

# Proposta de Modelo de Processamento e Classificação de Sinais de Eletroencefalografia Baseado em Imagética Motora

João Victor Soares de Assunção Santos  
Departamento de Engenharia Elétrica  
Universidade Federal do Piauí  
Teresina, Brasil  
vs.joaov71@gmail.com

José Maria Pires de Mesezes Junior  
Departamento de Engenharia Elétrica  
Universidade Federal do Piauí  
Teresina, Brasil  
josemenezesjr@gmail.com

**Resumo**—A construção de uma interface de comunicação direta entre o cérebro e o computador tem se mostrado uma tarefa desafiadora e que pode trazer inúmeras aplicações em controle. Em especial, ajudando na qualidade de vida de pessoas com disfunções no sistema neuromotor, ao permitir a criação de um novo canal de comunicação independente dos caminhos neurais. Este trabalho busca desenvolver um algoritmo de processamento digital de sinais cerebrais obtidos através de Eletroencefalografia (EEG), utilizando técnicas de Análise de Componentes Independentes (ICA), Análises Espectrais, padrões comuns espaciais (CSP) e banco de filtros (FBCSP), além de um classificador de Máquinas de Vetores de Suporte (SVM Linear). É implementada internamente uma solução para a classificação multiclasse utilizando um classificador binário. A integração dessas técnicas, implementadas individualmente, é feita pelo algoritmo, que foi construído pensando numa arquitetura que facilite a adição de outras técnicas, desde que trabalhando no formato de dados pré-definidos. Com as técnicas utilizadas, conseguiu-se taxas de acerto em um banco de dados de 4 classes, dependendo do usuário, de até 76,4% no melhor caso. Ademais, a forma de construção do algoritmo com a padronização dos dados, permite a constante melhoria do processamento com a fácil implementação de novas técnicas.

**Index Terms**—EEG, BCI, Imagética Motora, FBCSP, Processamento Digital

## I. INTRODUÇÃO

Técnicas de monitoramento das atividades cerebrais, como o Imageamento por Ressonância Magnética Funcional (fMRI), a Eletrocorticografia (ECoG), a Magnetoencefalografia (MEG) e a Eletroencefalografia (EEG), foram desenvolvidas nas últimas décadas principalmente com objetivos clínicos [1]. Nessa perspectiva, cada vez mais estudos sugerem a possibilidade da utilização dessas técnicas como um canal de comunicação do cérebro com o meio externo.

Uma interface cérebro-computador (BCI, *Brain Computer Interface*) é um tipo de inteligência computacional que alia software e hardware, com o objetivo de traduzir sinais cerebrais em informações úteis, de modo que possam ser utilizadas como um novo canal de comunicação do cérebro, expressando suas intenções para dispositivos como sintetizadores de fala, próteses ou computadores [2].

Dos métodos de aquisição de dados, destaca-se o EEG por ser um método considerado não invasivo, portátil, mais acessível e que apresenta menos riscos aos usuários, além de já ter se mostrado suficiente em muitas aplicações de BCIs [3], tendo até mesmo hardwares comerciais disponíveis.

Uma BCI é composta de algumas etapas que vão desde a aquisição de dados até o comando elétrico que será utilizado em algum dispositivo. Na etapa de aquisição dos dados, o sinal é digitalizado e enviado para a etapa seguinte: o processamento [1]. Na manipulação digital, os dados passam por uma etapa de pré-processamento, que pode ser constituída de uma remoção de ruídos e artefatos ou redução de dimensionalidade. Em seguida, técnicas de extração de características são aplicadas para reunir em vetores as melhores qualidades de um sinal que permitam a sua identificação pelos modelos de classificadores [1].

Após a classificação correta do sinal de entrada e a predição, da melhor forma possível, de a qual classe ele pertence (e.g. movimento da mão direita ou esquerda, movimento dos pés), o último passo é decidir qual comando deve ser enviado para o dispositivo de destino, que pode ser uma prótese, um computador, uma cadeira de rodas, entre outros.

Dessa forma, será apresentada uma proposta de classificação de sinais de EEG através de um algoritmo, que compreende desde a etapa de padronização dos dados até o modelo de classificadores para predição de sinais de EEG, baseados no paradigma de imagética motora (imaginação de movimentos). Além disso, a construção do código é pensada para ser flexível e permitir a implementação de diferentes técnicas, assim como nas várias etapas do modelo proposto, caso seja de interesse.

## II. METODOLOGIAS DE PROCESSAMENTO

### A. Análise de Componentes Independentes

Os sinais de EEG geralmente são contaminados por interferências do movimento ocular ou muscular. Dessa forma, se faz necessária a remoção de tais artefatos por técnicas de pré-processamento. Na literatura, é comum a utilização da Análise das Componentes Independentes (ICA, *Independent Component Analysis*), pois as perdas de informações dela

são mínimas quando comparada com outras, como a rejeição completa do sinal contaminado ou métodos com regressão linear [4].

