

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Gabriel Souza Machado

**Análise do desempenho de técnicas de
processamento de sinais para uma interface
cérebro-computador baseada em imagética
motora**

João Monlevade – MG

2018

Gabriel Souza Machado

Análise do desempenho de técnicas de processamento de sinais para uma interface cérebro-computador baseada em imagética motora

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Prof.^a Dr.^a Sarah Negreiros de Carvalho Leite

Coorientador: Prof. Dr. Glauco Ferreira Gazel Yared

João Monlevade – MG

2018

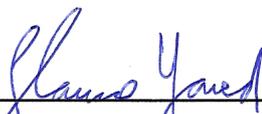
Ata de defesa

Aos 9 dias do mês de Julho de 2018, às 19 horas e 0 minutos, no Laboratório ProcSiMoS, foi realizada a defesa de Monografia pelo aluno Gabriel Souza Machado, orientado pela Prof.^a Dr.^a Sarah Negreiros de Carvalho Leite, sendo a Comissão Examinadora constituída pelos professores: Prof. Dr. Glauco Ferreira Gazel Yared, Prof. Dr. Harlei Miguel de Arruda Leite e Prof. Me. Fabricio Javier Erazo Costa.

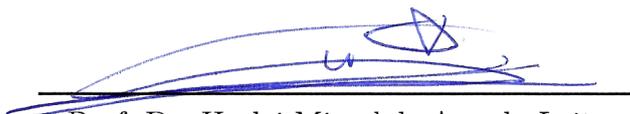
O candidato apresentou a monografia intitulada: “Análise do desempenho de técnicas de processamento de sinais para uma interface cérebro-computador baseada em imagética motora”. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 10, concedendo-lhe o prazo de 15 dias para incorporação no texto final das alterações sugeridas.

Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo formando.

João Monlevade, 12 de Julho de 2018.



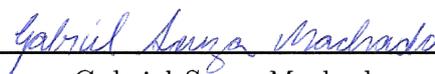
Prof. Dr. Glauco Ferreira Gazel Yared
Professor Coorientador/Presidente



Prof. Dr. Harlei Miguel de Arruda Leite
Professor Convidado



Prof. Me. Fabricio Javier Erazo Costa
Professor Convidado



Gabriel Souza Machado
Aluno

Resumo

Interfaces cérebro-computador são sistemas em *hardware* e *software* que permitem a seus usuários transmitir suas intenções a dispositivos externos com o envolvimento mínimo de nervos periféricos e músculos. Uma forma de se controlar essas interfaces é através de imagética motora, que consiste em se imaginar um movimento sem realizá-lo, provocando alterações em ritmos cerebrais que podem ser detectadas em sinais de eletroencefalograma (EEG). Visando identificar essas alterações, neste trabalho desenvolveu-se a etapa de processamento de sinais de uma interface cérebro-computador, compreendendo: pré-processamento, extração de atributos, seleção de atributos e classificação. O pré-processamento dos sinais foi realizado usando *Common Average Reference* (CAR), a extração de atributos através do periodograma de Welch, *Common Spatial Patterns* (CSP) e *Filter Bank Common Spatial Patterns* (FBCSP), a seleção de atributos baseada em informação mútua e a classificação com *Support Vector Machine* (SVM) e *Naïve Bayes*. O sistema foi avaliado usando uma base de dados contendo sinais de EEG de nove pessoas realizando imagética motora de quatro classes diferentes: mão esquerda, mão direita, pés e língua. A melhor combinação encontrada alcançou uma acurácia de 71.20% empregando CAR, FBCSP, informação mútua e *Naïve Bayes*. Os resultados indicam que a etapa de seleção de atributos é a que mais impacta a taxa de acerto, sendo que todas as melhores combinações de técnicas incluíram o FBCSP, e em todas as combinações o CSP superou a extração de atributos através do periodograma de Welch.

Palavras-chave: Interface cérebro-computador. Processamento de sinais. Imagética motora. Aprendizado de máquina.

Abstract

Brain-computer interfaces (BCI) are hardware and software systems that enable its users to convey their intentions to external devices without any involvement from the peripheral nerves and muscles. One way to control a BCI is through motor imagery, imagining a movement without actually performing it, causing changes in cerebral rhythms that can be detected through electroencephalography (EEG). Aiming to detect such changes, the signal processing stage of a BCI was implemented, encompassing: preprocessing, feature extraction, feature selection and classification. The preprocessing was done using Common Average Reference (CAR), the feature selection step employed Welch's method, Common Spatial Patterns (CSP) and Filter Bank Common Spatial Patterns (FCSP), feature selection was made using mutual information and the classification was done with Support Vector Machine (SVM) and Naive Bayes. These methods were evaluated using a dataset containing EEG signals from nine people performing motor imagery from four different classes. The best combination found achieved an accuracy of 71.20% using CAR, FBCSP, mutual information and Naive Bayes. The results indicate that feature selection is the most influential step in the system's performance, wherein all of the best performing combinations employed FBCSP, and all combinations using CSP outperformed Welch's method.

Keywords: Brain-computer interfaces. Signal processing. Motor imagery. Machine learning.

Lista de ilustrações

Figura 1 – Disposição dos eletrodos.	13
Figura 2 – Estrutura do ensaio.	14
Figura 3 – Etapas do processamento do sinal.	15
Figura 4 – Periodograma de Welch.	17
Figura 5 – Banco de filtros e filtragem espaciais do FBCSP.	19
Figura 6 – Banco de filtros Butterworth de 4 ^a ordem.	25
Figura 7 – Melhores resultados usando FBCSP.	27
Figura 8 – Melhores resultados usando CSP.	28
Figura 9 – Melhores resultados usando periodograma de Welch.	28
Figura 10 – Diagrama de fluxo da BCI com a configuração de melhor desempenho.	32

Lista de tabelas

Tabela 1 – Número de ensaios.	14
Tabela 2 – Métodos implementados	25
Tabela 3 – Impacto da técnica CAR sobre o desempenho do sistema.	28
Tabela 4 – Impacto da normalização dos atributos sobre o desempenho do sistema.	29
Tabela 5 – Impacto da informação mútua sobre o desempenho do sistema.	30
Tabela 6 – Melhores resultados por participante.	31
Tabela 7 – Acurácia em função de k	31
Tabela 8 – Comparação com os resultados da competição.	32

Lista de abreviaturas e siglas

BCI	<i>Brain-Computer Interface</i>
CSP	<i>Common Spatial Patterns</i>
DFT	<i>Discrete Fourier Transform</i>
EEG	Eletroencefalograma
ERD	<i>Event-Related-Desynchronization</i>
ERS	<i>Event-Related-Synchronization</i>
IM	Informação Mútua
FBCSP	<i>Filter Bank Common Spatial Patterns</i>
SVM	<i>Support Vector Machine</i>

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.2	Estrutura do trabalho	10
2	REVISÃO DA LITERATURA	11
3	MATERIAIS E MÉTODOS	13
3.1	Base de dados	13
3.2	Estrutura de uma BCI	14
3.2.1	Pré-processamento	15
3.2.1.1	<i>Common average reference (CAR)</i>	15
3.2.2	Extração de atributos	16
3.2.2.1	Periodograma	16
3.2.2.2	Periodograma de Welch	16
3.2.2.3	<i>Common spatial patterns</i>	17
3.2.2.4	<i>Filter bank common spatial patterns (FBCSP)</i>	18
3.2.3	Seleção de atributos	19
3.2.3.1	<i>Forward selection</i>	19
3.2.3.2	Informação mútua	20
3.2.4	Classificação	20
3.2.4.1	Classificador linear – Método dos mínimos quadrados	21
3.2.4.2	<i>Support vector machine</i>	22
3.2.4.3	<i>Naïve Bayes</i>	23
3.3	Implementação	24
4	RESULTADOS E DISCUSSÕES	27
4.1	Análise das técnicas de processamento	27
4.2	Melhores resultados	31
5	CONCLUSÃO	33
	REFERÊNCIAS	34

1 Introdução

Interfaces cérebro-computador (BCI – *Brain-Computer Interfaces*) são sistemas de comunicação em *hardware* e *software* que permitem aos seres humanos interagirem com o seu meio através de sinais adquiridos diretamente do cérebro, com o envolvimento mínimo de nervos periféricos e músculos. Essas interfaces criam um novo canal de comunicação para as pessoas transmitirem suas intenções a dispositivos externos, como computadores, sintetizadores de fala, próteses, etc. Sendo especialmente atrativas para pessoas com deficiências motoras severas (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

A aquisição dos sinais do cérebro pode se dar de forma invasiva, através de implantes inseridos diretamente no cérebro, ou não invasiva, através de dispositivos capazes de medir a atividade cerebral de fora do crânio. Interfaces invasivas possuem um apelo menor devido aos riscos e dificuldade envolvendo a instalação de implantes no cérebro. A forma mais frequentemente usada para se implementar uma BCI não invasiva faz uso do eletroencefalograma (EEG), cujo equipamento é relativamente barato e fácil de usar, outras possíveis formas não invasivas incluem magnetoencefalografia e ressonância magnética funcional (GRAIMANN; PFURTSCHELLER; ALLISON, 2010).

