^{2}(\theta_{4}-4\theta_{2}^{2})}{(1-\theta_{2})^{4}} \right\}$$
(2.25)

onde

$$\theta_1 = P_o, \tag{2.26}$$

$$\theta_2 = P_e, \tag{2.27}$$

$$\theta_3 = \frac{1}{m^2} \sum_{i=1}^c a_{ii} (a_{i+} + a_{+i}), \qquad (2.28)$$

$$\theta_4 = \frac{1}{m^3} \sum_{i=1}^c \sum_{j=1}^c a_{ij} (a_{j+} + a_{+i}).$$
(2.29)

Admitindo os coeficientes κ_1 e κ_2 , cuja as respectivas variâncias são $\sigma_{\kappa_1}^2$ e $\sigma_{\kappa_2}^2$, sabe-se que a estatística:

$$z_{\kappa} = \frac{|\kappa_1 - \kappa_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{2.30}$$

segue a distribuição gaussiana padrão ($z_{\kappa} \sim N(0, 1)$). Nessas condições, é possível afirmar que as classificações associadas a κ_1 e κ_2 são estatisticamente significantes a um nível α quando a probabilidade de ocorrer um valor igual ou maior que z_{κ} ; denominado p - valor, torna-se inferior a $1 - \alpha$. Em outras palavras, para $\alpha = 0,05$, um valor de z_{κ} maior ou igual a 1,96 significa que a diferença entre os classificadores é estatisticamente significativa, a um nível de confiança de 95%. Caso contrário, não há diferença significativa entre os classificadores.

A precisão é outra métrica comum no processo de avaliação de modelos de classificação. A precisão mede a quantidade de vezes que o modelo classifica corretamente uma determinada classe em relação ao total de vezes que ele associa um determinado exemplo a essa classe. A precisão de um modelo para uma determinada classe pode ser expressa como:

$$P_r = \frac{VP}{VP + FP} \tag{2.31}$$

onde VP indica a quantidade corretamente predita pelo modelo e o parâmetro FP expressa as previsões erradas do modelo para essa determinada classe.

A acurácia de classificação é utilizada para avaliar a quantidade de exemplos classificados corretamente pelos modelos avaliados neste trabalho. Os modelos de ICM personalizados devem ser ajustados com diferentes amostras dos sinais de EEG do usuário, de forma que extraiam uma maior variedade de informações da imagética motora. Portanto, espera-se que a combinação dos modelos aumente a quantidade de exemplos classificados corretamente. A análise do índice z_{κ} deve confirmar se a combinação dos modelos produz uma diferença estatisticamente significativa, uma vez que os modelos individuais já apresentam um bom desempenho de classificação. Além
disso, a precisão deve evidenciar o desempenho dos modelos ao classificar cada uma das classes de imagética motora avaliadas.

2.5 Trabalhos Relacionados

O objetivo desta pesquisa tem como escopo principal a aplicação das técnicas de aprendizado em conjunto para a classificação da imagética motora. Espera-se que a combinação de ICM's aproveite as melhores características que os diferentes modelos combinados apresentam diante do problema de classificação da imagética motora e, assim, aumente o desempenho de classificação. A utilização de um algoritmo de otimização para o ajuste de cada ICM utilizada deve garantir que os modelos apresentem um bom desempenho e classifiquem corretamente diferentes exemplos de imagética motora. Além disso, é esperado que os modelos combinados apresentem desempenho significativamente superior aos modelos individuais. Dessa forma, é necessário estudar como os algoritmos de aprendizado em conjunto podem ser aplicados ao desenvolvimento de interfaces cérebro-máquina. Assim, esta seção apresenta pesquisas que aplicam esses algoritmos no contexto de interfaces cérebro-máquina. Também é selecionada uma pesquisa que busca o aumento de desempenho de classificação da imagética motora através de uma técnica de pré-processamento alternativa ao filtro CSP, a escolha desta pesquisa tem como objetivo comparar as diferentes abordagens de melhoria de classificação.

Os autores de (MIAO et al., 2021) ajustam três diferentes modelos de interfaces cérebromáquina que, em seguida, são combinadas para realizar a classificação da imagética motora. Os modelos combinados possuem a mesma estrutura de pré-processamento com filtragem por sub-banda e SVM como classificador final, contudo, cada modelo é ajustado com uma janela de EEG diferente. O objetivo de utilizar janelas diferentes para o ajuste dos modelos é selecionar amostras diferentes do sinal de EEG de forma que cada modelo extraia informações diferentes da imagética motora do sujeito. Duas abordagens são utilizadas para a combinação dos modelos. A primeira consiste no uso da votação por maioria para combinar a saída de cada classificador, como no algoritmo de aprendizado em conjunto *Bagging*. A segunda consiste em utilizar um classificador final SVM para realizar as classificações finais com base no vetor de característica formado por cada SVM base, como no algoritmo *Stacking*. Os modelos propostos são então comparados entre si e entre outros modelos tradicionais de ICM, utilizando bases de dados disponibilizadas por competições de ICM. Os resultados obtidos, em termos de acurácia de classificação, são de 84,47% para a primeira abordagem e 84,57% para a segunda abordagem, contudo, não há diferença estatisticamente significante entre as duas.

A pesquisa desenvolvida pelos autores de (ALFEO et al., 2021) propõe uma arquitetura de ICM que combina a abordagem de *Stacking* com um esquema de classificação binária umcontra-um. A metodologia implementa a computação em duas camadas: a primeira emprega um conjunto de regressores com um esquema de decomposição um-vs-um, a segunda consiste em um classificador final. Tanto os regressores quanto o classificador são implementados como perceptrons multicamadas (MLP). O objetivo dos autores é aproveitar a especialidade que cada regressor base tem na distinção de cada par de classes de imagética motora para distinguir simultaneamente as quatro classes de imagética motora da base de dados avaliada. O resultados são produzidos utilizando uma base de dados pública e apresentam um índice *Kappa* médio de 85% para os sujeitos avaliados, superando as estratégias tradicionais utilizadas para comparação.

Na pesquisa de (WEI; DONG; ZHU, 2021), os autores utilizam modelos de ICM com filtragem por sub-banda e uma estratégia multiclasse um-contra-um na classificação de 4 classes, e a estratégia de Votação Majoritária é adotada para os diferentes classificadores individuais. A ideia principal da pesquisa é semelhante à abordagem de (ALFEO et al., 2021), contudo, os modelos utilizados para combinação possuem diferentes classificadores, assim como em (MIAO et al., 2021). Os resultados são apresentados em termos de acurácia de classificação e superam as metodologias utilizadas para comparação com uma média de 68,52%. Os resultados confirmam ainda que os modelos combinados superam o desempenho dos modelos individuais.

Finalmente, a pesquisa realizada por (BARACHANT et al., 2012) não aplica técnicas de aprendizado em conjunto, mas apresenta uma alternativa a filtragem de sub-banda utilizando filtros CSP. A metodologia utilizada denominada Espaço Tangente LDA (TSLDA, sigla em inglês para *Tangent Space LDA*), mapeia as matrizes de covariância no espaço tangente Riemanniano, onde as matrizes podem ser vetorizadas e tratadas como objetos euclidianos. Em seguida, um algoritmo de extração de características é aplicado para diminuir a dimensionalidade do vetor de características e uma classificação por LDA é realizada. A abordagem descrita procura aumentar a separabilidade entre as classes de imagética motora utilizando o conceito de distância de Riemann (CONGEDO; BARACHANT; BHATIA, 2017). O método descrito melhora o ajuste do classificador que é então capaz de distinguir com mais precisão as classes de imagética motora. Os resultados da pesquisa são gerados a partir de uma base de dados pública de competição de ICM para quatro classes e superam técnicas tradicionais de classificação apresentando uma acurácia de classificação média de 70,2%.

Ao avaliar as pesquisas selecionadas é possível perceber que a combinação de modelos de ICM é capaz de aumentar o desempenho da classificação da imagética motora. O trabalho de (MIAO et al., 2021) sugere que ICMs ajustadas com diferentes janelas de EEG extraem informações diferentes dos sinais de EEG que, ao serem combinadas, aumentam o desempenho de classificação. A pesquisa de (ALFEO et al., 2021) sugere que é pertinente ajustar modelos especialistas na classificação binária para, em seguida, combiná-los utilizando um classificador final, os resultados sugerem que o classificador final é capaz de aproveitar a especialidade que cada modelo base possui na distinção de duas classes. Os resultados da pesquisa desenvolvida por (WEI; DONG; ZHU, 2021) sugerem que a combinação de modelos que utilizam diferentes classificadores é capaz de aproveitar as diferentes características que cada um apresenta diante do problema. As pesquisas selecionadas combinam os modelos de ICM através de algoritmos

como o *Bagging* e o *Stacking* e utilizam estratégias como a Votação por Maioria e o *Soft Voting* para realizar a classificação final. Porém, as pesquisas mencionadas não aplicam técnicas de otimização de forma individual para o ajuste dos modelos, apresentando assim uma lacuna explorada por esta pesquisa.

Finalmente, o trabalho desenvolvido por (BARACHANT et al., 2012) propõe que um único modelo de ICM é capaz de produzir um bom desempenho de classificação da imagética motora ao utilizar técnicas diferentes de pré-processamento. Esta abordagem é particularmente interessante para fins de comparação pois apresenta uma forma mais simples de melhorar o desempenho de classificação, uma vez que realiza o ajuste de um único modelo de ICM.

3 METODOLOGIA

Este trabalho propõe investigar se a combinação de interfaces cérebro-máquina personalizadas com otimização é capaz de melhorar o desempenho de classificação da imagética motora. A metodologia aplicada também investiga qual é o melhor método de combinar os modelos ajustados ao aplicar e comparar o desempenho de algoritmos de aprendizado em conjunto.