A ideia central é extrair, dos sinais de EEG, uma componente resultante da contribuição de fontes que não são originadas da atividade cerebral, como, por exemplo, pulsos devido ao movimento dos olhos ou dos músculos da face e, então, subtraí-la do sinal original. Todo o equacionamento a seguir é visto em [5]. O ICA consiste em considerar um conjunto de  $n$  variáveis observadas ( $x_i$ ) e inferir que essas variáveis são, na verdade, uma combinação linear de  $n$  variáveis fonte ( $s_i$ ), ponderadas por coeficientes de mistura  $a_{ij}$ , de modo que:

$$x_i = \sum_{j=1}^n a_{ij} \times s_i, \quad (1)$$

em notação matricial:

$$x = A \times s. \quad (2)$$

A técnica ICA consiste em determinar o conjunto de sinais da fonte  $s$ , sendo que não se tem nenhuma informação prévia deles. Isso evidencia o problema do ICA, que é estimar uma matriz de separação  $W$ , tal que pode-se encontrar um conjunto de sinais  $y$ :

$$y = W \times x, \quad (3)$$

de forma que a diferença  $\|s - y\|$  seja mínima.

Para que seja possível estimar a matriz  $W$ , deve-se pressupor que as componentes da matriz  $s$  são estatisticamente independentes e não gaussianas, isso permite que sejam calculadas as componentes independentes, utilizando o conceito do teorema do limite central (TLC). Esse teorema diz que as misturas de duas variáveis aleatórias independentes sempre serão mais gaussianas do que os sinais originais.

Considera-se que, para encontrar uma única das componentes independentes, seja necessário aplicar a transformação  $y_i = bx$ , em que  $b$  é um vetor linha. Dessa maneira, tem-se que  $y_i = bAs$ , ou seja,  $y$  é uma combinação linear das componentes independentes com o vetor  $q = bA$ . Pelo TLC,  $y = qs$  sempre será mais gaussiano do que qualquer componente independente  $s$  a menos que  $y$  seja uma das componentes de  $s$ , pois, nesse caso, a não gaussianidade seria máxima. Então, se os valores de  $s$  fossem conhecidos, seria possível variar o valor de  $q$  até que uma das componentes fossem encontradas. Com a equação abaixo em mente:

$$y = qs = bx, \quad (4)$$

é possível estimar um vetor  $b$  que maximize a não gaussianidade de  $bx$ , isto é, consegue-se estimar uma componente independente  $y$  por meio dessa maximização, utilizando o teorema do Limite Central com o auxílio de alguma medida de gaussianidade, como a negentropia ou a curtose [5].

Uma vez separadas as componentes independentes, utilizou-se um periodograma de Welch para análise espectral e, com base na amplitude em baixas frequências, identificar aquelas componentes que estão contaminadas, pois elas que atingem picos visivelmente maiores em relação às saudáveis.

## B. Padrões Comuns Espaciais com Banco de Filtros

Na construção dos vetores de características representativos dos sinais, utilizou-se uma técnica chamada de FBCSP (*Filter Bank Common Spatial Patterns* – Padrões Espaciais Comuns com Banco de Filtros), uma aprimoração de outra chamada apenas de CSP (*Common Spatial Patterns*). A CSP se mostrou ser uma técnica muito eficiente para projetar dois sinais de classes diferentes em um espaço vetorial capaz de maximizar a diferença entre suas variâncias, podendo alcançar marcas de mais de 90% de acurácia [6] quando utilizada para classificação dos sinais de EEG binários.

Para este método de extração, é necessário que já se tenha um grande conjunto de amostras rotuladas de duas classes diferentes para alimentar o sistema no formato de matrizes, nas quais cada linha representa um dos canais e as colunas são as amostras no tempo. Toda a formulação matemática e fundamentação do método mostrado a seguir é apresentado em [6]. Uma vez que os dados estão organizados, calcula-se uma matriz de covariância espacial de todas as amostras como:

$$C = \frac{EE'}{\text{trace}(EE')}, \quad (5)$$

onde  $E$  é a matriz que representa o sinal. Uma vez calculadas as matrizes de projeção, determina-se a média de todas as matrizes da primeira classe ( $\overline{C_d}$ ) e da segunda classe ( $\overline{C_e}$ ), calculando, logo em seguida, a matriz de covariância espacial composta

$$C_c = \overline{C_d} + \overline{C_e}, \quad (6)$$

que se decompõe em autovetores e autovalores de ordem crescente  $C_c = U_c \lambda_c U_c'$ , estes sendo utilizados para definir uma transformação de branqueamento:

$$P = \sqrt{\lambda_c^{-1}} U_c'. \quad (7)$$

A transformação de branqueamento quando aplicada nas matrizes de covariância espacial média das duas classes  $S_e = PC_e P'$  e  $S_d = PC_d P'$ , geram as duas matrizes  $S$  que compartilham os mesmos autovetores ( $B$ ) quando são decompostas. Assim é possível criar uma matriz de decomposição  $W$  que quando aplicada nos sinais originais ( $E$ ) consegue gerar padrões espaciais que maximizam a diferença de variância entre as duas classes:

$$W = (B'P)'. \quad (8)$$

Como uma forma de reduzir a dimensionalidade dos vetores de características, só é necessário determinar a variância das primeiras e últimas linhas dos padrões espaciais ( $Z$ )

$$Z = W \times E, \quad (9)$$

então seleciona-se as  $m$  primeiras e  $m$  últimas linhas da matriz  $Z$ , calcula-se as variâncias dessas linhas, que são normalizadas com a escala logarítmica e formarão um vetor de características de tamanho  $2m$ :

$$f_p = \log \left[ \frac{\text{var}(Z_p)}{\sum_{i=0}^{2m} Z_i} \right]. \quad (10)$$

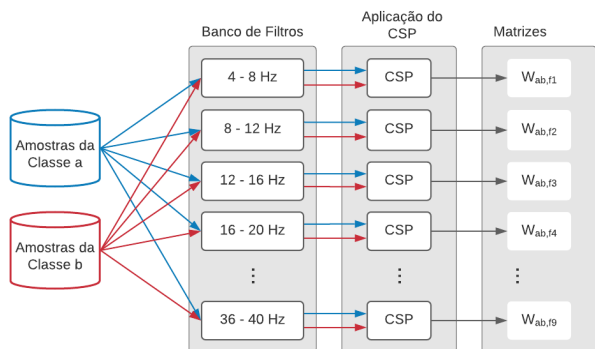


Fig. 1. Esquema de algoritmo FBCSP para geração das matrizes de projeção espacial.

Enquanto isso, o FBCSP consiste apenas em filtrar o sinal em várias bandas de frequência antes de aplicar o CSP, gerando uma matriz de projeção espacial  $W$  para cada banda de frequência do banco e, por consequência, um vetor de características para cada uma das bandas de frequência que serão usados em conjunto para treinamento do classificador. A separação do sinal por bandas de frequência entrega resultados substancialmente maiores que aqueles vistos no algoritmo CSP simples [7]. Um esquema do funcionamento do método FBCSP pode ser visto na Figura 1.

Neste trabalho, as matrizes de projeção espacial são identificadas, em primeiro lugar, pelas duas classes que elas são responsáveis por separar (no caso do esquema na Figura 4, as matrizes separam as classes a e b resultando nas matrizes  $W_{ab}$ ) e, depois, pela banda de frequência que ela assume, resultando nas matrizes para cada banda ( $W_{ab,f1}$ ,  $W_{ab,f2}$ ,  $W_{ab,f3}$ , ...,  $W_{ab,f9}$ ). Destaca-se que os sinais que foram utilizados para gerar as matrizes de projeção espacial são aqueles sinais de treinamento, ou seja, eles serão utilizados também para gerar os vetores de características com as etapas simbolizadas em (9) e (10), além de treinar algum algoritmo classificador com esses vetores.

### C. Classificação

O principal aspecto a ser apontado nessa etapa reflete diretamente do extrator de características escolhido para este trabalho (FBSCP), que funciona apenas para a separação de duas classes, enquanto na maioria dos problemas é necessária uma predição multiclasse.

Para demonstrar a forma como esse problema pode ser contornado, considera-se o banco de dados de 4 classes *IV BCI Competition - Dataset 2a* disponível em [8]. Ele contém sinais de 9 pessoas diferentes e amostras de 4 classes de movimentos diferentes: movimento da mão direita ( $r$ ), movimento da mão esquerda ( $l$ ), movimento dos pés ( $f$ ) e movimento da língua ( $t$ ).