O EEG mede variações de potencial elétrico causadas por correntes iônicas nos neurônios durante atividade sináptica. A aquisição dos sinais se dá a partir de eletrodos posicionados sobre o couro cabeludo, porém esses sinais são fracos e estão altamente sujeitos a ruídos com origem interna e externa ao cérebro (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

Uma abordagem de implementação de sistemas BCI é baseada na imagética motora, que consiste em se imaginar a realização de um movimento sem executá-lo. Há uma grande similaridade entre as áreas ativas no cérebro durante a execução de movimentos e imagética motora, tanto que uma imagem clara do movimento pretendido pode estar presente mesmo sem nenhuma atividade muscular (LOTZE; HALSBAND, 2006). A imagética motora provoca alterações nos ritmos cerebrais chamadas de sincronização relacionada a evento (ERS – *Event-Related Synchronization*), em que um grande número de neurônios disparam ritmicamente em sincronia em resposta a um estímulo externo ou interno ao cérebro, e dessincronização relacionada a evento (ERD – *Event-Related Desynchronization*), quando os neurônios disparam sem um padrão claro. Quando os neurônios disparam em sincronia, criam sinais elétricos que podem ser detectados através do EEG, e quando disparam de forma dessincronizada os sinais elétricos se cancelam, reduzindo a atividade percebida no EEG. Devido à limitada resolução espacial do EEG, normalmente são usadas para controlar a BCI imagética motora da mão esquerda, mão direita, ambos os pés e língua,

que provocam alterações nos ritmos em regiões do cérebro comparativamente grandes e topologicamente diferentes (GRAIMANN; PFURTSCHELLER; ALLISON, 2010).

Interfaces cérebro-computador trazem novas opções para pessoas com condições debilitantes devastadoras como esclerose lateral amiotrófica, acidente vascular de tronco encefálico, paralisia cerebral e lesão da medula espinhal. O principal propósito do desenvolvimento desses sistemas é para permitir a esses usuários transmitir seus desejos a cuidadores, usar algum *software* ou até controlar um braço robótico ou uma neuroprótese (MCFARLAND; WOLPAW, 2011).

1.1 Objetivos

- Desenvolver a etapa de processamento digital de sinais de uma BCI, compreendendo pré-processamento, extração de atributos, seleção de atributos e classificação.
- Avaliar o impacto de diferentes técnicas em cada estágio do processamento sobre a acurácia do sistema.
- Verificar quais combinações de técnicas produzem os melhores resultados.

1.2 Estrutura do trabalho

Este trabalho é dividido em cinco capítulos, no primeiro, são introduzidos conceitos básicos e possíveis aplicações para a proposta implementada.

No Capítulo 2 é feita uma breve revisão da literatura recente identificando as principais técnicas de processamento e estratégias para a elaboração de uma interface cérebro-computador.

No Capítulo 3 é apresentada uma descrição da base de dados utilizada e a definição dos métodos usados no decorrer deste trabalho.

O Capítulo 4 apresenta detalhes da implementação do sistema e define os parâmetros dos métodos descritos no Capítulo 3. Em seguida são apresentadas e discutidas as melhores combinações e o impacto das técnicas utilizadas.

Por fim, no Capítulo 5 é apresentada a conclusão bem como sugestões para trabalhos futuros.

2 Revisão da literatura

Segundo [Ang e Guan \(2017\)](#) há três diferentes estratégias para detectar imagética motora a partir do EEG: condicionamento do operante, consiste em se utilizar um modelo fixo sendo predefinidos os parâmetros e quais atributos extrair, assim os usuários da BCI precisam aprender a controlar ritmos cerebrais específicos para interagir com esses sistemas. Baseada em aprendizado de máquina, em que o modelo se adapta às características individuais dos usuários, eliminando a necessidade do usuário aprender a controlar ritmos específicos. Por fim, uma estratégia adaptativa, enquanto as BCIs baseadas em aprendizado de máquina são “concretizadas” com base em dados adquiridos em seções de calibração, uma BCI adaptativa continua a ajustar seu modelo conforme é utilizada. Também são apresentados os resultados do uso de dois sistemas usando o algoritmo FBCSP (*Filter Bank Common Spatial Patterns*), o primeiro *online* e baseado em aprendizado de máquina, e o segundo *offline* aplicando a estratégia adaptativa usando os dados adquiridos no primeiro experimento. A acurácia do sistema primeiro sistema durante o treinamento foi de 79.8% e caiu para 69.5% nas seções *online*, e a estratégia adaptativa resultou em uma melhoria de 12% no desempenho das seções *offline*.

No trabalho de [Deuel et al. \(2017\)](#) foi implementada uma BCI para controlar um sintetizador de piano usando imagética motora da mão direita explorando o ritmo *mu*, e é usado como atributo para classificação a energia na faixa de 8 a 12 Hz no sinal do eletrodo C3. Após a calibração do sistema, o valor da energia é quantizado e usado para selecionar a nota em uma escala musical, sendo oito possíveis valores. Seguindo a definição de [Ang e Guan \(2017\)](#), essa BCI se enquadra na estratégia de condicionamento do operante. Dentre 15 voluntários foi observada uma acurácia média de 57.1%, sendo a acurácia esperada em caso de escolha aleatória de 19.03%.

[Mondini, Mangia e Cappello \(2016\)](#) desenvolveu uma BCI adaptativa totalmente automatizada permitindo o controle de um braço com um grau de liberdade destinada a auxiliar no processo de treinamento para se controlar ritmos sensório-motores. O sistema realiza o pré-processamento dos sinais usando CAR (*Common Average Reference*), emprega o algoritmo CSP (*Common Spatial Patterns*) para extração de atributos e SVM (*Support Vector Machine*) para classificação. Foram realizados testes *online* com dez participantes, dos quais oito não possuíam experiência com imagética motora. Dos dez participantes, três alcançaram uma acurácia de 90% em três dias e outros quatro alcançaram 70%.

Em ([CARINO-ESCOBAR et al., 2016](#)) os autores buscaram desenvolver uma estratégia de classificação de imagética motora com boa performance e baixo custo computacional. É realizado o pré-processamento dos sinais com CAR, filtros FIR passa-baixas

e passa-altas com frequências de corte 30 e 8 Hz, respectivamente, e filtros rejeita-faixa centrados em 60 e 120 Hz com faixa de rejeição de 3 Hz de largura. Para a extração de atributos os sinais são convolvidos com *wavelets* de Morlet complexas variando a frequência de 8 a 30 Hz com resolução de 0.5 Hz. São usados como atributos a média da potência em todas as faixas de frequência (8 a 30 Hz) em janelas de 10 ms no intervalo de 1 s após 0.5 s da exibição da imagem indicando o tipo de imagética motora, totalizando 100 atributos por canal do EEG. Para realizar a classificação é aplicado o modelo de disparos neuronais (*spiking neurons model*) para identificar imagética motora de duas classes: mão esquerda e mão direita. A classificação é realizada usando uma rede neural com uma camada, e são testadas dois tipos de codificação para os atributos. Para comparação, também é avaliada a classificação com SVM, análise discriminante linear e rede neural artificial. O sistema foi testado em 4 seções com 6 pacientes que sofreram AVC e 6 participantes saudáveis, todos sem experiência com imagética motora. Os resultados obtidos indicam que o método aplicado alcança uma performance competitiva, comparado aos outros métodos implementados.

Em (MIAO et al., 2017) é apresentado o classificador *dictionary cleaned sparse representation-based classification* (DCSRC), fundamentado na classificação baseada em representação esparsa (*sparse representation-based classification* – SRC). O sistema foi testado com 5 participantes sem experiência prévia com BCIs, para diferenciar entre imagética motora do dedo indicador da mão direita e repouso. Foram gravados 120 ensaios para cada participante em 3 seções no mesmo dia. Os sinais foram inicialmente filtrados, sendo as frequências de corte definidas individualmente para cada participante, e a eliminação de ensaios contaminados por artefatos realizada por inspeção. Foram usados como atributos a energia e a entropia aproximada do sinal em janelas de 1 s com sobreposição de 50%, em intervalos de 4 s dos ensaios. O sistema alcançou uma acurácia média de 81.32% aplicando DCSRC e 78.79% com SRC.

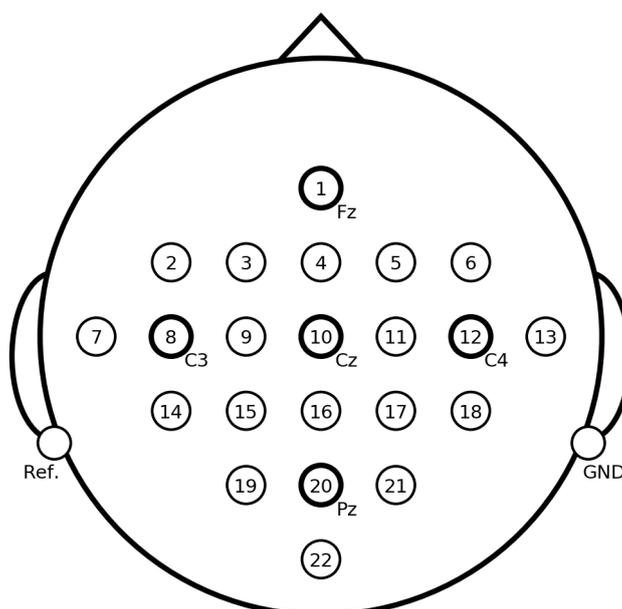
3 Materiais e métodos

Neste Capítulo são apresentadas as técnicas estudadas no decorrer deste trabalho e a estrutura dos dados utilizados para avaliá-las. E por fim são detalhados os parâmetros e ferramentas empregadas na implementação.