A proposta investiga se os fenômenos que definem a imagética motora do usuário podem ser melhor identificados pela combinação de diferentes ICM's ajustadas de forma personalizada para cada sujeito, uma vez que as interfaces são ajustadas com diferentes hiperparâmetros e, portanto, extraem informações distintas dos sinais que podem melhorar na distinção das classes. Para tal, o experimento conduzido nesta pesquisa se divide nas seguintes etapas: seleção das bases de dados e ambiente computacional, ajuste dos modelos, aplicação do aprendizado em conjunto e avaliação dos modelos propostos, como mostrado na Figura 20. Cada uma das etapas mencionadas está descrita nas seções a seguir.



Figura 20 - Etapas do desenvolvimento da metodologia.

Fonte: O autor.

3.1 Ambiente de Execução dos Experimentos e Descrição da Bases de Dados

O ajuste das interfaces cérebro-máquina requer a manipulação de uma grande quantidade de dados e a aplicação de diferentes algoritmos de aprendizado de máquina. Para isso, existem ambientes computacionais que dispõem de ferramentas específicas que facilitam o processo de ajuste e validação dessas interfaces.

Os experimentos nesta pesquisa serão realizados no *Google Colaboratory*, ambiente de notebooks *Jupyter* que utiliza a linguagem *Python*. O ambiente não requer configuração e

é executado na nuvem do *Google*, o que permite que as análises possam ser compartilhadas e o acesso a um grande recurso computacional. As principais bibliotecas utilizadas para o desenvolvimento das análises são a *scikit-learn* e a *MNE-Python*. O *scikit-learn* é uma biblioteca da linguagem *Python* desenvolvida especificamente para aplicação prática de aprendizado de máquina (PEDREGOSA et al., 2011). A biblioteca dispõe de ferramentas simples e eficientes para análise preditiva de dados. Já a *MNE-Python* é um pacote *Python* de código aberto para explorar, visualizar e analisar dados neurofisiológicos humanos, como o EEG (GRAMFORT et al., 2013). O MNE-*Python* possui interfaces para uma variedade de conjuntos de dados disponíveis publicamente, além de diversas ferramentas de pré-processamento muito úteis para o desenvolvimento de ICMs.

As pesquisas que estudam melhorar o desempenho das ICM's baseadas na imagética podem utilizar bases de dados públicas para o ajuste e validação dos modelos propostos. Essas bases possuem dados de EEG coletados de diferentes indivíduos com a ajuda de um protocolo experimental que demarca os períodos de execução da imagética motora, como apresentado na Seção 2.1.3. As coletas são feitas por especialistas e os dados adquiridos são fornecidos por competições de ICM junto com os detalhes do protocolo utilizado. Portanto, o conjunto de dados 2a da Competição IV de ICM (BRUNNER et al., 2008) é selecionada para o desenvolvimento desta pesquisa.

Os dados da base incluem sinais de EEG de 9 sujeitos enquanto realizavam 4 tarefas de IM: mão esquerda (ME), mão direita (MD), pés (PE) e língua (LG). O conjunto de treinamento possui 288 tentativas, sendo 72 para cada IM. Os sinais foram coletados de P = 22 canais de EEG (ver Figura 21), a uma taxa de amostragem F_s = 250 Hz. O conjunto de validação possui 288 tentativas, sendo 72 para cada classe.





Fonte: (BRUNNER et al., 2008).

3.2 Ajuste dos Modelos de Interface Cérebro-Máquina

Cada etapa da ICM-IM, descrita na seção 2.1, possui uma gama de algoritmos e técnicas possíveis que podem ser utilizadas para formar a cadeia de processamento dos sinais de EEG e cada algoritmo possui um ou mais hiperparâmetros de ajuste. Neste trabalho, a escolha desses hiperparâmetros é realizada de forma personalizada para os sujeitos dos conjuntos de dados a fim de produzir melhor desempenho de classificação. A Figura 22 exemplifica o processo de ajuste da ICM por etapa e cada um dos hiperparâmetros relacionados. Para aumentar a eficiência da busca pelos melhores hiperparâmetros de ajuste para cada sujeito, o experimento conduzido nesta pesquisa utiliza a plataforma *AutoBCI* desenvolvida no trabalho de (VILAS-BOAS; SILVA, 2020). A plataforma utiliza a Otimização Bayesiana para ajustar os parâmetros dos modelos de ICM-IM.





A Otimização Bayesiana é uma técnica de otimização utilizada para encontrar o valor ótimo de uma função com o menor número de avaliações dessa função. No trabalho de (VILAS-BOAS; SILVA, 2020) a proposta é modelar a acurácia de generalização do classificador final em função dos hiperparâmetros correspondentes e encontrar um conjunto ótimo de valores que minimizam o erro de classificação ou maximizam a acurácia esperada. A modelagem do problema utiliza a abordagem da TPE e a Esperança de Melhoria (EI) como função de aquisição. O espaço de configuração *H*, ou seja, o conjunto de hiperparâmetros da ICM-IM a serem otimizados, está representado na Tabela 1.

Hiperparâmetro	Variável	Domínio	Etapa da ICM
Largura	t _{min}	[0, 0.5, 2.5, 3]	Extração
e localização da janela de EEG	t _{máx}	[1, 1.5, 3.5, 4]	de épocas
Banda de filtragem	f_l	[0.5, 1, 29, 30]	Pré-
	f_h	[8, 9, 48, 49]	processamento
Número de sub-bandas	n _{band}	[1, 2 23, 24]	Pré- processamento
Número de CSPs	n _{csp}	[2, 4 <i>N_{canais}</i>]	Pré- processamento
Classificador	clf	[LDA, LR, SVM, KNN, MLP]	Classificação

Tabela 1 – Hiperparâmetros configuráveis em cada etapa da ICM.

Fonte: O autor.

A limitação da largura e localização da janela de EEG se dá pelo protocolo experimental utilizado na base de dados descrita em 3.1. A banda de filtragem escolhida para o espaço de busca compreende a banda de frequência onde a IM é mais presente. A quantidade de sub-bandas é limitada pela banda máxima de filtragem, pela largura de 4 Hz por sub-banda e pela sobreposição de 50% entre cada sub-banda. O número de CSPs é limitado pelo número de eletrodos disponíveis (descrito em 3.1).

Os classificadores, por serem algoritmos distintos entre si, possuem seus próprios hiperparâmetros de ajuste e, portanto, estes estão apresentados na Tabela 2. A Regressão Logística (LR) e a Análise de Discriminantes Lineares (LDA) utilizam o modelo padrão da biblioteca *scikit-learn* (PEDREGOSA et al., 2011).

Classificador	Hiperparâmetros	Domínio					
LDA	Padrão Sklearn	-					
LR	Padrão Sklearn	-					
SVM	Regularização	$[10^8, 10^7 \dots 10^4]$					
	Função de classificação	[Linear, Sigmoid, Gaussian, Polynomial]					
KNN	Número de vizinhos próximos	[2, 3 Népocas]					
	Número de camadas ocultas	[1, 2]					
MLP	Número de neurônios/Camada	[10, 20, 200]					
	Taxa de aprendizagem	$[10^5, 10^4 \dots 10^2]$					
Fonte: Q autor.							

Tabela 2 – Hiperparâmetros configuráveis em cada classificador.

O problema de encontrar valores ótimos h_{opt} para os hiperparâmetros da cadeia de processamento ICM-IM pode ser expresso como:

$$h_{opt} = \arg\max_{h \in H} g(h) \tag{3.1}$$

Para tanto, a cada iteração *i* da busca Bayesiana, com $i = 1, 2, ...N_{iter}$, avalia-se o desempenho médio de generalização da acurácia de classificação associado ao i-ésimo conjunto de hiperparâmetros h_i . Para a geração dos modelos de ICM-IM M_{hi} utilizados neste trabalho, foram utilizadas 100 iterações ($N_{iter} = 100$).

Por padrão a plataforma *AutoBCI* realiza uma validação cruzada de cinco pastas na busca pelo conjunto ótimo de hiperparâmetros. Isso significa que em cada uma das 100 iterações da otimização, o conjunto de treinamento D_T é dividido em cinco sub-conjuntos de igual tamanho. Uma série de cinco validações é executada, cada uma tomando um dos sub-conjuntos para representar o conjunto de teste D_{Te} e os quatro sub-conjuntos sobressalentes para representar D_{Tr} . A cada validação da série, um novo sub-conjunto é usado para representar D_{Te} e assim sucessivamente até que os cinco conjuntos tenham sido utilizados.

Em cada pasta o modelo M_{hi} é ajustado usando o conjunto D_{Tr} corrente e testado a partir do respectivo conjunto D_{Te} . Ao final das 5 pastas, a acurácia média de generalização entre elas define a saída g(hi) associada à i-ésima combinação do vetor de hiperparâmetros. Em outras palavras, a função g(hi), a qual deseja-se maximizar, representa a acurácia de classificação média da i-ésima iteração, em que o desempenho na j-ésima avaliação na validação cruzada é dado pela relação entre o número de épocas corretamente classificadas e o total de épocas no conjunto D_{Te} nos termos da Equação 2.22.

Ao final das 100 iterações, a plataforma registra as melhores configurações de hiperparâmetros para cada sujeito avaliado. Esse processo é realizado para a distinção aos pares de cada classe de imagética motora da base de dados avaliada, ou seja, ME *versus* MD, ME *versus* PE, ME *versus* PE, MD *versus* PE, MD *versus* TG e PE *versus* TG. A Figura 23 ilustra o processo de geração dos modelos de ICMs-IM otimizados.

Figura 23 – Diagrama funcional do processo de ajuste do modelo de classificação a partir da plataforma *AutoBCI*.



Fonte: Adaptado de (VILAS-BOAS; SILVA, 2020).

Os modelos de ICM-IM gerados para cada sujeito e para cada par de classes estão descritos no Apêndice A deste trabalho.