Neste banco de dados é possível construir um total de seis matrizes de separação espacial, sendo elas  $W_{lr}$ ,  $W_{lf}$ ,  $W_{lt}$ ,  $W_{rf}$ ,  $W_{rt}$  e  $W_{ft}$ , com cada letra nos índices representando

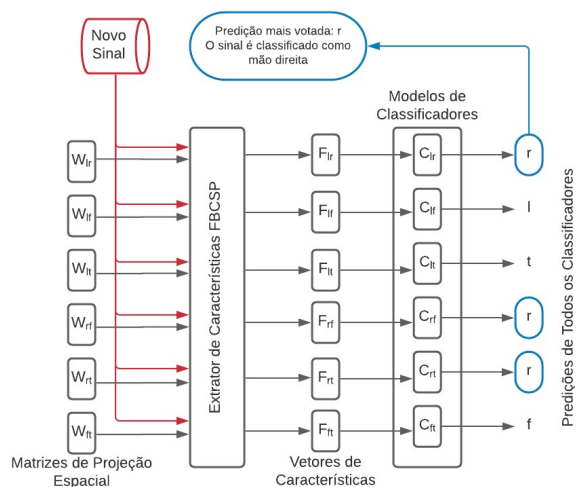


Fig. 2. Esquemático mostrando o funcionamento da predição através da votação dos modelos classificadores. Cada classificador dá a sua saída e aquela saída que mais se repete é a saída do sistema.

uma das 4 classes. Cada uma dessas matrizes é utilizada para gerar as características dos dados já rotulados do conjunto de teste e construir seis modelos de classificadores treinados com essas características.

Quando um sinal proveniente da imaginação da mão direita, por exemplo, é desconhecido pela BCI e deve ser classificado, bastaria que ele passasse por qualquer das três matrizes de projeção espacial que conseguem abranger a imaginação da mão direita, que são:  $W_{lr}$ ,  $W_{rf}$  e  $W_{rt}$ , mas como o objetivo é determinar a classe do sinal, não há como direcioná-lo para as matrizes que irão corretamente extrair as características.

Quando um sinal novo é apresentado para ser classificado, ele irá passar por todas as matrizes de projeção espacial, gerando seis vetores de características, que passarão pelos seis modelos classificadores citados anteriormente, com cada um informando a sua predição. O palpite que ocorrer mais vezes dentre os classificadores é o tido como saída final do sistema. Esse formato de classificação, na qual são feitas comparações de todas as classes duas a duas, é chamada de modelo um contra um (1x1) e seu esquemático é visto na Figura 2.

Os classificadores da Figura 2 são modelos construídos a partir dos vetores de características do conjunto de treino, os mesmos dados que são utilizados para gerar as matrizes de separação espacial passam pela extração e alimentam o modelo. Podem ser utilizados diversos tipos já existentes nas bibliografias, como redes neurais, KNN, máquinas de vetores de suporte (SVM), entre outros. Neste trabalho, foram implementados classificadores que utilizam SVM Linear.

Vale destacar que os dados de cada sujeito não são misturados em nenhum momento do algoritmo, fazendo necessário o treinamento de classificadores individuais para cada paciente. Isso garante uma maior simplicidade das técnicas utilizadas, pois respeita a individualidade que cada paciente possui no que se refere à manifestação das suas atividades cerebrais [2].

### III. DESCRIÇÃO DO ALGORITMO

O primeiro passo realizado pelo algoritmo construído é uma formatação dos dados (foi escolhido o formato .fif, por ter bastante suporte na biblioteca MNE Tools), de forma que o sistema reconheça os sinais corretamente. Tendo em vista que ainda não existe um formato padrão para armazenamento de dados de EEG, mas sim vários, que são utilizados por diferentes empresas com headsets diferentes [2], considera-se necessária uma padronização dos dados, para que haja uma interface entre qualquer que seja o formato dos dados originais para o formato que é utilizado em todo o algoritmo.

Em seguida os sinais passam por um processo de amostragem, onde serão extraídos apenas as janelas de gravação com informações úteis: as gravações possuem vários ensaios, cada um com uma pequena janela de onde são retiradas as amostras que possuem a imaginação de movimento. Essas janelas são organizadas por classes e salvas nas pastas "Épocas de Cada Classe", uma para teste e outra para treino, conforme a Figura 3.

Ainda na Figura 3 são representadas todas as pastas geradas pelo algoritmo ao longo do treinamento da BCI, que ficam armazenadas dentro de um ficheiro dedicado para cada usuário. Apenas a primeira pasta, "Dados de Treinamento Originais", não é criada pelo código, pois são os arquivos iniciais e podem estar em qualquer padrão comercial, devendo passar por uma formatação, citada anteriormente.