3.1 Base de dados

A base de dados usada foi disponibilizada para a BCI Competition IV ([BRUNNER et al.](#),) e contém dados rotulados de nove participantes executando tarefas de imagética motora de quatro classes: mão esquerda, mão direita, pés e língua. Os sinais do EEG foram gravados com os eletrodos distribuídos de acordo com o sistema 10-10, como mostrado na [fig. 1](#), com uma taxa de amostragem de 250 Hz, os sinais também foram filtrados com um filtro passa-faixa de 0.5 a 100 Hz e um filtro *notch* centrado em 50 Hz para eliminar ruído da rede.

Figura 1 – Disposição dos eletrodos.



Cada ensaio teve duração de 7.5 segundos, ao início do ensaio é exibida uma cruz de fixação por 2 segundos seguida de uma imagem indicando para qual classe o participante deve realizar imagética motora. A classe é escolhida aleatoriamente e a imagem permanece

na tela por 1.252 s. Os primeiros 2.5 segundos dos sinais foram desconsiderados, levando em conta o período antes da imagem ser exibida e o tempo de reação dos participantes. A fig. 2 ilustra a estrutura de um ensaio da base de dados, e o número de ensaios de cada tipo para cada participante é mostrado na tabela 1.

Figura 2 – Estrutura do ensaio.

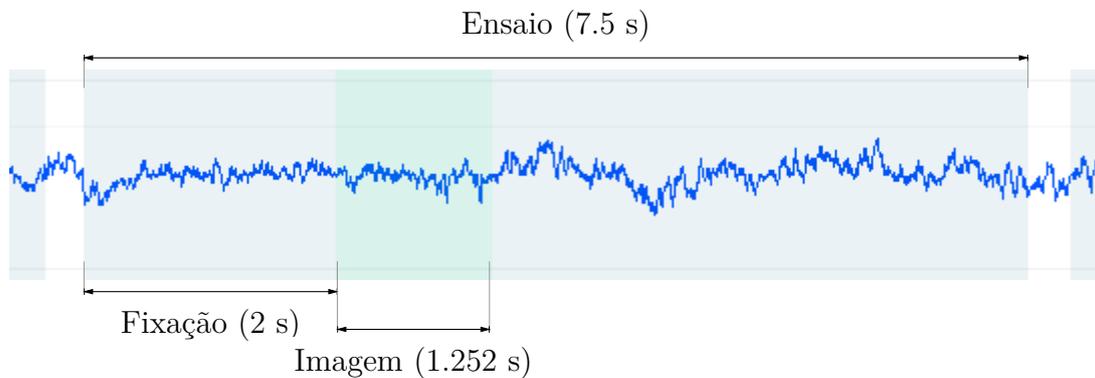


Tabela 1 – Número de ensaios.

Participante	Esquerda	Direita	Pés	Língua
1	140	139	137	138
2	138	140	135	140
3	136	138	134	135
4	121	124	126	119
5	133	131	138	136
6	109	112	103	110
7	138	135	138	137
8	132	134	133	136
9	118	128	130	125

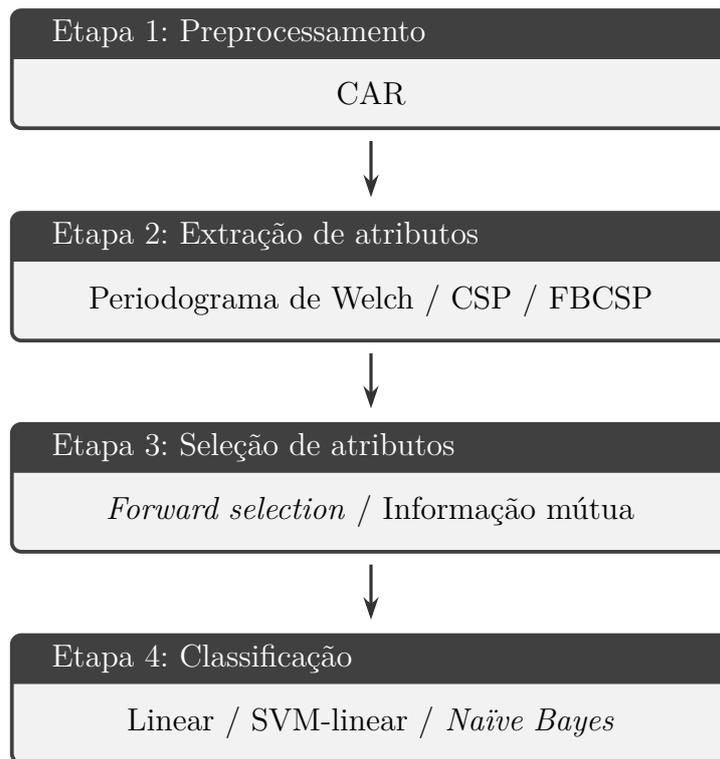
3.2 Estrutura de uma BCI

Uma BCI completa engloba a aquisição e amplificação dos sinais, processamento dos sinais para identificar comandos, e uma interface de controle para transmitir esses comandos para uma aplicação.

O processamento dos sinais por si pode ser dividido em quatro etapas: pré-processamento, extração de atributos, seleção de atributos e classificação. Em cada uma dessas etapas diferentes estratégias e algoritmos podem ser aplicados.

A figura 3 mostra as técnicas implementadas em cada etapa neste trabalho.

Figura 3 – Etapas do processamento do sinal.



3.2.1 Pré-processamento

Os eletrodos usados para adquirir sinais de EEG são sensíveis a diversos tipos de ruído, incluindo ruído de linha, de instrumentação, térmico e outras fontes de interferência biológicas. Na etapa de pré-processamento busca-se suprimir características indesejáveis do sinal como ruídos e artefatos e exaltar características desejáveis.

Ruídos podem ser correlacionados, comuns a todos os canais, ou não correlacionados. Ruídos correlacionados podem ser introduzidos por variações de potencial elétrico na região usada como referencial elétrico durante a aquisição dos sinais.

3.2.1.1 *Common average reference* (CAR)

CAR é uma técnica simples que visa criar uma referência elétrica melhor através da média aritmética dos sinais gravados, permitindo eliminar boa parte do ruído correlacionado, e como demonstrado em Ludwig et al. (2009), tende a aumentar significativamente a relação sinal-ruído.

O valor da referência para o n -ésimo canal pode ser computado como:

$$CAR_n(t) = \frac{1}{N-1} \sum_{i \neq n} x_i(t) \quad (3.1)$$

onde N é o número de canais do EEG e $x_i(t)$ é o sinal do i -ésimo canal. Esse sinal de referência é subtraído de cada canal:

$$x'_i(t) = x_i(t) - CAR_i(t) \quad (3.2)$$

3.2.2 Extração de atributos

Nessa etapa busca-se extrair características dos sinais de entrada que podem ser usadas para a discriminação das classes de imagética. As alterações dos ritmos cerebrais causadas pela imagética motora são mais proeminentes nos ritmos *mu* (8-12 Hz) e *beta* (12-30 Hz) (GRAIMANN; PFURTSCHELLER; ALLISON, 2010). Tendo isso em mente, uma das estratégias mais intuitivas para extração de atributos é tentar buscar alterações na energia do sinal nessas faixas de frequência, para isso pode ser usada a densidade espectral de potência estimada pelo periodograma e periodograma de Welch.

3.2.2.1 Periodograma

O periodograma é uma forma de se estimar a densidade espectral de potência de um sinal, é obtido pelo quadrado do módulo da transformada de Fourier discreta (DFT – *Discrete Fourier Transform*) (OPPENHEIM, 2010). Sendo a DFT definida como:

$$X[k] = \sum_{n=0}^{T-1} x[n]e^{-j2\pi \frac{kn}{T}} \quad (3.3)$$

onde $x[n]$ é um sinal discreto qualquer e T o número de amostras do mesmo, o periodograma pode ser computado da forma:

$$S[k] = |X[k]|^2 = \left| \sum_{n=0}^{T-1} x[n]e^{-j2\pi \frac{kn}{T}} \right|^2 \quad (3.4)$$

Por fim, define-se a matriz de atributos A com a energia estimada dos sinais dos N canais do EEG nas dez faixas: 0 a 4 Hz, 4 a 8 Hz, ... e 36 a 40 Hz.

$$A = \begin{bmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,10} \\ e_{2,1} & e_{2,2} & \dots & e_{2,10} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N,1} & e_{N,2} & \dots & e_{N,10} \end{bmatrix}, \quad e_{n,i} = \sum_{k=\lfloor 4(N-1)/f_n \rfloor}^{\lfloor 4N/f_n \rfloor - 1} S_n[k] \quad (3.5)$$

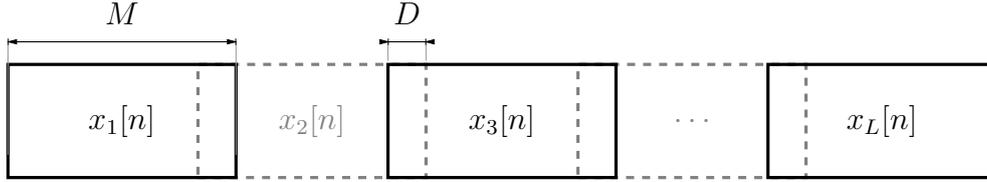
3.2.2.2 Periodograma de Welch

O periodograma de Welch é uma forma de se estimar a densidade espectral de potência com a vantagem de se reduzir a quantidade de ruído estimada em troca de uma menor resolução na frequência.