3.3 Aplicação dos Algoritmos de Aprendizado em Conjunto

Para a aplicação dos algoritmos de aprendizado em conjunto proposta neste trabalho, são selecionados os três modelos que apresentam as maiores acurácias de classificação com a mesma largura de janela $(t_{máx} - t_{min})$ para cada sujeito e para a distinção de cada um dos pares de classes supracitados. A escolha deste número de modelos garante o menor número de modelos combinados para realizar a Votação por Maioria descrita na Seção 2.3.1. Portanto, para cada sujeito avaliado ao todo são produzidos dezoito modelos de ICM otimizados. A escolha de uma largura de janela padroniza o número de épocas que cada modelo utiliza para ajuste, neste estudo, a largura da janela escolhida é de dois segundos ($t_{máx} - t_{min} = 2$). A Figura 24 exemplifica três modelos selecionados para a classificação das classes ME e MD do sujeito 1.



		P	ré-pro	ocessa	mento	D C			
	fi	fh	tmin	tmax	ncsp	Nband	Classificador		
[]									
Modelo 1	11	49	0,5	2,5	2	18	{'C': -1.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}		
Modelo 2	10	50	0	2	4	19	{'model': 'LDA'}		
					1				
Modelo 3	3	30	1	3	22	11	{'model': 'KNN', 'neig': 2}		

Fonte: O autor.

A aplicação dos algoritmos de aprendizado em conjunto consiste na manipulação do conjunto de dados que ajustam os modelos de ICMs e combinação das classificações individuais dos modelos. Essa abordagem tem se mostrado eficaz em uma variedade de tarefas de classificação, pois permite que os erros de um modelo sejam compensados por outros, resultando em uma melhor generalização e maior resistência a *overfitting*.

Entre os métodos mais comuns de aprendizado em conjunto, destacam-se o *Bagging* e o *Stacking*. O *Bagging* tem como princípio básico combinar os resultados de modelos treinados de forma independente para reduzir a variância do modelo final e, assim, melhorar seu desempenho. A ideia central do *stacking* é treinar diversos modelos de base, gerar suas previsões, e depois usar essas previsões como entradas para um classificador que vai aprender a melhor forma de

combinar essas previsões. Nesta pesquisa, é investigada a aplicação do *bagging* e do *Stacking* para a combinação dos modelos de ICMs otimizados.

Nesta pesquisa, os algoritmos de aprendizado em conjunto são aplicado aos modelos em duas abordagens: classificação binária e classificação multi-classe. Na classificação binária, os modelos ajustados para a distinção de um determinado par de classes são combinados com o objetivo de aumentar o desempenho de classificação do modelo combinado em relação ao respectivo par de classes. Em contrapartida, na abordagem multi-classe, todos os dezoito modelos ajustados de forma binária para cada sujeito são combinados para realizar a distinção das quatro classes de imagética motora. Na abordagem binária é utilizado o *bagging* com o *Soft Voting* e Votação por Maioria como estratégias de classificação final. Já na abordagem multi-classe são utilizados o *Bagging* com *Soft Voting* e o *Stacking* com um MLP como classificador final. A metodologia de cada uma das abordagens é descrita nas seções a seguir.

3.3.1 Aplicação do Bagging - Abordagem binária

Para a aplicação do algoritmo de aprendizado em conjunto *Bagging*, o conjunto de dados contendo as amostras de IM de cada sujeito é dividido em quatro conjuntos com diferentes exemplos e de igual quantidade cada. Sendo três utilizadas para treinamento dos modelos personalizados e um para a validação. Na etapa de treinamento, cada modelo é ajustado com um conjunto diferente dos dados, para garantir que os modelos extraiam diferentes informações dos dados. Na etapa de validação, os modelos classificam as mesmas amostras do conjunto de dados para a classificação, sendo essas extraídas da janela de EEG compreendida entre $t_{min} = 0,5$ e $t_{máx} = 2,5$ segundos. Após a classificação dos dados de validação, as saídas dos classificadores são combinadas para gerar a classificação final. Duas estratégias de combinação são utilizadas: a Votação por Maioria e *Soft Voting*. Este processo é repetido quatro vezes, alternando o conjunto utilizado para a validação dos dados. A Figura 25 ilustra a metodologia aplicada.

O processo de aplicação do *Bagging* descrito é feito para a distinção de cada um dos pares de classes. A metodologia proposta busca combinar o desempenho de classificação dos modelos para produzir um modelo combinado que desempenha melhor na distinção das duas classes. É esperado que os modelos individuais classifiquem corretamente exemplos diferentes e contribuam para o bom desempenho do modelo combinado.

3.3.2 Aplicação do Bagging - Abordagem multi-classe

Para a aplicação do algoritmo de aprendizado em conjunto *Bagging* para a abordagem multi-classe, o conjunto de dados de EEG de um determinado sujeito, contendo as amostras das quatro classes de IM é dividido em quatro conjuntos com diferentes exemplos e de igual quantidade cada. Sendo três utilizadas para treinamento dos modelos personalizados para a distinção binária e um para compor o conjunto de validação multi-classe. Na etapa de treinamento,



Figura 25 – Esquema de aplicação do algoritmo Bagging para a classificação binária.

cada modelo é ajustado com um conjunto diferente dos dados de duas classes, para garantir que os modelos extraiam diferentes informações dos sinais de EEG. Na etapa de validação, os modelos classificam amostras do conjunto de dados multi-classe, sendo essas extraídas da janela de EEG compreendida entre $t_{min} = 0,5$ e $t_{máx} = 2,5$ segundos. Após a classificação dos dados de validação, as saídas dos classificadores são combinadas para gerar a classificação final. A estratégia de combinação utilizada é a *Soft Voting*. Este processo é repetido quatro vezes, alternando os conjuntos utilizados para compor o conjunto de validação multi-classe. A Figura 26 ilustra a metodologia aplicada.



Figura 26 – Esquema de aplicação do algoritmo bagging para a classificação multi-classe.

A metodologia descrita busca aproveitar o desempenho individual de cada um dos

modelos personalizados para a distinção das quatro classes de imagética motora avaliadas. É esperado que os modelos treinados de forma binária apresentem maior certeza ao classificar as amostras das classes em que foram ajustados e, assim, contribuam de forma mais significativa no *Soft Voting*. Naturalmente, a acurácia dos modelos individuais para a distinção das quatro classes é baixa, porém, devem apresentar precisão elevada na distinção de pelo menos duas classes. Ao aplicar o *Bagging* multi-classe, o modelo combinado deve aproveitar essa precisão e apresentar uma acurácia de classificação elevada para as quatro classes.

3.3.3 Aplicação do *Stacking* - Abordagem multi-classe

A aplicação do algoritmo de aprendizado em conjunto *Stacking* para a abordagem multiclasse segue uma metodologia semelhante ao *bagging*, porém, um MLP é utilizado para realizar a classificação final. O processo é dividido em três etapas: treinamento dos modelos de classificação binária, treinamento do MLP e validação. O treinamento dos modelos para a classificação binária segue a mesma metodologia descrita na seção anterior. Na etapa de treinamento do MLP, os conjuntos utilizados para o ajuste dos modelos de classificação binária são combinados em um conjunto de treinamento multi-classe e entregues para os modelos ajustados. As probabilidades atribuídas por cada modelo são então utilizadas como vetor de características para o treinamento do MLP. Assim, na etapa de validação, um conjunto de avaliação multi-classe é entregue aos modelos ajustados e o MLP classifica as amostras (Extraídas da janela de EEG compreendida entre $t_{min} = 0, 5$ e $t_{máx} = 2, 5$ segundos) com base nas probabilidades atribuídas por cada um dos modelos. O processo também é repetido quatro vezes alternando os conjuntos de validação. O MLP utilizado possui 100 neurônios e apenas uma camada oculta. A Figura 27 ilustra o processo descrito.



Figura 27 – Esquema de aplicação do algoritmo stacking para a classificação multi-classe.

A adição de um classificador para realizar a classificação final tem como objetivo utilizar

as probabilidades atribuídas por cada modelo como informação a respeito das amostras avaliadas. Naturalmente, os modelos binários não são capazes de classificar corretamente todas as quatro classes de imagética motora, porém, devem atribuir maior probabilidade as classes em que são especialistas e baixa probabilidade para as classes que desconhecem. Assim, é esperado que o classificador final consiga distinguir as quatro classes com base no comportamento dos modelos binários.

3.4 Avaliação dos modelos propostos

Na abordagem binária, após a combinação das classificações dos modelos, são calculadas as métricas de avaliação dos modelos individuais e do modelo combinado: A acurácia média, κ médio, o índice z_{κ} entre o melhor modelo individual e o melhor modelo combinado (em termos de acurácia) e o índice z_{κ} entre as duas estratégias de combinação. A acurácia média e o índice *kappa* médio são calculados, em porcentagem, somando as matrizes de confusão das quatro execuções da metodologia apresentada na Figura 25 e as dividindo por quatro. E o indicador z_{κ} é calculado conforme descrito na Seção 2.4. A acurácia média indica se o modelo combinado é capaz de acertar mais exemplos que os modelos individuais. O índice κ indica se os modelos (individuais e combinados) apresentam um nível de classificação significativo. E o índice z_{κ} indica se o modelo combinado possui uma diferença significativa entre os modelos individuais.

Na abordagem multi-classe, são calculadas, em porcentagem, a acurácia de classificação média, o índice *kappa* médio e a precisão média dos modelos combinados. As métricas descritas são utilizadas para comparar o desempenho das duas metodologias multi-classe. Além disso, é utilizado como comparação o resultado de dois artigos que desenvolvem modelos de ICM para a classificação da imagética motora utilizando a base de dados descrita na Seção 3.1.

O primeiro artigo selecionado para comparação é o de (WEI; DONG; ZHU, 2021), onde é desenvolvida uma ICM baseada no algoritmo *Bagging*. O modelo proposto na pesquisa utiliza a filtragem por sub-banda e ajusta diferentes classificadores para então combiná-los utilizando a Votação por Maioria. Os autores da pesquisa apontam que o modelo combinado possui melhor desempenho que todos os modelos individuais combinados. O artigo é utilizado como comparação pois apresenta uma metodologia distinta de aplicação do algoritmo *Bagging*.