As subpastas da Figura 3 referentes ao método de comparação (1x1, 2x2, 1xAll) foram criadas apenas com a previsão de implementação, somente com o formato 1x1 realmente tendo sido implementado no algoritmo até a escrita deste artigo, mas com a acessibilidade de adição dos novos modelos através de interfaces com a classe já criada no algoritmo. O mesmo vale para os modelos classificadores, embora já tenha sido implementado o classificador SVM Linear, o programa está acessível para a inclusão de novos métodos,

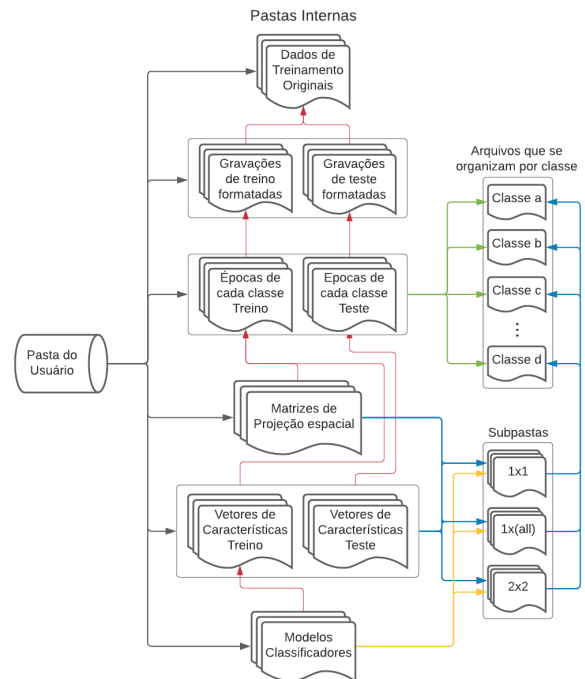


Fig. 3. Representação esquemática do sistema de arquivos utilizado. Setas em vermelho, especificamente, indicam a origem dos dados daquela pasta. Outras cores indicam a existência de subpastas dentro das pastas internas, quando for o caso.

através da interface que os mantém unidos ao código.

O diagrama de classes do algoritmo pode ser visto na Figura 4 e mostra com mais clareza a possibilidade de substituir elementos do código, tanto através de interfaces, como removendo alguma das metodologias (por exemplo até mesmo o extrator de características FBCSP) e substituindo por outra, conforme desejado.

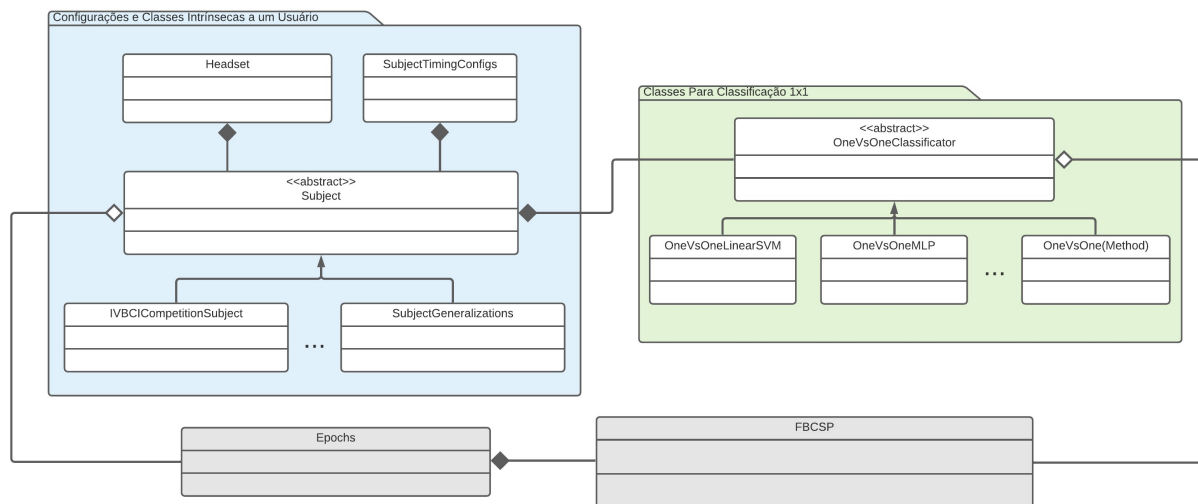


Fig. 4. Diagrama de Classes e Métodos do algoritmo que foi construído.