A densidade espectral estimada pelo periodograma de Welch é computada pelo seguinte procedimento:

1. Divide-se o sinal de entrada $x[n]$ em L segmentos de tamanho M de forma que segmentos lado a lado tem uma sobreposição de D pontos como ilustrado na fig. 4.
2. Multiplica-se cada segmento por uma função de janelamento $w[n]$.
3. Computa-se o periodograma de cada segmento janelado.
4. Calcula-se a média dos periodogramas.

Figura 4 – Periodograma de Welch.



Esse algoritmo também pode ser escrito como:

$$\hat{S}[k] = \frac{1}{L} \sum_{i=0}^{L-1} \left| \sum_{n=0}^{M-1} x_i[n] w[n] e^{-j2\pi \frac{kn}{T}} \right|^2 \quad (3.6)$$

sendo $x_i[n] = x[n - i(M - D)]$ o i -ésimo segmento (OPPENHEIM, 2010).

Os atributos são extraídos como descrito na equação 3.5.

3.2.2.3 Common spatial patterns

O objetivo desse método é o projeto de um filtro espacial cuja saída é uma série temporal ótima para separar duas classes, o CSP decompõe os sinais em subcomponentes de modo a maximizar a variância entre duas janelas (RAMOSER; MULLER-GERKING; PFURTSCHELLER, 2000). Os sinais do EEG de cada ensaio são representados como uma matriz $N \times T$, onde N é o número de canais e T o número de amostras por canal.

Primeiramente obtém-se a matriz de covariância espacial normalizada de cada ensaio E :

$$C = \frac{EE^T}{\text{tr}(EE^T)} \quad (3.7)$$

sendo $\text{tr}(X)$ o traço (soma dos elementos da diagonal principal) da matriz.

Supondo que se deseja diferenciar duas classes \mathbf{a} e \mathbf{b} , $\bar{C}_{\mathbf{a}}$ e $\bar{C}_{\mathbf{b}}$ representam as médias das matrizes de covariância dos ensaios de cada classe. A variância espacial composta é então dada por:

$$C_c = \bar{C}_{\mathbf{a}} + \bar{C}_{\mathbf{b}} \quad (3.8)$$

Essa matriz pode ser fatorada na forma $C_c = U_c \lambda_c U_c^T$, onde λ_c é a matriz diagonal dos autovalores de C_c ordenados de maior para menor e U_c é a matriz com os autovetores correspondentes em suas colunas.

A transformação de branqueamento (*whitening*) $P = \sqrt{\lambda_c^{-1}} U_c^\top$ equaliza a variância no espaço definido por U_c . Aplicando essa transformação a \bar{C}_a e \bar{C}_b obtém-se as matrizes $S_a = P\bar{C}_aP^\top$ e $S_b = P\bar{C}_bP^\top$ que compartilham os mesmos autovetores:

$$S_a = B\lambda_a B^\top, \quad S_b = B\lambda_b B^\top, \quad \lambda_a + \lambda_b = I \quad (3.9)$$

onde I é a matriz identidade.

Por fim define-se a matriz de projeção como $W = (B^\top P)^\top$, que pode ser usada para fazer a decomposição de cada ensaio E da forma:

$$Z = W^\top E \quad (3.10)$$

O CSP maximiza a diferença entre as variâncias dos sinais para as duas classes, sendo assim são usados como atributos as variâncias normalizadas dos sinais transformados:

$$f_p = \log \left[\frac{\text{var}(Z_p)}{\sum_{i=1}^{2m} \text{var}(Z_i)} \right], \quad Z = \hat{W}^\top E \quad (3.11)$$

onde Z_i ($i \in \{1, 2, \dots, 2m\}$) é a i -ésima linha de Z , $\hat{W} \in \mathbb{R}^{C \times 2m}$ é a matriz composta pelas primeiras e últimas m colunas de W e $\log x$ é usado para aproximar uma distribuição normal (RAMOSER; MULLER-GERKING; PFURTSCHELLER, 2000) (ANG et al., 2012).

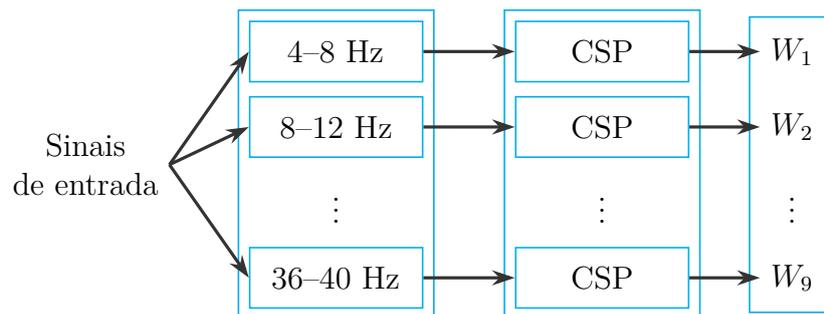
3.2.2.4 Filter bank common spatial patterns (FBCSP)

A performance da classificação de imagética motora usando atributos extraídos diretamente dos sinais decompostos pelo CSP geralmente é baixa quando os sinais de entrada não são filtrados ou são filtrados com uma gama de frequências inapropriada (ANG et al., 2008).

Para contornar essa limitação evitando a necessidade de se escolher manualmente as faixas de frequências para cada indivíduo, o FBCSP emprega um banco de filtros à entrada do CSP.

Os sinais do EEG passam por nove filtros passa-faixa com bandas de passagem de 4 a 8 Hz, 8 a 12 Hz, ... e 36 a 40 Hz. O sinal filtrado por cada um dos passa-faixas é então decomposto pelo CSP como mostrado na figura 5.

Figura 5 – Banco de filtros e filtragem espaciais do FBCSP.



Seja W_b a matriz de projeção do CSP para cada faixa de passagem do banco de filtros, os atributos extraídos de cada ensaio E podem ser definidos da forma:

$$V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_9], \quad \mathbf{v}_b = \log \left[\frac{\text{diag}(\hat{W}_b^T E_b E_b^T \hat{W}_b)}{\text{tr}(\hat{W}_b^T E_b E_b^T \hat{W}_b)} \right] \quad (3.12)$$

onde $\hat{W}_b \in \mathbb{R}^{C \times 2m}$ é a matriz composta pelas primeiras e últimas m colunas de W_b e E_b é a saída do b -ésimo filtro (ANG et al., 2012).

3.2.3 Seleção de atributos

Após a extração de atributos é preciso escolher quais atributos serão usados para classificar os sinais. Uma possibilidade é usar todos os atributos extraídos, porém dependendo da quantidade, a classificação pode ter um custo computacional muito elevado, e dependendo do classificador, ainda pode resultar em uma taxa de acertos menor.

A seleção de atributos consiste em selecionar uma combinação de atributos de modo a maximizar a performance do classificador.

3.2.3.1 Forward selection

A seleção de atributos pode ser vista como um problema de otimização: busca-se construir um subconjunto de atributos de modo a maximizar o desempenho do sistema. Sendo assim, é possível aplicar técnicas de otimização como meta-heurísticas e algoritmos gulosos nessa etapa.

Um algoritmo guloso simples para realizar a seleção de atributos é o *forward selection* (alg. 1), que usa como heurística alguma métrica relacionada ao desempenho do sistema, como a taxa de acertos. Começa com um conjunto vazio e a cada iteração inclui o atributo que mais aumenta a taxa de acertos (EFROYMSON, 1960).

Procedure 1 *Forward selection***Input:** conjunto de atributos F **Output:** atributos selecionados

```

1:  $S \leftarrow \emptyset$  ▷ atributos selecionados
2:  $r \leftarrow 0$  ▷ taxa de acertos
3: while  $F \neq \emptyset$  do
4:    $r_i \leftarrow 0$ 
5:   for  $f \in F$  do
6:      $r'_i \leftarrow eval(S \cup \{f\})$  ▷ calcula a taxa de acertos incluindo o atributo
7:     if  $r'_i > r_i$  then
8:        $r_i \leftarrow r'_i$ 
9:        $f_{max} \leftarrow f$  ▷ salva o atributo que mais aumenta a taxa de acertos
10:    end if
11:  end for
12:  if  $r_i > r$  then
13:     $S \leftarrow S \cup \{f_{max}\}$ 
14:     $F \leftarrow F - \{f_{max}\}$ 
15:  else
16:    return  $S$  ▷ termina quando o atributo não aumenta a taxa de acertos
17:  end if
18: end while

```

3.2.3.2 Informação mútua

Informação mútua é uma medida da dependência entre variáveis aleatórias. É uma forma mais geral que o coeficiente de correlação, não estando limitada a valores reais e sendo capaz de identificar dependências que não se manifestam na covariância, tanto que a informação mútua é zero se e somente se as variáveis aleatórias são estritamente independentes (KRASKOV; STÖGBAUER; GRASSBERGER, 2004).

Sejam X e Y duas variáveis aleatórias, a informação mútua entre as mesmas pode ser formalmente definida como:

$$I(X, Y) = \int_Y \int_X p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right) dx dy \quad (3.13)$$

onde $p(x, y)$ é a distribuição de probabilidade conjunta de X e Y , e $p(x)$ e $p(y)$ são as funções de probabilidade marginais de X e Y , respectivamente.

É computada a informação mútua entre cada atributo e a saída esperada do classificador e são escolhidos os k atributos com o maior índice.

3.2.4 Classificação

Na última etapa os dados dos atributos selecionados são usados para treinar um classificador que, espera-se, será capaz de reconhecer novos dados corretamente.