O segundo artigo selecionado para comparação de desempenho é o de (BARACHANT et al., 2012), onde um único modelo é ajustado para a classificação multi-classe utilizando a estratégia "Um versus Um". A pesquisa avalia o método de pré-processamento dos sinais de EEG baseado na geometria de Riemann (CONGEDO; BARACHANT; BHATIA, 2017) como alternativa à filtragem espacial utilizando os filtros CSP. Os resultados apontam que a metodologia proposta pelos autores apresentam desempenho, em termos de acurácia, superior aos modelos que utilizam o CSP. O artigo é utilizado como comparação pois apresenta um método alternativo de classificação multi-classe, sem a combinação de modelos ajustados individualmente.

4 RESULTADOS E DISCUSSÕES

Este Capítulo apresenta os resultados obtidos a partir dos experimentos descritos no Capítulo 3. Os resultados apresentados são divididos em duas seções, uma para a abordagem binária e outra para a abordagem multi-classe. Os resultados são avaliados e discutidos de acordo com as métricas descritas na Seção 2.4. Para a abordagem multi-classe, o resultado de dois artigos são utilizados para comparação.

4.1 Resultados da aplicação do Bagging - Abordagem Binária

As seções a seguir apresentam o resultado da aplicação do *Bagging* para a abordagem binária na distinção de cada um dos pares de classes contidos na base de dados utilizada. Os 9 sujeitos avaliados, contidos na base de dados descrita na seção 3.1, são representados nas tabelas pela numeração de 1 a 9. Os modelos combinados para cada sujeito, descritos no Apendice A são enumerados de 1 a 3. As siglas ME, MD, PE e LG referem-se, respectivamente, às classes mão esquerda, mão direita, pés e língua.

As siglas VM e SV fazem referência aos algoritmos de Votação por Maioria e *Soft Voting*. A coluna "SV x Modelo" refere-se ao cálculo do z_{κ} entre o modelo combinado por *Soft Voting* e o modelo individual com maior acurácia média. O *soft Voting* foi escolhido como estratégia de comparação entre os modelos individuais pois foi verificado, por meio do z_{κ} , que não existe diferença estatística entre as duas estratégias de combinação de classificação avaliadas para todos os casos estudados. Na parte inferior das tabelas, o símbolo μ representa a média das acurácias e índices *kappa* dos modelos individuais e combinados para todos os sujeitos. O indicador σ representa o desvio médio padrão para todos os sujeitos e métricas supracitadas.

4.1.1 *Bagging* para as classes ME x MD

A Tabela 3 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho na combinação dos modelos para os sujeitos 1, 4, 5, 6, 7, 8 e 9, indicando que os modelos individuais erram exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Sendo o *Soft Voting* a melhor estratégia de combinação dos modelos para a maioria dos sujeitos. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos individuais e a metodologia proposta.

ME x MD									
		Acu	rácia (%	6)	Ka	ppa (%)	Zκ	
Suj	Modelo	Modelo	Bag	ging	Modolo	Bag	ging	SV v Modolo	
		WIUUCIU	VM	SV	Modelo	VM	SV	SV A MOUCIO	
	1	86,46			72,92				
1	2	72,22	86,46	87,15	44,44	72,92	74,31	0,12	
	3	80,21			60,42				
	1	52,78			5,56				
2	2	56,6	56,25	56,6	13,19	12,5	13,19	0	
	3	52,78			5,56				
	1	96,53			93,06				
3	2	92,01	96,18	96,18	84,03	92,36	92,36	0,11	
	3	95,14			90,28				
	1	67,36		34,72					
4	2	65,28	71,18	3 71,88	30,56	42,36	43,75	0,59	
	3	64,58			29,17				
	1	80,9			61,81				
5	2	82,29	88,89	89,24	64,58	77,78	78,47	1,03	
	3	83,33			66,67				
	1	56,94			13,89				
6	2	62,15	64,58	65,28	24,31	29,17	30,56	0,39	
	3	62,15			24,31			,	
	1	83,33			66,67				
7	2	80,21	90,97	90,62	60,42	81,94	81,25	0,66	
	3	87,15			74,31			.,	
	1	92,36			84,72				
8	2	95,49	95,49	95,49	90,97	90,97	90,97	0	
	3	93,75			87,5				
	1	94,1			88,19				
9	2	93,75	94,79	94,1	87,5	89,58	88,19	0	
	3	91,67			83,33				
	μ	82,29	88,89	89,24	64,58	77,78	78,47		
	σ	14,80	14,89	14,58	29,59	29,77	29,17		

Tabela 3 - Resultados da aplicação do Bagging para as classes ME x MD.

Fonte: O autor.

4.1.2 *Bagging* para as classes ME x PE

A Tabela 4 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho na combinação dos modelos para os sujeitos 2, 4, 5, 6, 7, 8 e 9, indicando que os modelos individuais erram exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Nesse caso, o *Soft Voting* e a Votação por Maioria apresentam desempenho semelhante. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos

individuais e a metodologia proposta.

	ME x PE									
		Acu	rácia (%	6)	Ka	ppa (%)	Zκ		
Suj	Modelo	Modelo	Bag	ging	Modelo	Bag	ging	SV x Modelo		
	1	0 (00	VIVI	SV	02.75	VIVI	SV			
	1	96,88	0	0= 00	93,75	04.44	04.44	0		
I	2	95,14	97,22	97,22	90,28	94,44	94,44	0		
	3	97,22			94,44					
	l	80,56			61,11	<i></i>		0.40		
2	2	80,21	82,29	83,33	60,42	64,58	66,67	0,43		
	3	79,17			58,33					
	1	96,53		95,83	93,06					
3	2	90,28	96,53		80,56	93,06	91,67	0,22		
	3	89,93			79,86					
	1	75	86,81 87,5		50		75	0,77		
4	2	82,99		87,5	65,97	73,61				
	3	82,29			64,58					
	1	78,12			56,25	63,19	63,19	0,52		
5	2	73,61	81,6	81,6	47,22					
	3	66,67			33,33					
	1	67,36			34,72		53,47	1,26		
6	2	71,88	76,04	76,74	43,75	52,08				
	3	65,62			31,25					
	1	98,96			97,92					
7	2	99,31	100	100	98,61	100	100	0,71		
	3	98,96	-		97,92			,		
	1	88,54			77,08					
8	2	87,5	90,62	91,32	75	81,25	82,64	0,55		
	3	82,64			65,28					
	1	94,44			88,89					
9	2	94,1	95,49	49 94,44	88,19	90,97	88,89	0		
	3	93,75			87,5		30,09			
	μ	87,5	90,62	91,32	75	81,25	82,64			
	σ	10,74	8,36	7,94	21,48	16,72	15,88			

Tabela 4 – Resultados da aplicação do Bagging para as classes ME x PE.

Fonte: O autor.

4.1.3 Bagging para as classes ME x LG

A Tabela 5 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho na combinação dos modelos para os sujeitos 1, 2, 3, 4, 5, 7, 8 e 9, indicando que os modelos individuais erram exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Nesse caso, o *Soft Voting* e a Votação por Maioria apresentam desempenho

semelhante. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos individuais e a metodologia proposta.

ME x LG									
		Acu	rácia (%	6)	Ka	ppa (%)	Zκ	
Suj	Modelo	Modelo	Bag	ging	Modelo	Bagging		SV x Modelo	
		moucio	VM	SV	Modelo	VM	SV	by a modelo	
	1	97,57			95,14				
1	2	96,88	97,57	97,92	93,75	95,14	95,83	0,14	
	3	95,49			90,97				
	1	65,97			31,94				
2	2	68,4	73,61	72,92	36,81	47,22	45,83	0,6	
	3	67,71			35,42				
	1	96,18			92,36				
3	2	96,53	97,57	97,57	93,06	95,14	95,14	0,37	
	3	93,75	ŕ		87,5				
	1	82,99	85,42	42 85,07 57,64					
4	2	78,82			57,64	70,83	70,14	0,34	
	3	81,6			63,19				
	1	72,22			44,44	61,81	60,42	0,56	
5	2	75,69	80,9	80,21	51,39				
	3	76,39			52,78				
	1	62,85			25,69	39,58	37,5	0,46	
6	2	70,14	69,79	68,75	40,28				
	3	61,11			22,22				
	1	98,61			97,22				
7	2	97,57	98,96	99,31	95,14	97,92	98,61	0,41	
	3	96,88			93,75				
	1	95,14			90,28				
8	2	90,62	95,83	95,83	81,25	91,67	91,67	0,2	
	3	92,71			85,42				
	1	96,53			93,06				
9	2	97,57	97,57	97,57	95,14	95,14	95,14	0,29	
	3	98,26			96,53				
	μ	92,71	95,83	95,83	85,42	91,67	91,67		
	σ	13,13	11,45	11,92	26,26	22,90	23,84		

Tabela 5 - Resultados da aplicação do Bagging para as classes ME x LG.

Fonte: O autor.

4.1.4 Bagging para as classes MD x PE

A Tabela 6 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho na combinação dos modelos para todos os sujeitos, indicando que os modelos individuais erram

exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Nesse caso, o *Soft Voting* e a Votação por Maioria apresentam desempenho semelhante. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos individuais e a metodologia proposta.

MD x PE									
		Acu	rácia (%	6)	Ka	ppa (%)	Ζĸ	
Suj	Modelo	Madala	Bag	ging	Madala	Bag	ging		
		wioueio	VM	SV	wioueio	VM	SV	S V X MOUEIO	
	1	95,83			91,67				
1	2	97,92	98,26	98,26	95,83	96,53	96,53	0,15	
	3	96,53			93,06				
	1	80,56			61,11				
2	2	82,64	87,15	87,85	65,28	74,31	75,69	0,83	
	3	82,99			65,97				
	1	96,18			92,36				
3	2	95,83	95,83	96,53	91,67	91,67	93,06	0,11	
	3	80,21	,		60,42				
	1	81,6			63,19		75	0,6	
4	2	84,03	88,19	87,5	68,06	76,39			
	3	77,08			54,17				
	1	77,08	80,9		54,17	61,81	62,5	0,62	
5	2	75,69		81,25	51,39				
	3	70,14			40,28				
	1	68,75			37,5				
6	2	70,83	77,08	77,08	41,67	54,17	54,17	0,81	
	3	71,18			42,36				
	1	99,65			99,31				
7	2	99,31	100	100	98,61	100	100	0,5	
	3	99,31			98,61				
	1	88,89			77,78				
8	2	88,19	88,89	89,58	76,39	77,78	79,17	0,13	
	3	74,31			48,61				
	1	84,38			68,75				
9	2	82,29	87,5	88,19	64,58	75	76,39	0,67	
	3	84,03			68,06				
	μ	82,99	88,19	88,19	65,97	76,39	76,39		
	σ	9,97	7,64	7,64	19,93	15,27	15,28		

4.1.5 Bagging para as classes MD x LG

A Tabela 7 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho

Fonte: O autor.

na combinação dos modelos para os sujeitos 2, 3, 4, 5, 6, 7 e 9, indicando que os modelos individuais erram exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Nesse caso, o *Soft Voting* apresentou desempenho ligeiramente superior. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos individuais e a metodologia proposta.