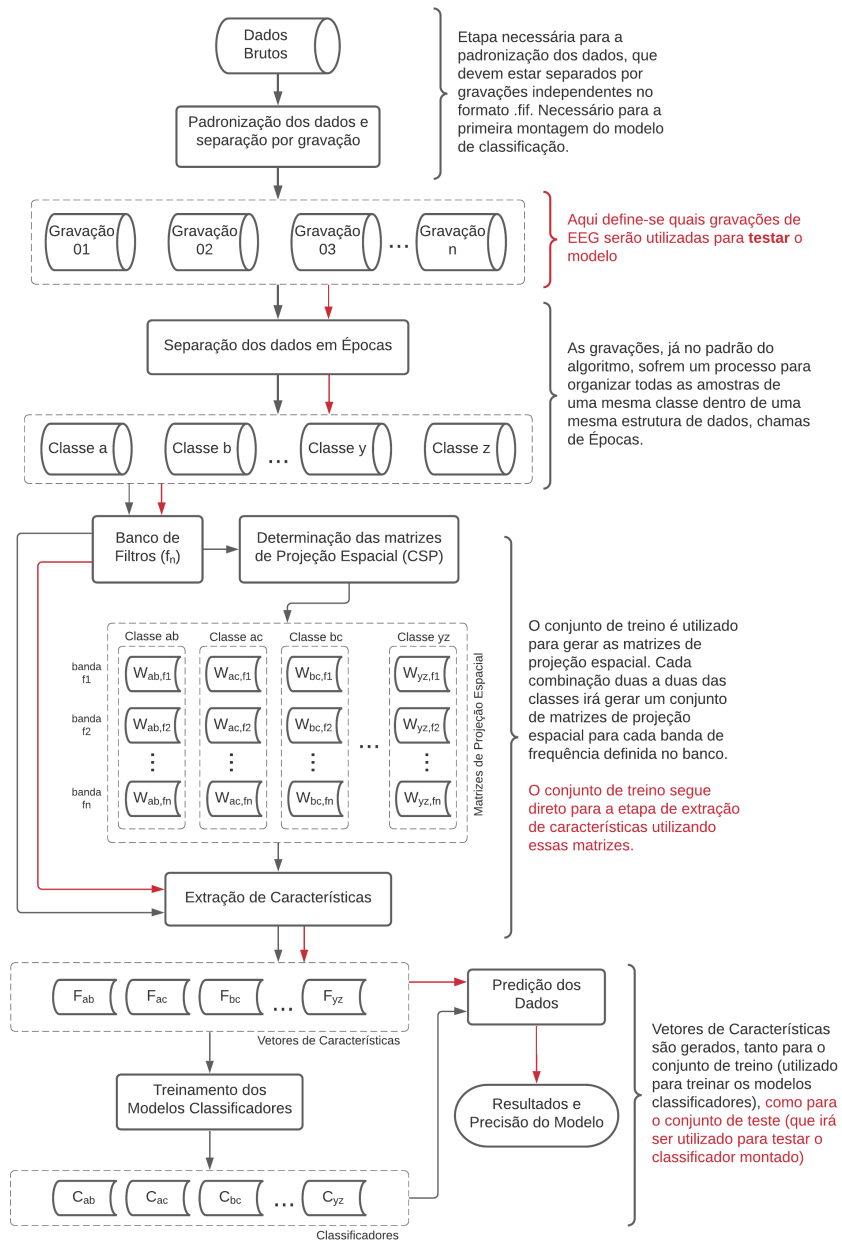


Fig. 5. Fluxograma do algoritmo construído, mostrando o fluxo e o caminho que os dados seguem até concluir o treinamento da etapa de processamento da BCI. As setas em preto mostram o fluxo seguido pelo conjunto de treino, enquanto as setas vermelhas mostram o fluxo do conjunto de teste ou para classificação.

As classes do algoritmo funcionam em volta da classe principal "Subject", que representa um paciente e a ele estão associados, por uma agregação, um "Headset", com as especificações de aquisição; e um "SubjectTimingConfigs" contendo informações de amostragem das janelas de dados. As suas subclasses implementam os métodos que formatam os dados originais (lembrando que podem estar em qualquer formato, daí cabe ao programador escrever as conversões para o formato .fif) para que possam ser utilizados pelo algoritmo.

A classe "Epochs" reúne um conjunto de amostras de

uma mesma classe, podendo ser criadas instâncias de quantas classes forem necessárias para cada paciente. Observando o método de comparação do classificador e utilizando as épocas de um paciente, a classe "FBCSP" se encarrega de gerar os vetores de características.

A interface de classificação (em verde na Figura 4) utiliza uma das técnicas disponíveis nas suas subclasses para modelar os classificadores. Podem ser definidas várias técnicas de classificadores nas subclasses, inclusive possibilitando a construção de um *ensemble* e selecionando o classificador com

os melhores desempenhos em cada caso (paciente, classes, headsets).

Com isso estabelecido, o algoritmo fica com seu funcionamento totalmente descrito após o fluxograma da Figura 5, mostrando o fluxo de dados e os procedimentos adotados com dados de quantas classes se queira treinar a BCI. Embora possa parecer que a figura mostra apenas a sequência de treinamento e de testagem, não realizando a classificação dos dados desconhecidos, basta passar o sinal desconhecido e que deve ser classificado na mesma *pipeline*, seguida pelo conjunto de treino (em vermelho) e mostrado com mais profundidade na Figura 1.

Todos os passos mostrados no fluxograma da Figura 5 são executados pelo algoritmo. Cabe ao programador criar novas subclasses da interface "Subject", se tiver que utilizar algum padrão diferente de dados para realizar a conversão para o formato utilizado no algoritmo. Também criar instâncias de "Headsets" diferentes conforme suas especificações e configurar a temporização das amostras com instâncias da classe "SubjectTimingConfigs", para que o código consiga identificar as janelas com informações úteis.