3.2.4.1 Classificador linear – Método dos mínimos quadrados

Uma das abordagens mais simples e computacionalmente eficientes para classificação são os classificadores lineares. Esse tipo de classificador normalmente representa as observações como pontos em um espaço n -dimensional e computa um hiperplano separando duas classes. Muitas vezes, porém, não existe um hiperplano capaz de separar todos os dados observados, nesse caso uma solução é buscar minimizar o erro de classificação (THEODORIDIS, 2009).

A função do hiperplano a ser obtida é dada por:

$$y(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + w_0 \quad (3.14)$$

sendo \mathbf{x} o vetor de atributos, \mathbf{w} o vetor de pesos do classificador, e w_0 o deslocamento do plano no espaço. Para simplificar a formulação matemática, w_0 pode ser “incluso” em \mathbf{w} , mas para isso é preciso acrescentar uma dimensão a \mathbf{x} com o valor 1. Feito isso, a função de erro pode ser descrita como:

$$\mathbf{J}(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \mathbf{w})^2 \quad (3.15)$$

onde \mathbf{x}_i o i -ésimo vetor de atributos e $y_i \in \{-1, 1\}$ sua respectiva saída esperada. Minimizando essa função, tem-se:

$$\sum_{i=1}^N \mathbf{x}_i (y_i - \mathbf{x}_i^\top \mathbf{w}) = 0 \quad (3.16)$$

$$\sum_{i=1}^N \mathbf{x}_i y_i = \left(\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w} \quad (3.17)$$

Definindo X como a matriz $N \times l$, sendo N o número de observações e l o número de atributos, é possível reescrever a equação acima usando produtos de matrizes:

$$X = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nl} \end{bmatrix} \implies \begin{cases} \sum_{i=1}^N \mathbf{x}_i y_i = X^\top \mathbf{y} \\ \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top = X^\top X \end{cases} \quad (3.18)$$

$$X^\top \mathbf{y} = (X^\top X) \mathbf{w} \quad (3.19)$$

Por fim, pode-se isolar o vetor de pesos:

$$\mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y} \quad (3.20)$$

Tendo obtido o vetor de pesos, a classificação de dados novos se dá pela relação:

$$y(\mathbf{x}) = \text{sgn}(\mathbf{x}^\top \mathbf{w}), \quad \text{sgn}(x) = \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases} \quad (3.21)$$

3.2.4.2 Support vector machine

De forma semelhante ao classificador linear, a SVM constrói um conjunto de hiperplanos no espaço n -dimensional separando as classes de dados. Esses hiperplanos são posicionados de forma a tentar maximizar a distância entre cada hiperplano e amostras de qualquer classe (PEDREGOSA et al., 2011).

De forma semelhante ao classificador linear define-se um hiperplano que separa duas classes da forma:

$$y(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + w_0 = 0 \quad (3.22)$$

onde \mathbf{w} é o vetor de pesos do classificador e w_0 o deslocamento do hiperplano no espaço. Define-se uma margem em relação à eq. 3.22 resultando no par de hiperplanos:

$$\mathbf{x}^\top \mathbf{w} + w_0 = \pm 1 \quad (3.23)$$

Sendo a variável de folga ξ_i tal que:

$$y_i y(\mathbf{x}_i) \leq 1 - \xi_i, \quad \xi_i \geq 0 \quad (3.24)$$

onde $y_i \in \{-1, 1\}$ é a classe do vetor de atributos \mathbf{x}_i . Os vetores de atributos caem em três categorias:

- Vetores fora da margem classificados corretamente ($\xi_i = 0$).
- Vetores dentro da margem classificados corretamente ($0 < \xi_i \leq 1$).
- Vetores classificados incorretamente ($1 < \xi_i$).

Agora busca-se tornar a margem o maior possível e ao mesmo tempo manter o número de vetores de atributos com $\xi_i > 0$ o menor possível, para isso define-se a função de custo:

$$J(\mathbf{w}, w_0, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (3.25)$$

onde C é uma constante positiva que define a influência relativa dos dois termos.

Pode-se simplificar esse problema de otimização usando multiplicadores de Lagrange, para isso a função Lagrangiana é definida como:

$$\mathcal{L}(\mathbf{w}, w_0, C, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \lambda_i [y_i (\mathbf{x}_i^\top \mathbf{w} + w_0) - 1 + \xi_i] \quad (3.26)$$

Assim o problema se torna:

$$\begin{aligned} & \text{maximizar: } \mathcal{L}(\mathbf{w}, w_0, C, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ & \text{com as restrições: } y_i (\mathbf{x}_i^\top \mathbf{w} + w_0) \leq 1 - \xi_i \\ & \qquad \qquad \qquad \xi_i \geq 0 \\ & \qquad \qquad \qquad C - \mu_i - \lambda_i = 0 \\ & \qquad \qquad \qquad \mu_i \geq 0 \\ & \qquad \qquad \qquad \lambda_i \geq 0 \end{aligned}$$

Como $J(\mathbf{w}, w_0, \boldsymbol{\xi})$ é uma função convexa, no ponto ótimo tem-se:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, w_0, C, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \implies \mathbf{w} = \sum_{i=1}^N \lambda_i \mathbf{x}_i y_i \quad (3.27)$$

$$\frac{\partial}{\partial w_0} \mathcal{L}(\mathbf{w}, w_0, C, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \implies \sum_{i=1}^N \lambda_i y_i = 0 \quad (3.28)$$

$$\frac{\partial}{\partial \xi_i} \mathcal{L}(\mathbf{w}, w_0, C, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \implies C - \mu_i - \lambda_i = 0 \quad (3.29)$$

Substituindo as equações acima na eq. 3.26 pode-se simplificar o problema para a forma:

$$\begin{aligned} & \max_{\boldsymbol{\lambda}} \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ & \text{com as restrições: } 0 \leq \lambda_i \leq C \\ & \qquad \qquad \qquad \sum_{i=1}^N \lambda_i y_i = 0 \end{aligned}$$

Obtida a solução $\boldsymbol{\lambda}^*$, \mathbf{w}^* pode ser computado pela relação 3.27 e w_0^* pela expressão:

$$w_0^* = \frac{1}{N_{sv}} \sum_{\mathbf{x}_i \in SV} (y_i - \mathbf{x}_i^\top \mathbf{w}^*) \quad (3.30)$$

onde SV é o conjunto dos vetores de suporte (*support vectors*), vetores de atributos que residem sobre os hiperplanos das margens ($\mathbf{x}_i \mid \lambda_i^* > 0$).

Por fim, a classificação de dados novos pode ser feita usando a relação:

$$y(\mathbf{x}) = \text{sgn}(\mathbf{x}^\top \mathbf{w}^* + w_0^*) \quad (3.31)$$

3.2.4.3 Naïve Bayes

Naïve Bayes é um classificador probabilístico baseado no teorema de Bayes e na premissa de que todos os atributos são independentes, porém o classificador ainda pode ser muito robusto mesmo com violações dessa suposição (THEODORIDIS, 2009) (ZHANG, 2004). O teorema de Bayes se dá pela equação:

$$p(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k) p(C_k)}{p(\mathbf{x})} \quad (3.32)$$

onde $p(C_k | \mathbf{x})$ representa a probabilidade do vetor de atributos \mathbf{x} pertencer à classe C_k , $p(C_k)$ é a frequência da classe C_k e supondo que todas as variáveis x_i são independentes, $p(\mathbf{x} | C_k)$ é dado por:

$$p(\mathbf{x} | C_k) = \prod_{i=1}^n p(x_i | C_k) \quad (3.33)$$

A distribuição $p(x_i | C_k)$ é estimada com base no conjunto de treinamento. Assumindo uma distribuição gaussiana, a probabilidade de um valor observado v dada a classe C_k é:

$$p(x_i = v | C_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} \exp\left[-\frac{(v - \mu_{ik})^2}{2\sigma_{ik}^2}\right] \quad (3.34)$$

sendo μ_{ik} e σ_{ik}^2 , respectivamente, a média e a variância da variável x_i associada à classe k nos dados de treinamento.

A classificação de um vetor de atributos pode ser feita da forma:

$$y(\mathbf{x}) = \arg \max_{C_k} \left[p(C_k) \prod_{i=1}^n p(x_i | C_k) \right] \quad (3.35)$$

observando que o denominador da eq. 3.32 não depende de C_k e foi desconsiderado.

3.3 Implementação

A implementação do sistema foi realizada em Python 3.6 (ROSSUM, 1995) fazendo uso intensivo das bibliotecas SciPy (JONES et al., 2001–) e *scikit-learn* (PEDREGOSA et al., 2011). Python é uma linguagem de alto nível e de código aberto, possui uma vasta gama de recursos e bibliotecas disponíveis para computação científica, permitindo a criação de sistemas complexos de forma rápida e flexível (OLIPHANT, 2007). A biblioteca SciPy implementa rotinas numéricas – operações com matrizes, FFT (*Fast Fourier Transform*), etc. – de forma otimizada em linguagens de baixo nível e a biblioteca *scikit-learn* inclui métodos de processamento, extração de atributos e classificação, destinados à concepção de sistemas de aprendizado de máquina.

Como algumas etapas do processamento dos sinais são opcionais ou possuem mais de um método para desempenhá-las, foi elaborado um procedimento para gerar todas as combinações aplicáveis de métodos e parâmetros. A tabela 2 lista todas as etapas e seus respectivos métodos.