	MD x LG									
		Acu	rácia (%	6)	Ka	ppa (%)	Zκ		
Suj	Modelo	Modelo	Bag	ging	Modelo	Bag	ging	SV x Modelo		
			VM	SV	0.0.64	VM	SV			
	1	99,31			98,61					
1	2	99,65	99,65	99,65	99,31	99,31	99,31	0		
	3	99,65			99,31					
	1	68,75			37,5					
2	2	66,67	72,22	72,57	33,33	44,44	45,14	0,5		
	3	67,71			35,42					
	1	96,88		97,57	93,75					
3	2	94,79	96,88		89,58	93,75	95,14	0,13		
	3	97,22			94,44					
	1	74,31		81,94 82,64	48,61		65,28	0,48		
4	2	79,51	81,94		59,03	63,89				
	3	75,69			51,39					
	1	84,72		9,24 89,58	69,44	78,47				
5	2	85,07	89,24		70,14		79,17	0,82		
	3	82,64			65,28					
	1	61.81			23.61		34.72	0,04		
6	2	67.01	66.67	67.36	34.03	33.33				
	3	56,25	,		12,5	ŕ	,			
	1	98.61			97.22					
7	2	97.92	98.61	98.61	95.83	97.22	97.22	0		
	3	97.92		,	95.83					
	1	88.89			77.78					
8	2	88.54	90.28	90.28	77.08	80.56	80.56	0.14		
	3	90.97	,20,20	,	81 94	00,00	00,00	0,11		
	1	92.71			85.42					
9	2	90.62	92.71	93,4	81.25	85.42	86.81	0.16		
	3	90,02	, , , , , , , , , , , , , , , , , , , ,		80.56	05,42	00,01	0,10		
	11	88.80	90.28	90.28	77 78	80.56	80.56			
	σ	13.22	11 70	11 56	26.44	23 10	23 13			
	0	13,22	11,70	11,50	20,44	25,40	23,13			

Tabela 7 –	Resultados	la aplicação	do <i>Bagging</i> pa	ara as classes	MD x LG.
Iabela /	Resultauos	ia apiicaçao	uo Dussins pu	i a as classes	

4.1.6 *Bagging* para as classes PE x LG

A Tabela 8 apresenta o desempenho dos modelos individuais e combinados após a aplicação do *Bagging*. Em termos da acurácia média, é possível observar que houve ganho na combinação dos modelos para os sujeitos 1, 3, 4, 5, 6, 8 e 9, indicando que os modelos individuais erram exemplos diferentes e que a combinação dos classificadores é capaz de atenuar os erros individuais. Nesse caso, o *Soft Voting* apresentou desempenho superior como estratégia de combinação. Entretanto, ao avaliar o índice z_{κ} entre o *soft voting* e o melhor modelo individual, é possível perceber que, para todos os sujeitos, não há diferença significativa entre os modelos individuais e a metodologia proposta.

	PE x LG									
		Acu	rácia (%	6)	Ka	ppa (%)	Zκ		
Suj	Modelo	Modele	Bag	ging	Madala	Bagging		SV v Modelo		
		Modelo	VM	SV	widueio	VM	SV	S V X MOUEIO		
	1	71,88			43,75					
1	2	68,06	73,26	73,61	36,11	46,53	47,22	0,23		
	3	67,36			34,72					
	1	85,42			70,83					
2	2	81,25	85,76	85,76	62,5	71,53	71,53	0,06		
	3	78,47			56,94					
	1	75,35			50,69					
3	2	76,04	82,29	83,33	52,08	64,58	66,67	0,64		
	3	79,17			58,33					
	1	72,57		56 82,29	45,14		64,58	1,17		
4	2	73,61	80,56		47,22	61,11				
	3	74,31			48,61					
	1	67,01			34,03	41,67	44,44	0,28		
5	2	59,03	70,83	72,22	18,06					
	3	70,14			40,28					
	1	74,65			49,31					
6	2	73,96	75	75,69	47,92	50	51,39	0,14		
	3	59,03			18,06					
	1	88,89			77,78					
7	2	77,08	86,11	87,5	54,17	72,22	75	0,26		
	3	81,94			63,89					
	1	87,15			74,31					
8	2	86,81	88,89	90,97	73,61	77,78	81,94	0,74		
	3	82,29			64,58					
	1	88,89			77,78					
9	2	88,19	90,97	91,67	76,39	81,94	83,33	0,15		
	3	90,97			81,94					
	μ	76,04	82,29	83,33	52,08	64,58	66,67			
	σ	8,77	7,15	7,27	17,54	14,30	14,55			

Tabela 8 - Resultados da aplicação do Bagging para as classes PE x LG.

4.2 Resultados da Aplicação do *Bagging* e *Stacking* - Abordagem Multi-classe

A seguir são apresentados os resultados para a aplicação do *Bagging* e *Stacking* na abordagem multi-classe, descritos nas Seções 3.3.2 e 3.3.3, respectivamente. A Tabela 9 apresenta as acurácias de classificação e índices *kappa* médios dos dois modelos propostos. As Tabelas 10 e 11 apresentam a precisão média dos modelos para cada classe da base de dados. A Tabela 12 apresenta a comparação de acurácia de classificação entre os modelos propostos e os dois artigos selecionados.

Ao analisar a Tabela 9, é possível observar que, para a base de dados avaliada, o *Bagging* apresenta melhor desempenho de classificação em termos de acurácia, com uma média (μ) de 74,86%. O sujeito 7 apresenta a maior acurácia para os dois métodos propostos, com 89,41%. Com relação ao índice *kappa*, o *Bagging* também apresenta melhor desempenho, sendo os sujeitos 2 e 6 os piores para ambos modelos.

	Stacking x Bagging multi-classe										
Sujoito	Acurá	cia (%)	Kappa (%)								
Sujeno	Stacking	Bagging	Stacking	Bagging							
1	79,86	80,38	73,15	73,84							
2	56,6	56,42	42,13	41,9							
3	86,98	87,5	82,64	83,33							
4	66,32	69,44	55,09	59,26							
5	67,53	68,06	56,71	57,41							
6	48,78	52,26	31,71	36,34							
7	89,41	89,41	85,88	85,88							
8	84,2	84,2	78,94	78,94							
9	85,07	86,11	80,09	81,48							
$\mu \pm \sigma$	$73,86 \pm 14,60$	74,86 ± 13,88	$65,15 \pm 19,47$	66,49 ± 18,51							

Tabela 9 – Acurácia de classificação e índice kappa dos modelos Stacking e Bagging Multi-classe.

Fonte: O autor.

Ao analisar os resultados da Tabela 10, é possível notar que o *Bagging* apresenta, em média, precisão de classificação superior a 70% para todas as classes da base de dados avaliada. Os resultados sugerem que o modelo proposto é capaz de distinguir de maneira eficiente os padrões das quatro classes de imagética motora dos usuários. O sujeito 3 apresenta os maiores valores de precisão para as classes ME e MD, porém, o sujeito 7 apresenta uma maior média de precisão para as quatro classes.

Bagging Multi-Classe										
Sujoito	Precisão (%)									
Sujeno	ME	MD	PE	LG						
1	83,33	90,28	68,06	79,86						
2	39,58	43,06	79,17	63,89						
3	94,44	95,83	79,17	80,56						
4	74,31	61,81	76,39	65,28						
5	69,44	79,86	57,64	65,28						
6	54,86	45,83	61,81	46,53						
7	89,58	93,06	90,28	84,72						
8	86,11	84,03	79,86	86,81						
9	91,67	82,64	75	95,14						
$\mu\pm\sigma$	$75,92 \pm 18,52$	$75,16 \pm 20,01$	$74,15 \pm 10,07$	$74,23 \pm 15,05$						

Tabela 10 – Desempenho de precisão do modelo bagging multi-classe.

Fonte: O autor.

A Tabela 11 evidencia que o *Stacking* também apresenta, em média, precisão de classificação acima de 70% para as quatro classes avaliadas. Os sujeitos 3 e 7 também apresentam os melhores desempenhos entre os sujeitos. Os resultados sugerem que o *Bagging* possui desempenho superior ao *Stacking* na distinção das quatro classes de imagética motora da base de dados avaliada.

Stacking Multi-classe									
Sujeito	Precisão								
	ME	MD	PE	LG					
1	86,11	87,5	71,53	74,31					
2	44,44	45,14	81,25	55,56					
3	93,06	93,75	75,69	85,42					
4	68,06	56,94	70,83	69,44					
5	70,14	76,39	59,72	63,89					
6	52,78	40,97	54,86	46,53					
7	91,67	91,67	87,5	86,81					
8	88,19	83,33	79,86	85,42					
9	91,67	79,86	78,47	90,28					
$\mu\pm\sigma$	$76,24 \pm 18,25$	$72,84 \pm 20,05$	$73,10 \pm 10,45$	$73,07 \pm 15,41$					

Tabela 11 – Desempenho de precisão do modelo Stacking multi-classe.

Fonte: O autor.