Uma vez com os dados formatados, o código irá realizar toda a amostragem, construir as matrizes de projeção espacial, os modelos de classificadores e disponibilizar, dentro da classe subject, métodos para a classificação de novos dados, utilizando o modelo construído.

O código fonte deste projeto está totalmente hospedado na plataforma de controle de versão e difusão de iniciativas open source github, na url [https://github.com/vsJoao/bci\\_project](https://github.com/vsJoao/bci_project). Destaca-se ainda que todo o código fonte foi escrito em Python com o auxílio de pacotes como *MNE Tools* [9], *NumPy* [10], *Pandas* [11], *Scikit Learn* [12] e *SciPy* [13].

#### IV. RESULTADOS E DISCUSSÕES

Para testar o algoritmo, foi utilizado o banco de dados de [4] já descrito anteriormente. Não foram utilizados os dados da pessoa identificada com o número 04 por apresentar muitos sinais defeituosos. Em quantidades, o banco contém um total de 6 gravações por pessoa, com cada gravação contendo 48 amostras rotuladas, totalizando 288 janelas para o treinamento do algoritmo. O banco contém, ainda, outro conjunto do mesmo tamanho para teste.

A primeira visualização do funcionamento algoritmo se dá na construção dos vetores de características. Na Figura 6, foram impressos os gráficos de dispersão da banda de frequência de 8-12Hz da pessoa número 01 (lembrando que, do método utilizado, são gerados seis conjuntos de vetores de características), resultante da aplicação da técnica de FBCSP explorada anteriormente. Mesmo o método separando apenas duas classes por vez, é visível a sua eficiência nessa tarefa, facilitando o trabalho do classificador posteriormente.

Como métrica de desempenho do modelo, foi utilizado o índice *kappa*, uma métrica de concordância apresentada em [14] e utilizada, principalmente, em casos de diagnósticos médicos, mas que também pode ser utilizada para medição de desempenho de classificadores multiclasse. Quanto maior

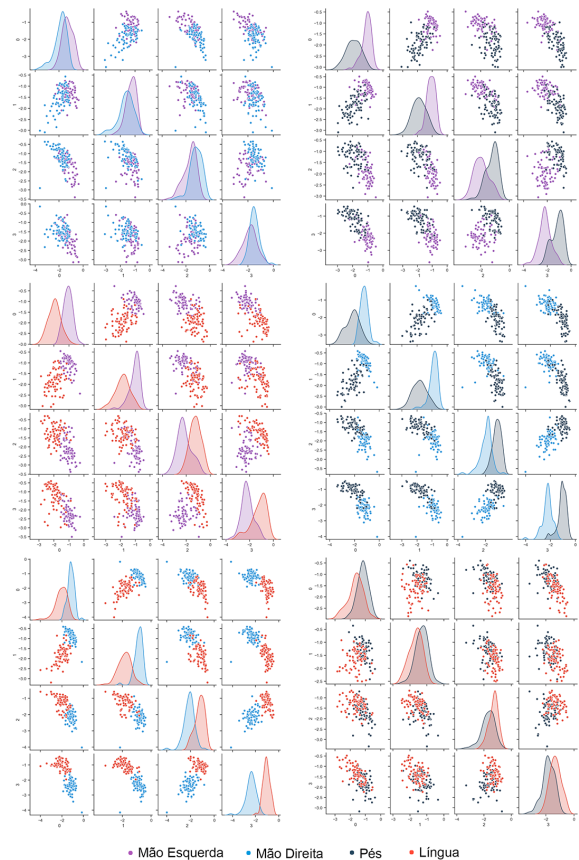


Fig. 6. Gráficos de dispersão da banda 8-12Hz impressos dois a dois para classificação

a coerência do classificador, mais se aproxima do valor 1. E quanto mais aleatória é a classificação, mais o indicador kappa se aproxima de zero. Em [15] é apresentada uma Tabela (mostrada na Tabela I) para interpretar o índice kappa conforme seu valor.

Na classificação, já era esperado que cada um dos indivíduos fossem ter resultados diferentes. Conseguir controlar os próprios sinais eletrofisiológicos é uma habilidade que deve ser desenvolvida dentre aqueles que desejam utilizar uma BCI com máxima eficiência, uma vez que controlar os músculos é diferente de projetar essas intenções de movimentos em uma BCI [1].

Tendo isso em vista, analisou-se os resultados da classificação de cada um dos 8 convidados através da Figura 7, que mostra as taxas de acerto obtidas para cada um dos convidados e também o índice kappa.