Para o periodograma de Welch foi usada uma janela de *Hanning* com duração de dois segundos e sobreposição de 50%, que apresentou os melhores resultados durante testes preliminares.

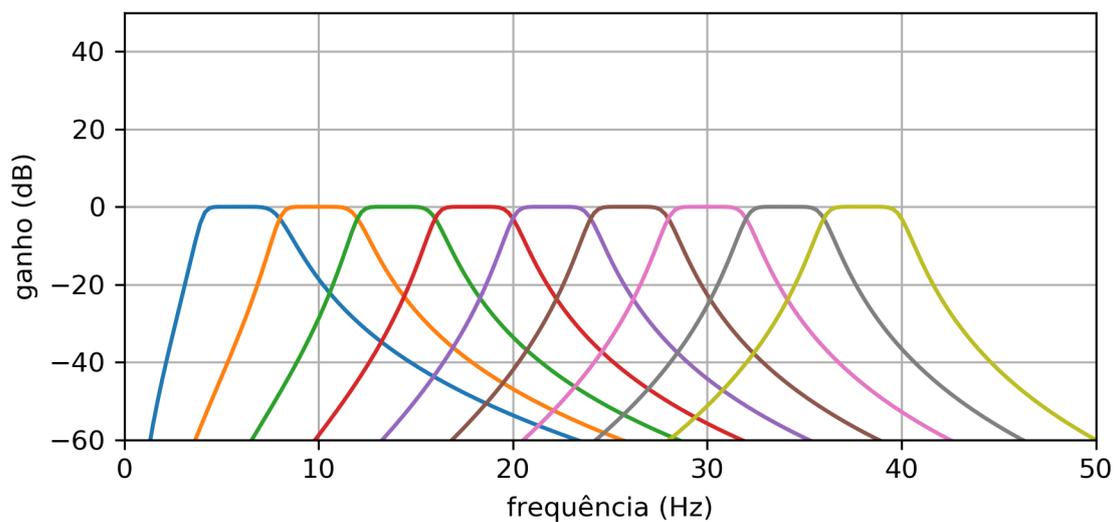
Para o CSP e FBCSP foi usado $m = 4$, selecionando os quatro filtros espaciais mais relevantes para cada classe, como sugerido em Ang et al. (2008). Os sinais passaram

Tabela 2 – Métodos implementados

Etapa	Opcional	Método
Pré-processamento	Sim	CAR
Extração de atributos	Não	Periodograma de Welch CSP FBCSP
Pós-processamento	Sim	Normalização dos atributos
Seleção de atributos	Sim	Informação mútua ($k \in \{3, 5, 8, 10\}$)
Classificação	Não	SVM ($C \in \{0.01, 0.1, 1, 10\}$) <i>Naïve Bayes</i>

por um filtro passa-faixa Butterworth de quarta ordem com banda de passagem de 8 a 30 Hz. O banco de filtros do FBCSP também foi implementado usando filtros passa-faixa Butterworth de quarta ordem (fig. 6), os quais possuem uma resposta mais uniforme na banda de passagem em comparação a filtros Chebyshev tipo II, levando a uma perda menor de informação.

Figura 6 – Banco de filtros Butterworth de 4ª ordem.



A normalização dos atributos consiste em subtrair a média dos mesmos ao longo de todos os dados de treinamento e então dividi-los por seu desvio padrão, isso resulta em uma distribuição centrada na origem com variância unitária.

Variando todas as opções de etapas/parâmetros listadas acima totalizam-se 300 combinações possíveis, a acurácia de cada uma dessas configurações foi aferida através de validação cruzada usando o método *holdout*: os dados foram embaralhados e divididos em conjuntos de treinamento e validação com 80 e 20% dos dados, respectivamente. Esse

processo foi repetido dez vezes, e o valor usado para a acurácia foi a média da taxa de acertos obtida com os dados de validação.

Vários dos métodos listados na tabela 2 permitem a discriminação de somente duas classes, para usá-los para diferenciar dentre quatro classes foi aplicada a estratégia um-contra-um: treina-se seis classificadores, um para cada par de classes, e a saída final do sistema é decidida por “votação”, em caso de empate, um dos mais votados é escolhido aleatoriamente.

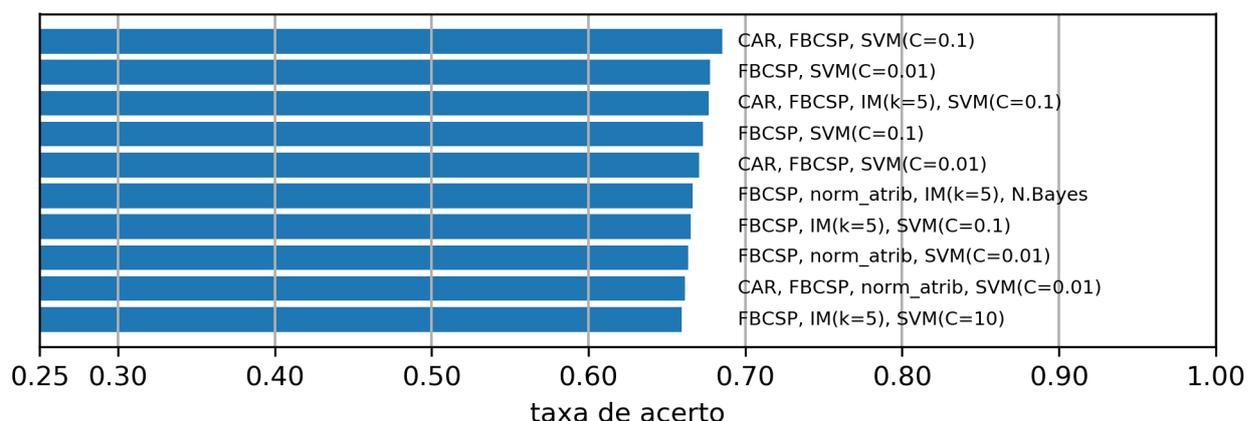
4 Resultados e discussões

Neste Capítulo são apresentados e discutidos os desempenhos das melhores combinações de etapas de processamento, bem como o impacto das técnicas de pré-processamento e seleção de atributos para cada método de extração/classificação. Por fim, é avaliado o desempenho das melhores configurações para cada indivíduo da base de dados.

4.1 Análise das técnicas de processamento

Os melhores resultados empregam FBCSP sendo a maior taxa de acertos 69.48%, enquanto os melhores resultados usando CSP e periodograma de Welch tiveram acurácia de 64.78% e 53.04%, respectivamente. As figuras 7-9 apresenta os dez melhores resultados para cada método de extração de atributos, que se mostrou a etapa que mais impacta a acurácia do sistema.

Figura 7 – Melhores resultados usando FBCSP.



Os três melhores resultados usando o FBCSP também empregam a extração de atributos baseada em informação mútua e classificação com *Naïve Bayes*. Tanto para o CSP quanto para o FBCSP, nove dos dez melhores resultados usaram informação mútua. Todos os melhores resultados obtidos aplicando CSP usam SVM, e a normalização dos atributos é aplicada em sete dos dez melhores resultados usando CSP e do periodograma de Welch. CAR é usada em todos os melhores resultados do periodograma de Welch.

O pré-processamento dos sinais com CAR afeta pouco desempenho final do sistema em combinação com CSP e FBCSP, porém melhora a taxa de acerto quando associada ao periodograma de Welch, como pode ser visto na tabela 3, que apresenta a acurácia média das combinações variando o método de extração, classificação e pré-processamento.

Figura 8 – Melhores resultados usando CSP.

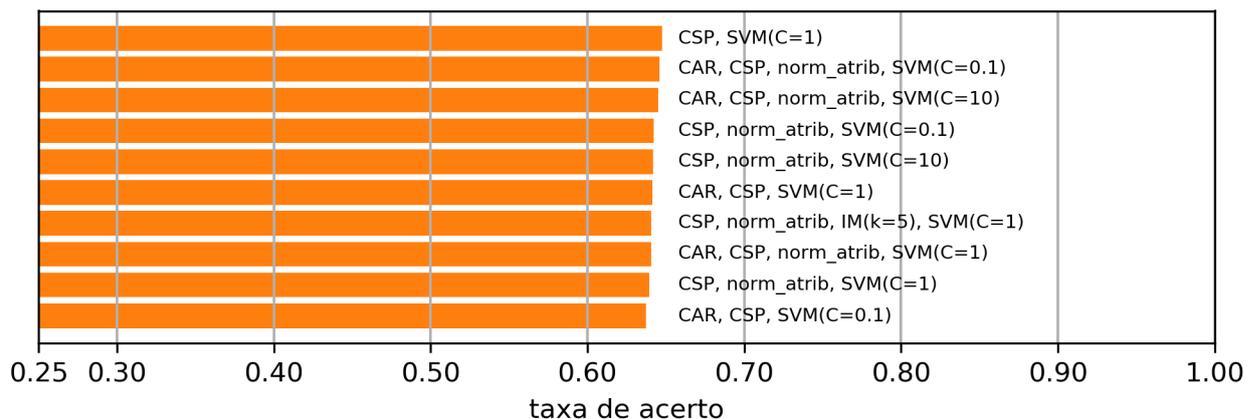


Figura 9 – Melhores resultados usando periodograma de Welch.