A Tabela 12 apresenta a acurácia de classificação dos dois modelos propostos nesta pesquisa e os resultados de acurácia dos artigos selecionados para comparação. Como esperado, não há uma abordagem que é melhor para todos os sujeitos avaliados devido as características únicas de cada sujeito. No entanto, o *Bagging* proposto nesta pesquisa supera os resultados das pesquisas de (WEI; DONG; ZHU, 2021) e (BARACHANT et al., 2012) para seis sujeitos (3,4,5,7,8 e 9). Ao analisar a média das acurácias (μ) de todos os sujeitos e o desvio médio

padrão (σ), é possível notar que o *Bagging* apresenta a maior média de acurácia, com 74,9% e o segundo menor desvio padrão, com 13,9%, superando as demais metodologias avaliadas. Além disso, os menores valores de acurácia apresentados pelas abordagens propostas nesta pesquisa, 48% do *Stacking* e 52% do *Bagging*, superam os menores valores de acurácia apresentados pelos artigos de (WEI; DONG; ZHU, 2021) e (BARACHANT et al., 2012), sendo 44,4% e 45%, respectivamente.

	Acurácia (%)							
Sujeito	Stacking	Bagging	(WEI; DONG; ZHU, 2021)	(BARACHANT et al., 2012)				
1	79,9	80,4	81,9	80,5				
2	56,6	56,4	58,0	51,3				
3	87,0	87,5	76,7	87,5				
4	66,3	69,4	64,2	59,3				
5	67,5	68,1	61,8	45				
6	48,8	52,3	44,4	55,3				
7	89,4	89,4	83,0	82,1				
8	84,2	84,2	76,7	84,8				
9	85,1	86,1	69,8	86,1				
$\mu \pm \sigma$	$73,86 \pm 14,60$	74,9 ± 13,9	$68,5 \pm 12,7$	$70,2 \pm 17,1$				

Tabela 12 – Comparação do desempenho de acurácia (%) entre os modelos de aprendizado em conjunto propostos e os resultados dos artigos selecionados.

5 CONCLUSÕES E PERSPECTIVAS FUTURAS

O desenvolvimento de uma ICM personalizada busca explorar as características particulares que a IM de cada sujeito possui para produzir resultados de classificação mais eficientes. Enquanto o aprendizado em conjunto tem como princípio básico combinar os resultados de modelos treinados de forma independente para reduzir a variância do modelo final e, assim, melhorar seu desempenho. Assim, este trabalho investiga se a combinação de ICMs personalizadas é capaz de melhorar o desempenho de classificação da imagética motora de forma significativa.

A aplicação dos algoritmos de aprendizado em conjunto é realizada em duas abordagens: binária e multi-classe. Na abordagem binária, os resultados são obtidos em termos de acurácia de classificação e índice z_{κ} para avaliar se a metodologia proposta apresenta diferença significativa em relação aos modelos individuais personalizados. Na abordagem multi-classe, os resultados são obtidos em termos de acurácia de classificação, índice *Kappa* e precisão, além disso, os resultados de acurácia são comparados com dois artigos que desenvolvem ICMs para a classificação multiclasse.

Ao analisar os resultados obtidos com a aplicação do *Bagging* no conjunto de dados 2a da IV competição de ICM é possível perceber que a metodologia aplicada na abordagem binária, embora apresente ganho em relação à acurácia média, estatisticamente não apresenta diferença significativa com relação aos modelos personalizados otimizados. Também não há diferença de desempenho entre as estratégias de combinação avaliadas, Votação por Maioria e *Soft Voting*. Portanto, o desempenho apresentado não justifica a utilização da metodologia para a classificação da imagética motora na distinção de apenas duas classes.

Os resultados sugerem que o critério escolhido para a seleção dos modelos otimizados deve ser reavaliado, levando em consideração o índice z_{κ} entre os modelos selecionados. A estratégia mencionada deve garantir que o modelo final combinado obtenha uma diferença estatística maior com relação aos modelos individuais. Também é pertinente reavaliar a divisão do conjunto de dados utilizado para o ajuste dos modelos, utilizando estratégias de reamostragem com reposição, para aumentar a variabilidade dos conjuntos.

Os resultados obtidos na aplicação do *Bagging* e *Stacking* na abordagem multi-classe sugerem que a combinação de modelos binários otimizados é capaz de produzir um bom desempenho de classificação, em termos de acurácia e precisão. Os experimentos expõem que a combinação dos modelos é capaz de aproveitar a especialidade que cada modelo binário possui na distinção de duas classes para realizar a distinção das quatro classes avaliadas. O *Bagging* apresenta a melhor acurácia de classificação superando, inclusive, os dois artigos selecionados para comparação.

O desempenho produzido pelo *Stacking* sugere que é necessário avaliar os parâmetros de ajuste do classificador final MLP, uma hipótese é utilizar a otimização bayesiana para ajustar

o classificador final. A metodologia aplicada na abordagem multi-classe expõe ainda que é pertinente investigar a adição de novos modelos individuais para a combinação.

Enfim, futuras pesquisas podm explorar a metodologia exposta neste trabalho para outras base de dados públicas. Também é importante avaliar a aplicação de outros algoritmos de aprendizado em conjunto, além de estudar a adição de novos modelos ajustados com otimização.

REFERÊNCIAS

ABIRI, R. et al. A comprehensive review of eeg-based brain-computer interface paradigms. **Journal of Neural Engineering**, v. 16, 11 2018.

ALFEO, A. L. et al. Recognizing motor imagery tasks from eeg oscillations through a novel ensemble-based neural network architecture. In: **2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)**. [S.l.: s.n.], 2021. p. 5983–5986.

BARACHANT, A. et al. Multiclass brain–computer interface classification by riemannian geometry. **IEEE Transactions on Biomedical Engineering**, v. 59, n. 4, p. 920–928, 2012.

BARRERA, E. Brain functions to help improve our body. 2016. Disponível em: https://gravitywerks.com/brain-functions-to-help-improve-our-body/. Acesso em: 23 fev. 2020.

BASHASHATI, H.; WARD, R. K.; BASHASHATI, A. Bayesian optimization of bci parameters. In: **2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)**. [S.l.: s.n.], 2016. p. 1–5.

BHUVANA, R.; MAHESHWARI, S.; SASIKALA, S. Predict the heart disease using a logistic regression classifier algorithm. In: **2023 12th International Conference on System Modeling Advancement in Research Trends (SMART)**. [S.1.: s.n.], 2023. p. 649–652.

BRUNNER, C. et al. BCI competition 2008 – graz data set a. **Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces)**, Graz University of Technology, v. 16, 2008. Disponível em: http://www.bbci.de/competition/iv/desc_2a.pdf>. Acesso em: 14 mar. 2020.

CHEN, Y. Y. et al. Mu oscillations and motor imagery performance: A reflection of intra-individual success, not inter-individual ability. **Human Movement Science**, v. 78, p. 102819, 2021. ISSN 0167-9457. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167945721000671.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. **IEEE Access**, v. 9, p. 78368–78381, 2021.

CONGEDO, M.; BARACHANT, A.; BHATIA, R. Riemannian geometry for eeg-based brain-computer interfaces; a primer and a review. **Brain-Computer Interfaces**, v. 4, p. 1–20, 03 2017.

DAULTON, S. et al. Unexpected improvements to expected improvement for bayesian optimization. In: **Proceedings of the 37th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2024. (NIPS '23).

DOLZHIKOVA, I. et al. An ensemble cnn for subject-independent classification of motor imagery-based eeg. In: **2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)**. [S.l.: s.n.], 2021. p. 319–324.

ESQUIVEL, J. Z.; VARGAS, J. A. C.; LOPEZ-MEYER, P. Fractional adaptation of activation functions in neural networks. In: **2020 25th International Conference on Pattern Recognition** (**ICPR**). [S.l.: s.n.], 2021. p. 7544–7550.

GRAMFORT, A. et al. MEG and EEG data analysis with MNE-Python. Frontiers in Neuroscience, v. 7, n. 267, p. 1–13, 2013.

HALTAŞ, K.; ERGÜZEN, A.; ERDAL, E. Classification methods in eeg based motor imagery bci systems. In: **2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)**. [S.l.: s.n.], 2019. p. 1–5.

KAMHI, S. et al. Multi-classification of motor imagery eeg signals using bayesian optimization-based average ensemble approach. **Applied Sciences (Switzerland)**, v. 12, n. 12, 2022.

KANG, J.-S.; KAVURI, S.; LEE, M. Adaptive eeg noise filtering for coherence analysis. In: **2014 International Winter Workshop on Brain-Computer Interface (BCI)**. [S.l.: s.n.], 2014. p. 1–4.

KUMAR, J. S.; BHUVANESWARI, P. Analysis of electroencephalography (eeg) signals and its categorization–a study. **Procedia Engineering**, v. 38, p. 2525–2536, 2012. ISSN 1877-7058. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING. Disponível em: https://www.sciencedirect.com/science/article/pii/S1877705812022114>.

LINDIG, C.; BOUGRAIN, L. Comparison of sensorimotor rhythms in eeg signals during simple and combined motor imageries over the contra and ipsilateral hemispheres. In: . [S.l.: s.n.], 2015. v. 2015, p. 3953–3956.

LLANOS, C. et al. Mu-rhythm changes during the planning of motor and motor imagery actions. **Neuropsychologia**, v. 51, n. 6, p. 1019–1026, 2013. ISSN 0028-3932. Disponível em: https://www.sciencedirect.com/science/article/pii/S002839321300050X>.

LOPES, T. de J. Bacharelado em Engenharia Elétrica, **Estudo, Implementação e Comparação de Diferentes Tipos de Pré-Processamento Aplicados a um Sistema interface Cérebro-Computador**. Tucuruí: [s.n.], 2019. 71 f.

LOTTE, F. A tutorial on EEG signal-processing techniques for mental state recognition in brain–computer interfaces. In: **Guide to Brain-Computer Music Interfacing**. Springer, London, 2014. p. 133–161. ISBN 978-1-4471-6583-5. Disponível em: https://hal.inria.fr/hal-01055103. Acesso em: 11 out. 2019.