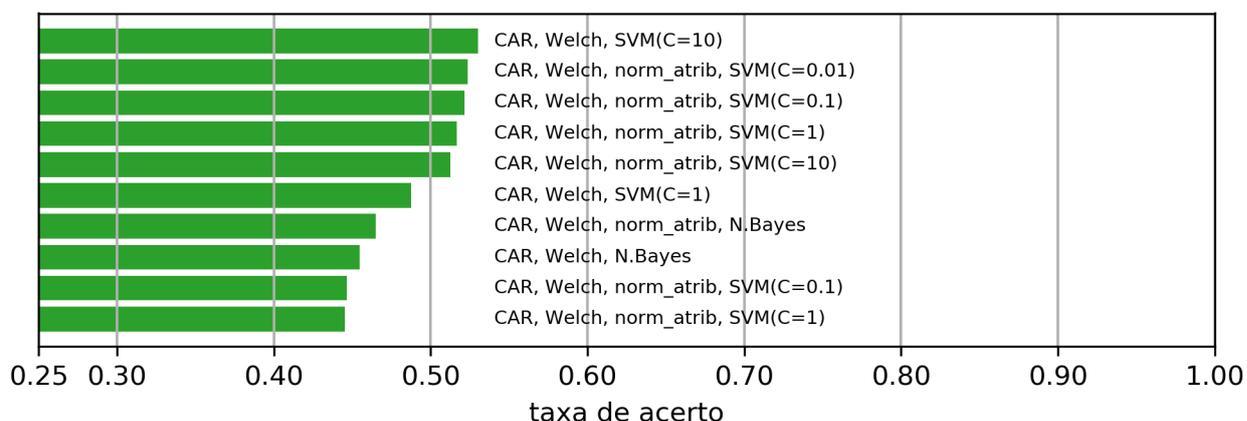


Tabela 3 – Impacto da técnica CAR sobre o desempenho do sistema.

Extração de atributos	Classificador	Sem pré-processamento	CAR
FBCSP	<i>Naïve Bayes</i>	0.6586	0.6618
	SVM(C=0.01)	0.6562	0.6565
	SVM(C=0.1)	0.6598	0.6629
	SVM(C=1)	0.6490	0.6480
	SVM(C=10)	0.6424	0.6404
CSP	<i>Naïve Bayes</i>	0.6111	0.6119
	SVM(C=0.01)	0.6135	0.6122
	SVM(C=0.1)	0.6270	0.6303
	SVM(C=1)	0.6352	0.6332
	SVM(C=10)	0.6308	0.6322
Welch	<i>Naïve Bayes</i>	0.3228	0.4328
	SVM(C=0.01)	0.3040	0.3783
	SVM(C=0.1)	0.3308	0.4157
	SVM(C=1)	0.3535	0.4382
	SVM(C=10)	0.3645	0.4519

Como apresentado na tabela 4, a normalização dos atributos extraídos do periodograma de Welch melhora consideravelmente o desempenho do sistema, especialmente em combinação com SVM. Também pode ser observado um aumento, porém menor, quando esse procedimento é aplicado aos atributos do CSP em combinação com SVM. Já para o FBCSP os resultados foram, em geral, ligeiramente inferiores.

Tabela 4 – Impacto da normalização dos atributos sobre o desempenho do sistema.

Extração de atributos	Classificador	Sem normalização	Com normalização
FBCSP	<i>Naïve Bayes</i>	0.6592	0.6612
	SVM(C=0.01)	0.6519	0.6608
	SVM(C=0.1)	0.6690	0.6538
	SVM(C=1)	0.6577	0.6393
	SVM(C=10)	0.6488	0.6340
CSP	<i>Naïve Bayes</i>	0.6123	0.6107
	SVM(C=0.01)	0.5981	0.6275
	SVM(C=0.1)	0.6231	0.6341
	SVM(C=1)	0.6341	0.6343
	SVM(C=10)	0.6288	0.6343
Welch	<i>Naïve Bayes</i>	0.3594	0.3604
	SVM(C=0.01)	0.2614	0.3971
	SVM(C=0.1)	0.3167	0.4007
	SVM(C=1)	0.3599	0.4024
	SVM(C=10)	0.3888	0.4000

A seleção de atributos baseada em informação mútua aumenta a acurácia das combinações baseadas em FBCSP, especialmente com valores maiores de k . Já para os sistemas usando CSP e periodograma de Welch a aplicação desse método resultou em uma queda no desempenho, como pode ser visto na tabela 5. Observando que o CSP com $m = 4$ tem como saída apenas oito atributos, por isso a seleção de atributos com $k \in \{8, 10\}$ não é aplicável.

Tabela 5 – Impacto da informação mútua sobre o desempenho do sistema.

Extração	Classificador	Sem seleção	IM ($k = 3$)	IM ($k = 5$)	IM ($k = 8$)	IM ($k = 10$)
FBCSP	<i>Naïve Bayes</i>	0.6475	0.6338	0.6577	0.6766	0.6854
	SVM(C=0.01)	0.6682	0.6088	0.6523	0.6718	0.6806
	SVM(C=0.1)	0.6636	0.6355	0.6630	0.6708	0.6740
	SVM(C=1)	0.6425	0.6357	0.6497	0.6569	0.6578
	SVM(C=10)	0.6390	0.6299	0.6496	0.6429	0.6455
CSP	<i>Naïve Bayes</i>	0.6081	0.6112	0.6152	–	–
	SVM(C=0.01)	0.6222	0.5976	0.6187	–	–
	SVM(C=0.1)	0.6406	0.6153	0.6300	–	–
	SVM(C=1)	0.6423	0.6222	0.6381	–	–
	SVM(C=10)	0.6383	0.6205	0.6357	–	–
Welch	<i>Naïve Bayes</i>	0.3707	0.3439	0.3560	0.3623	0.3665
	SVM(C=0.01)	0.3767	0.3013	0.3125	0.3219	0.3335
	SVM(C=0.1)	0.4184	0.3189	0.3369	0.3523	0.3670
	SVM(C=1)	0.4417	0.3392	0.3585	0.3786	0.3879
	SVM(C=10)	0.4611	0.3488	0.3703	0.3918	0.4000

4.2 Melhores resultados

A tabela 6 apresenta a melhor acurácia obtida para cada participante e a respectiva configuração. Seria possível implementar um sistema usando uma combinação diferente de técnicas de processamento para cada participante com uma acurácia média de 73.44%, em contraste com o melhor resultado de 69.48% mostrado na fig. 7, porém essa abordagem implicaria em maior risco de sobreajuste aos dados de treinamento.

Tabela 6 – Melhores resultados por participante.

Participante	Acurácia	Configuração
1	0.8055	FBCSP, norm_atrib, IM(k=5), SVM(C=0.01)
2	0.7389	FBCSP, IM(k=8), <i>Naïve Bayes</i>
3	0.8981	CAR, FBCSP, IM(k=5), SVM(C=0.1)
4	0.5276	CSP, SVM(C=1)
5	0.7208	CAR, FBCSP, IM(k=8), SVM(C=0.1)
6	0.4841	CAR, FBCSP, IM(k=10), <i>Naïve Bayes</i>
7	0.8745	CAR, FBCSP, IM(k=8), SVM(C=1)
8	0.8509	FBCSP, SVM(C=0.01)
9	0.7089	CAR, CSP, SVM(C=10)
média:	0.7344	

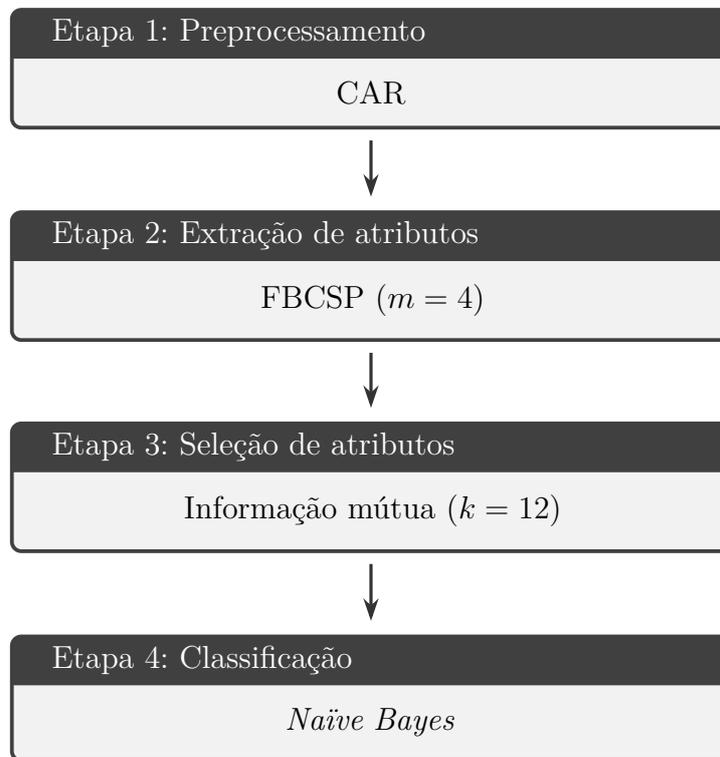
Os dados apresentados na fig. 7 sugerem que os melhores desempenhos do sistema podem ser obtidos com configurações incluindo FBCSP para extração de atributos, IM com k maior para seleção de atributos, e *Naïve Bayes* para classificação. Baseando-se nisso foram testadas combinações adicionais visando maximizar a acurácia. A tabela 7 lista a taxa de acerto da combinação CAR, FBCSP, informação mútua, e *Naïve Bayes*, com diferentes valores para o parâmetro k .