LOTTE, F. et al. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. **Journal of Neural Engineering**, IOP Publishing, v. 15, n. 3, p. 031005, apr 2018. Disponível em: https://dx.doi.org/10.1088/1741-2552/aab2f2>.

LOTZE, M.; HALSBAND, U. Motor imagery. Journal of physiology, Paris, v. 99, p. 386–95, 07 2006.

MANKAR, N. V.; KHOBRAGADE, A.; RAGHUWANSHI, M. M. Classification of remote sensing image using svm kernels. In: **2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)**. [S.l.: s.n.], 2016. p. 1–5.

MIAO, Y. et al. Learning common time-frequency-spatial patterns for motor imagery classification. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 29, p. 699–707, 2021.

MITRA, P.; BIESSMANN, F. Automated computational energy minimization of ml algorithms using constrained bayesian optimization. 07 2024.

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.

NEGRI, R. Reconhecimento de Padrões: Um estudo dirigido. [S.l.: s.n.], 2021. ISBN 9786555061635.

NOVI, Q. et al. Sub-band common spatial pattern (sbcsp) for brain-computer interface. In: **2007 3rd International IEEE/EMBS Conference on Neural Engineering**. [S.l.: s.n.], 2007. p. 204–207.

PAPANASTASIOU, G. et al. Brain computer interface based applications for training and rehabilitation of students with neurodevelopmental disorders. a literature review. **Heliyon**, v. 6, n. 9, p. e04250, 2020. ISSN 2405-8440. Disponível em: https://www.sciencedirect.com/science/article/pii/S240584402031094X>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, v. 12, p. 2825–2830, 2011.

RAHIMI, M. et al. Ensemble methods combination for motor imagery tasks in brain computer interface. In: **2016 23rd Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME)**. [S.l.: s.n.], 2016. p. 336–340.

SOBREIRO, V. et al. Uma estimaÇÃo do valor da commodity de aÇÚcar utilizando redes neurais artificiais. 01 2008.

TANG, Z. et al. Single-trial classification of different movements on one arm based on erd/ers and corticomuscular coherence. **IEEE Access**, v. 7, p. 128185–128197, 2019.

TEPLAN, M. Fundamentals of EEG measurement. **Measurement science review**, v. 2, n. 2, p. 1–11, Jan. 2002. ISSN 1335-8871. Disponível em: https://www.researchgate.net/publication/228599963_Fundamental_of_EEG_Measurement>. Acesso em: 26 jun. 2019.

VILAS-BOAS, V.; JORGE, V.; SILVA, C. D. Towards ideal time window for classifying motor imagery in brain-computer interfaces. In: **Anais do VIII Symposium on Knowledge Discovery, Mining and Learning**. Porto Alegre, RS, Brasil: SBC, 2020. p. 73–80. ISSN 2763-8944. Disponível em: https://sol.sbc.org.br/index.php/kdmile/article/view/11961>.

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>.

WATANABE, S. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. 2023.

WOLPAW jonathan R.; WOLPAW, E. W. **Brain-Computer Interfaces**: Principles and pratice. [S.l.]: Oxford University Press, 2012.

A MODELOS DE ICMS-IM GERADOS

ME x MD							
		Parâmetros de Pré-processamento					
Sujeito	Modelo	fl	fh	t _{min}	t _{máx}	ncsp	nband
	1	11	49	0,5	2,5	2	18
1	2	10	50	0	2	4	19
	3	3	30	1	3	22	11
	1	2	26	1	3	10	12
2	2	5	36	0,5	2,5	6	15
	3	5	45	1,5	3,5	2	18
	1	0	29	0,5	2,5	4	6
3	2	4	32	1	3	8	2
	3	3	33	0	2	4	17
	1	6	39	1,5	3,5	22	18
4	2	5	36	1	3	4	5
	3	2	45	2	4	10	7
	1	1	36	0,5	2,5	8	22
5	2	4	47	0	2	2	20
	3	12	44	2	4	2	7
	1	10	30	0	2	6	9
6	2	3	35	0,5	2,5	4	5
	3	6	35	1	3	4	7
	1	5	30	1	2	2	5
7	2	1	28	0,5	2,5	22	8
	3	1	29	1,5	3,5	2	7
	1	2	39	1	3	8	24
8	2	1	39	0,5	2,5	2	6
	3	1	26	0	2	2	22
	1	3	37	0,5	2,5	2	8
9	2	6	34	0	2	2	3
	3	3	32	1	3	22	8

Tabela 13 – Configuração de hiperparâmetros de pré-processamento para cada modelo de ICM por sujeito para o par ME x MD.

Tabela 14 – Configuração de hiperparâmet	tros do classificador par	ra cada modelo de ICN	A por sujeito
para o par ME x MD.			

ME x MD							
Sujeito	Modelo	Classificador e Hiperparâmetros					
	1	{'C': -1.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}					
1	2	{'model': 'LDA'}					
	3	{'model': 'KNN', 'neig': 2}					
	1	{'C': -2.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}					
2	2	{'model': 'KNN', 'neig': 17}					
	3	{'model': 'LDA'}					
	1	{'model': 'LR'}					
3	2	{'model': 'LR'}					
	3	{'model': 'LR'}					
	1	{'model': 'KNN', 'neig': 8}					
4	2	{'model': 'LDA'}					
	3	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 100.0}					
5	1	{'model': 'LR'}					
	2	{'model': 'LR'}					
	3	{'eta': -4.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}					
	1	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 100.0}					
6	2	{'model': 'KNN', 'neig': 2}					
	3	{'model': 'KNN', 'neig': 17}					
_	1	{'model': 'LR'}					
7	2	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}					
	3	{'model': 'LR'}					
	1	{'model': 'LR' }					
8	2	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 60.0}					
	3	{'model': 'KNN', 'neig': 12}					
6	1	{'model': 'LDA'}					
9	2	{'model': 'KNN', 'neig': 5}					
	3	{'model': 'LDA'}					

ME x PE							
		Pa	râme	etros d	e Pré-p	process	amento
Sujeito	Modelo	fl	fh	t _{min}	t _{máx}	ncsp	nband
	1	12	41	0,5	2,5	2	21
1	2	7	26	1	3	8	15
	3	3	39	0	2	2	20
	1	12	34	2	4	4	15
2	2	12	35	1	3	6	23
	3	14	37	1,5	3,5	4	21
	1	5	40	0,5	2,5	2	19
3	2	11	41	1	3	8	24
	3	2	37	1,5	3,5	8	12
	1	0	34	0	2	22	16
4	2	5	38	0,5	2,5	6	18
	3	2	33	1	3	6	12
	1	0	37	0	2	8	13
5	2	9	35	0,5	2,5	4	12
	3	5	47	1,5	3,5	2	15
	1	1	46	0,5	2,5	6	14
6	2	5	30	1	3	8	7
	3	12	34	1,5	3,5	8	7
	1	5	50	0,5	2,5	8	13
7	2	2	31	1	3	8	9
	3	1	39	0	2	6	12
	1	1	46	0,5	2,5	6	20
8	2	4	46	1	3	6	23
	3	10	49	1,5	3,5	8	12
	1	3	26	0,5	2,5	8	22
9	2	2	33	0	2	6	20
	3	8	28	1	3	6	16

Tabela 15 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par ME x PE.

Tabela 16 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par ME x PE.

ME x PE									
Sujeito	Modelo	Classificador e Hiperparâmetros							
	1	{'model': 'KNN', 'neig': 7}							
1	2	{'model': 'KNN', 'neig': 18}							
	3	{'model': 'KNN', 'neig': 5}							
	1	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 100.0}							
2	2	{'model': 'KNN', 'neig': 11}							
	3	{'model': 'KNN', 'neig': 5}							
	1	{'model': 'KNN', 'neig': 15}							
3	2	{'model': 'KNN', 'neig': 10}							
	3	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 100.0}							
	1	{'model': 'LDA'}							
4	2	{'model': 'LR'}							
	3	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 40.0}							
	1	{'model': 'LR'}							
5	2	{'model': 'KNN', 'neig': 17}							
	3	{'model': 'LDA'}							
	1	{'model': 'LDA'}							
6	2	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}							
	3	{'model': 'LDA'}							
	1	{'model': 'LR'}							
7	2	{'model': 'LR'}							
	3	{'C': -1.0, 'kernel': {'kf': 'rbf'}							
	1	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}							
8	2	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 100.0}							
	3	{'model': 'KNN', 'neig': 3}							
	1	{'C': -1.0, 'kernel': {'kf': 'rbf'}, 'model': 'SVM'}							
9	2	{'model': 'LR'}							
	3	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 60.0}							

ME x LG							
		Pa	râme	etros d	e Pré-p	process	amento
Sujeito	Modelo	fl	fh	t _{min}	t _{máx}	ncsp	nband
	1	2	34	0,5	2,5	10	16
1	2	0	41	0	2	22	7
	3	12	30	1,5	3,5	2	17
	1	6	28	1,5	3,5	8	8
2	2	7	33	1	3	8	3
	3	9	25	2	4	4	9
	1	9	35	0,5	2,5	8	14
3	2	9	42	1,5	3,5	2	11
	3	7	46	0	2	8	11
	1	5	26	1	3	4	2
4	2	14	25	0	2	4	1
	3	5	46	0,5	2,5	6	25
	1	4	41	1	3	2	16
5	2	10	35	0,5	2,5	4	21
	3	2	39	1,5	3,5	6	17
	1	9	27	1	3	2	21
6	2	9	28	1,5	3,5	8	3
	3	5	47	2	4	22	19
	1	9	44	1	3	8	6
7	2	12	46	0,5	2,5	22	5
	3	15	39	1,5	3,5	4	21
	1	2	38	1	3	22	15
8	2	2	36	0,5	2,5	8	19
	3	6	48	1,5	3,5	2	16
	1	0	50	0,5	2,5	8	22
9	2	3	46	0	2	22	11
	3	3	38	1	3	22	6

Tabela 17 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par ME x LG.

Tabela 18 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par ME x LG.