Tabela 7 – Acurácia em função de k .

Participante	$k = 10$	$k = 11$	$k = 12$	$k = 13$	$k = 15$
1	0.7745	0.8018	0.7820	0.8036	0.7564
2	0.7111	0.6892	0.6739	0.6676	0.7241
3	0.8537	0.8486	0.8413	0.8321	0.8426
4	0.4642	0.5551	0.5755	0.5724	0.4925
5	0.7000	0.6426	0.6824	0.6713	0.6868
6	0.4841	0.4908	0.4977	0.4724	0.4364
7	0.8564	0.8745	0.8700	0.8755	0.8309
8	0.8472	0.7710	0.8000	0.7832	0.8321
9	0.5625	0.7149	0.6851	0.6653	0.5542
média:	0.6948	0.7098	0.7120	0.7048	0.6840

Com isso, a melhor configuração obtida realiza o pré-processamento dos sinais usando CAR, extrai atributos com FBCSP, seleciona os melhores atributos com base em informação mútua com $k = 12$ e os classifica usando *Naïve Bayes*, com uma acurácia de 71.20%.

Figura 10 – Diagrama de fluxo da BCI com a configuração de melhor desempenho.



Na tabela 8 são apresentados os coeficientes *kappa* de Cohen (1960) obtidos pelos participantes da BCI Competition IV para cada indivíduo da base de dados e os valores alcançados pelo sistema mostrado na fig. 10 usando os mesmos conjuntos de treinamento e teste. Comparando o desempenho médio dos sistemas, a configuração da fig. 10 se classificaria entre o primeiro e segundo colocados da competição.

Tabela 8 – Comparação com os resultados da competição.

Colocação	Acurácia por indivíduo									Média
	1	2	3	4	5	6	7	8	9	
1°	0.68	0.42	0.75	0.48	0.40	0.27	0.77	0.75	0.61	0.570
2°	0.69	0.34	0.71	0.44	0.16	0.21	0.66	0.73	0.69	0.514
3°	0.38	0.18	0.48	0.33	0.07	0.14	0.29	0.49	0.44	0.311
4°	0.46	0.25	0.65	0.31	0.12	0.07	0.00	0.46	0.42	0.304
5°	0.41	0.17	0.39	0.25	0.06	0.16	0.34	0.45	0.37	0.289
Resultado:	0.73	0.47	0.71	0.49	0.33	0.21	0.70	0.60	0.43	0.518

5 Conclusão

Neste trabalho foram avaliadas diferentes técnicas para cada etapa do processamento de sinais de uma interface cérebro-computador baseada em imagética motora com quatro classes (mão esquerda, mão direita, pés e língua) usando a base de dados disponibilizada para a BCI Competition IV. Para tal, foi testado exhaustivamente o desempenho de combinações de várias dessas técnicas. Baseado nesses testes foi constatado que, para os métodos empregados, a seleção de atributos é a etapa mais impactante no desempenho final do sistema.

Usando as técnicas avaliadas, o processamento de sinais que alcançou o melhor desempenho consiste no pré-processamento dos sinais usando CAR, extração de atributos com FBCSP, seleção dos melhores atributos ranqueados através da informação mútua e classificação usando *Naïve Bayes*, com a acurácia de 71.20%.

Em um trabalho futuro pode-se buscar implementar a melhor combinação de técnicas de processamento identificada em um sistema *online*, para tal pode ser interessante reavaliar esses métodos utilizando uma janela de tempo menor que 5 segundos. Também pode-se realizar o processamento de forma adaptativa, usando uma fase de calibração ao invés de treinamento, de modo a permitir *feedback* o mais rápido possível, mesmo para usuários sem qualquer treinamento prévio.

Referências

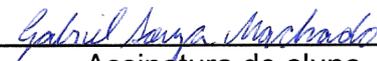
- ANG, K. K. et al. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in Neuroscience*, Frontiers Media SA, v. 6, 2012. Disponível em: <<https://doi.org/10.3389/fnins.2012.00039>>. Citado 2 vezes nas páginas 18 e 19.
- ANG, K. K. et al. Filter bank common spatial pattern (fbcsp) in brain-computer interface. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. [S.l.: s.n.], 2008. p. 2390–2397. ISSN 2161-4393. Citado 2 vezes nas páginas 18 e 24.
- ANG, K. K.; GUAN, C. EEG-based strategies to detect motor imagery for control and rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 25, n. 4, p. 392–401, apr 2017. Citado na página 11.
- BRUNNER, C. et al. Bci competition 2008 — graz data set a. Graz University of Technology. Citado na página 13.
- CARINO-ESCOBAR, R. I. et al. Classification of motor imagery electroencephalography signals using spiking neurons with different input encoding strategies. *Neural Computing and Applications*, Springer Nature, dec 2016. Citado na página 11.
- COHEN, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, SAGE Publications, v. 20, n. 1, p. 37–46, apr 1960. Disponível em: <<https://doi.org/10.1177/001316446002000104>>. Citado na página 32.
- DEUEL, T. A. et al. The encephalophone: A novel musical biofeedback device using conscious control of electroencephalogram (EEG). *Frontiers in Human Neuroscience*, Frontiers Media SA, v. 11, apr 2017. Citado na página 11.
- EFROYMSON, M. A. *Mathematical methods for digital computers*. New York: John Wiley & Sons, 1960. Citado na página 19.
- GRAIMANN, B.; PFURTSCHELLER, G.; ALLISON, B. (Ed.). *Brain-Computer Interfaces*. Springer Berlin Heidelberg, 2010. Disponível em: <<https://doi.org/10.1007/978-3-642-02091-9>>. Citado 3 vezes nas páginas 9, 10 e 16.
- JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001–. [Acessado em 1 de agosto de 2018]. Disponível em: <<http://www.scipy.org/>>. Citado na página 24.
- KRASKOV, A.; STÖGBAUER, H.; GRASSBERGER, P. Estimating mutual information. *Phys. Rev. E*, American Physical Society, v. 69, p. 066138, Jun 2004. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevE.69.066138>>. Citado na página 20.
- LOTZE, M.; HALSBAND, U. Motor imagery. *Journal of Physiology-Paris*, Elsevier BV, v. 99, n. 4-6, p. 386–395, jun 2006. Citado na página 9.
- LUDWIG, K. A. et al. Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *J. Neurophysiol.*, v. 101, n. 3, p. 1679–1689, Mar 2009. Citado na página 15.

- MCFARLAND, D. J.; WOLPAW, J. R. Brain-computer interfaces for communication and control. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 54, n. 5, p. 60, may 2011. Disponível em: <<https://doi.org/10.1145/1941487.1941506>>. Citado na página 10.
- MIAO, M. et al. Index finger motor imagery EEG pattern recognition in BCI applications using dictionary cleaned sparse representation-based classification for healthy people. *Review of Scientific Instruments*, AIP Publishing, v. 88, n. 9, p. 094305, sep 2017. Citado na página 12.
- MONDINI, V.; MANGIA, A. L.; CAPPELLO, A. EEG-based BCI system using adaptive features extraction and classification procedures. *Computational Intelligence and Neuroscience*, Hindawi Limited, v. 2016, p. 1–14, 2016. Citado na página 11.
- NICOLAS-ALONSO, L. F.; GOMEZ-GIL, J. Brain computer interfaces, a review. *Sensors*, MDPI AG, v. 12, n. 2, p. 1211–1279, jan 2012. Disponível em: <<https://doi.org/10.3390/s120201211>>. Citado na página 9.
- OLIPHANT, T. E. Python for scientific computing. *Computing in Science and Engg.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 9, n. 3, p. 10–20, maio 2007. ISSN 1521-9615. Disponível em: <<http://dx.doi.org/10.1109/MCSE.2007.58>>. Citado na página 24.
- OPPENHEIM, A. *Discrete-time signal processing*. Upper Saddle River: Pearson, 2010. ISBN 978-0-13-198842-2. Citado 2 vezes nas páginas 16 e 17.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 22 e 24.
- RAMOSER, H.; MULLER-GERKING, J.; PFURTSCHELLER, G. Optimal spatial filtering of single trial eeg during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, v. 8, n. 4, p. 441–446, Dec 2000. ISSN 1063-6528. Citado 2 vezes nas páginas 17 e 18.
- ROSSUM, G. van. *Python tutorial*. Amsterdam, 1995. Citado na página 24.
- THEODORIDIS, S. *Pattern recognition*. Burlington, MA London: Academic Press, 2009. ISBN 9781597492720. Citado 2 vezes nas páginas 21 e 23.
- ZHANG, H. *The Optimality of Naive Bayes*. 2004. Citado na página 23.

TERMO DE RESPONSABILIDADE

Eu, Gabriel Souza Machado, declaro que o texto do trabalho de conclusão de curso intitulado “*Análise do desempenho de técnicas de processamento de sinais para uma interface cérebro-computador baseada em imagética motora*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

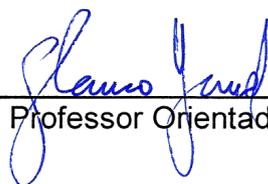
João Monlevade, 26 de Julho de 2018


Assinatura do aluno

DECLARAÇÃO DE CONFORMIDADE

Certifico que o aluno Gabriel Souza Machado, autor do trabalho de conclusão de curso intitulado “Análise do desempenho de técnicas de processamento de sinais para uma interface cérebro-computador baseada em imagética motora” efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, 26 de Julho de 2018.



Professor Orientador