ME x LG							
Sujeito	Modelo	Classificador e Hiperparâmetros					
	1	{'model': 'KNN', 'neig': 6}					
1	2	{'model': 'KNN', 'neig': 11}					
	3	{'model': 'KNN', 'neig': 12}					
	1	{'model': 'LDA'}					
2	2	{'model': 'LDA'}					
	3	{'model': 'LDA'}					
	1	{'model': 'KNN', 'neig': 9}					
3	2	{'model': 'KNN', 'neig': 15}					
	3	{'model': 'KNN', 'neig': 11}					
	1	{'model': 'LDA'}					
4	2	{'model': 'LDA'}					
	3	{'model': 'KNN', 'neig': 4}					
	1	{'model': 'LR'}					
5	2	{'model': 'KNN', 'neig': 2}					
	3	{'model': 'KNN', 'neig': 7}					
	1	{'model': 'KNN', 'neig': 11}					
6	2	{'model': 'KNN', 'neig': 14}					
	3	{'model': 'LR'}					
	1	{'model': 'KNN', 'neig': 14}					
7	2	{'model': 'LR'}					
	3	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 60.0}					
	1	{'model': 'KNN', 'neig': 11}					
8	2	{'model': 'LDA'}					
	3	{'model': 'LR'}					
	1	{'model': 'LDA'}					
9	2	{'model': 'LDA'}					
	3	{'model': 'KNN', 'neig': 16}					
MD x PE							
---------	--------	---------------------------------	----	-------------------------	------------------	------	-------
		Parâmetros de Pré-processamento					
Sujeito	Modelo	fl	fh	t _{min}	t _{máx}	ncsp	nband
	1	12	25	1	3	6	20
1	2	5	25	0,5	2,5	2	22
	3	4	27	0	2	8	11
	1	11	45	1,5	3,5	6	6
2	2	10	45	2	4	4	7
	3	3	35	1	3	4	20
	1	6	46	0,5	2,5	6	8
3	2	4	40	0	2	4	22
	3	5	44	1	3	10	13
	1	7	28	1	3	4	4
4	2	12	46	0,5	2,5	8	8
	3	10	25	0	2	8	10
	1	8	41	1	3	2	13
5	2	0	40	0,5	2,5	4	4
	3	8	32	0	2	4	5
	1	1	35	0	2	6	10
6	2	5	34	0,5	2,5	6	12
	3	9	33	1	3	6	2
	1	9	47	0,5	2,5	8	8
7	2	9	29	1	3	8	7
	3	2	33	1,5	3,5	4	15
	1	0	50	2	4	10	17
8	2	1	45	0	2	4	4
	3	14	42	1,5	3,5	8	14
	1	2	34	0,5	2,5	22	23
9	2	3	35	0	2	8	15
	3	6	49	1	3	4	8

Tabela 19 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par MD x PE.

Tabela 20 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par MD x PE.

MD x PE									
Sujeito	Modelo	Classificador e Hiperparâmetros							
	1	{'model': 'KNN', 'neig': 4}							
1	2	{'model': 'KNN', 'neig': 19}							
	3	{'model': 'LDA'}							
	1	{'model': 'KNN', 'neig': 9}							
2	2	{'model': 'KNN', 'neig': 6}							
	3	{'model': 'LR'}							
	1	{'model': 'LR'}							
3	2	{'model': 'KNN', 'neig': 8}							
	3	{'eta': -4.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 100.0}							
	1	{'model': 'KNN', 'neig': 14}							
4	2	{'model': 'LDA'}							
	3	{'model': 'LDA'}							
5	1	{'model': 'LDA'}							
	2	{'model': 'LDA'}							
	3	{'model': 'LDA'}							
-	1	{'model': 'LR'}							
6	2	{'model': 'LR'}							
	3	{'model': 'LR'}							
_	1	{'model': 'KNN', 'neig': 15}							
7	2	{'model': 'KNN', 'neig': 3}							
	3	{'model': 'KNN', 'neig': 7}							
0	1	{'model': 'LR' }							
8	2	{'model': 'LR' }							
	3	{'model': 'LR' }							
0	1	{ 'model': 'KNN', 'neig': 18}							
9	2	{'model': 'KNN', 'neig': 14}							
	3	{'model': 'LR' }							

MD x LG							
Parâmetros de Pré-processan					amento		
Sujeito	Modelo	fl	fh	t _{min}	t _{máx}	ncsp	nband
	1	1	38	1,5	3,5	8	6
1	2	7	32	1	3	4	9
	3	1	42	0,5	2,5	6	11
	1	5	34	2	4	10	6
2	2	12	37	1	3	10	1
	3	9	35	1,5	3,5	7	7
	1	10	31	0,5	2,5	6	20
3	2	5	28	0	2	8	14
	3	2	36	1	3	8	22
	1	1	26	0	2	8	11
4	2	5	38	1	3	2	22
	3	6	25	0,5	2,5	8	12
5	1	1	46	0.5	2,5	6	24
	2	5	39	0	2	8	17
	3	0	33	1	3	6	8
	1	3	34	0,5	2,5	4	15
6	2	3	39	0	2	4	18
	3	5	41	1,5	3,5	6	17
	1	12	35	1	3	4	11
7	2	13	32	0,5	2,5	4	3
	3	4	43	1,5	3,5	10	14
8	1	11	35	0,5	2,5	22	11
	2	7	41	1	3	4	10
	3	1	28	1,5	3,5	8	1
	1	8	40	0,5	2,5	10	19
9	2	9	29	0	2	10	20
	3	7	49	1	3	6	14

Tabela 21 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par MD x LG.

Tabela 22 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par MD x LG.

MD x LG								
Sujeito	Modelo	Classificador e Hiperparâmetros						
	1	{'model': 'LR'}						
1	2	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 80.0}						
	3	{'model': 'KNN', 'neig': 18}						
	1	{'model': 'LR'}						
2	2	{'model': 'LDA'}						
	3	{'model': 'LDA'}						
	1	{'C': -1.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}						
3	2	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}						
	3	{'model': 'KNN', 'neig': 3}						
	1	{'model': 'LDA'}						
4	2	{'C': -1.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}						
	3	{'model': 'LDA'}						
_	1	{'model': 'LR'}						
5	2	{'model': 'KNN', 'neig': 13}						
	3	{'model': 'LDA'}						
-	1	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 40.0}						
6	2	{'model': 'KNN', 'neig': 19}						
	3	{'model': 'LDA'}						
_	1	{'model': 'LR'}						
7	2	{'model': 'LR' }						
	3	{'model': 'KNN', 'neig': 12}						
0	1	{'model': 'KNN', 'neig': 18}						
8	2	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 2, 'n_neurons': 80.0}						
	3	{'model': 'LR'}						
1 {'model': 'LR'}		{'model': 'LR' }						
9	2	{'model': 'KNN', 'neig': 14}						
	3	{'model': 'LR'}						

PE x LG							
		Parâmetros de Pré-processamento					imento
Sujeito	Modelo	fl	fh	t _{min}	tmax	ncsp	nband
	1	4	48	1	3	10	5
1	2	8	43	1,5	3,5	8	21
	3	7	50	2	4	8	5
	1	15	42	1	3	2	5
2	2	11	40	2	4	2	1
	3	15	45	1,5	3,5	10	4
	1	6	26	0,5	2,5	2	21
3	2	6	28	1	3	8	19
	3	4	32	0	2	6	15
	1	5	30	1	3	8	18
4	2	6	40	0,5	2,5	6	11
	3	2	41	1,5	3,5	4	10
	1	4	39	1,5	3,5	2	11
5	2	8	39	2	4	2	11
	3	0	40	1	3	4	13
	1	0	30	0	2	4	25
6	2	0	32	0,5	2,5	4	23
	3	5	31	1	3	2	24
	1	4	48	0	2	2	11
7	2	7	39	1	3	22	16
	3	10	46	0,5	2,5	4	15
	1	5	46	1	3	10	2
8	2	10	34	1,5	3,5	22	4
	3	4	36	0,5	2,5	4	4
	1	9	37	0,5	2,5	2	23
9	2	10	39	0	2	10	23
	3	8	41	1	3	8	10

Tabela 23 – Configuração de hiperparâmetros do classificador para cada modelo de ICM por sujeito para o par PE x LG.

Tabela 24 – Configuração de hiperparâmetros	do classificador para	i cada modelo de IC	CM por sujeito
para o par PE x LG.			

PE x LG						
Sujeito	Modelo	Classificador e Hiperparâmetros				
	1	{'model': 'KNN', 'neig': 13}				
1	2	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 40.0}				
	3	{'model': 'LR'}				
	1	{'model': 'LDA'}				
2	2	{'model': 'LDA'}				
	3	{'model': 'KNN', 'neig': 20}				
	1	{'model': 'LDA'}				
3	2	{'model': 'LDA'}				
	3	{'model': 'KNN', 'neig': 14}				
4	1	{'model': 'KNN', 'neig': 9}				
	2	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 100.0}				
	3	{'model': 'LR'}				
5	1	{'eta': -3.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 40.0}				
	2	{'model': 'KNN', 'neig': 8}				
	3	{'eta': -2.0, 'model': 'MLP', 'n_hidden': 1, 'n_neurons': 40.0}				
	1	{'model': 'KNN', 'neig': 8}				
6	2	{'model': 'KNN', 'neig': 4}				
	3	{'model': 'LDA'}				
	1	{'C': -0.0, 'kernel': {'kf': 'sigmoid'}, 'model': 'SVM'}				
7	2	{'model': 'KNN', 'neig': 5}				
	3	{'model': 'LR'}				
_	1	{'model': 'LDA'}				
8	2	{'model': 'KNN', 'neig': 11}				
	3	{'C': -1.0, 'kernel': {'kf': 'linear'}, 'model': 'SVM'}				
	1	{'C': -3.0, 'kernel': {'kf': 'linear'}, 'model': 'SVM'}				
9	2	{'model': 'KNN', 'neig': 16}				
	3	{'C': -1.0, 'kernel': {'kf': 'rbf'}, 'model': 'SVM'}				