Observando a Figura 7, o classificador não conseguiu separar de forma totalmente satisfatória os dados dos sujeitos de número 5 e 6, agindo de forma quase que aleatória (uma situação completamente aleatória iria gerar uma taxa de acerto em torno de 0,25 e um índice kappa próximo de zero). Conforme a Tabela I, os pacientes 2, 5 e 6 conseguiram apenas resultados razoáveis, enquanto todos os outros conseguiram classificações de moderadas a substanciais pelo índice kappa.

A publicação do banco de dados em [8] revela apenas os

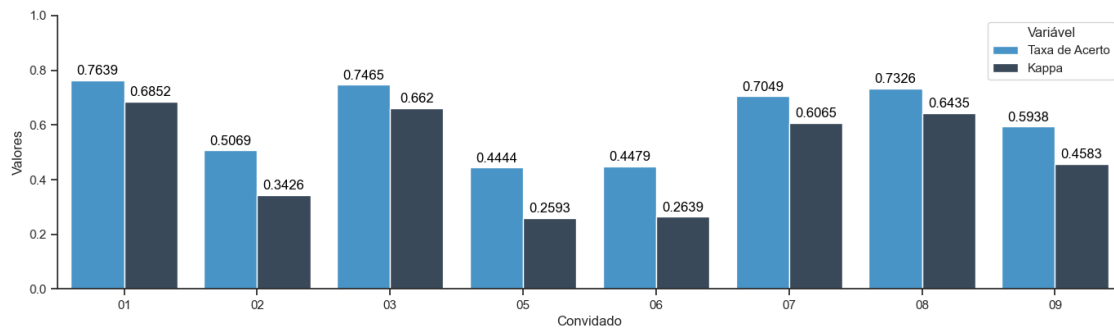


Fig. 7. Taxas de acerto e Índice Kappa por Convidado.

TABELA I  
INTERPRETAÇÃO DO ÍNDICE KAPPA CONFORME SEU VALOR. [15]

Kappa	Interpretação
menor que zero	insignificante
Entre 0 e 0,2	fraca
Entre 0,21 e 0,4	razoável
Entre 0,41 e 0,6	moderada
Entre 0,61 e 0,8	substancial
Entre 0,81 e 1	quase perfeita

detalhes do procedimento de aquisição, mas não a experiências que as pessoas poderiam ter na utilização de uma BCI antes da realização do experimento, impossibilitando uma análise relacionando a habilidade que a pessoa possui ao manusear uma BCI aos resultados que ela obteve neste trabalho.

A Tabela II apresenta um resumo de todos os trabalhos da competição com suas metodologias abordadas e os resultados. O primeiro colocado utilizou FBCSP como extrator, mas com um classificador de Bayes e janelas de Parzen e a estratégia um-contra-todos para definir os modelos multiclasse. Já o segundo colocado utilizou apenas a técnica CSP com o sinal filtrado em 8-30Hz, também com um classificador Bayesiano e Análise Discriminante Linear para redução da dimensionalidade. O terceiro, no pré-processamento, utilizou regressão linear com os canais de EOG, CSP como extrator e SVM como classificador na estratégia de dividir as classes em duas hierarquias, primeiro um dois-contra-dois e depois um-contra-um [8].

Identifica-se uma semelhança (dos competidores) com este trabalho na utilização do algoritmo CSP já consolidado na área, que possui variações como o FBCSP e é utilizado em todos os trabalhos aqui citados. Como destaques do algoritmo desenvolvido, a utilização de ICA com auxílio de um periodograma de Welch como um pré-processamento, além da estratégia de classificação multiclasse um-contra-um e votação entre os modelos SVM são as técnicas que formam a identidade deste trabalho em relação aos outros citados no parágrafo anterior.

Os resultados foram resumidos na Tabela III utilizando o índice kappa como métrica de comparação de todos os competidores e este trabalho. As técnicas implementadas neste tra-

balho conseguiram se posicionar em terceiro na classificação dos trabalhos inscritos na competição, com valores próximos aos dos dois primeiros. Mesmo com aqueles resultados tendo sido publicados em 2008 [8], essas técnicas ainda são consideradas estado da arte até hoje [16].

## V. CONCLUSÕES

Com este trabalho, conseguiu-se propor um modelo de processamento que funcionou com um desempenho semelhante às melhores propostas da competição utilizando um conjunto de técnicas disponível na literatura discutida.

Apresenta, ainda, a melhoria de ter uma arquitetura com códigos funcionando de forma dinâmica, com a possibilidade de adaptação para os diversos formatos de arquivos e headsets, além de métodos através das interfaces criadas com a ajuda da orientação a objetos. Conseguiu-se um sistema que está pronto para implementações de especializações de classes e testes em tempo real.

Em trabalhos futuros objetiva-se colocar colocar em prática a interligação do algoritmo com métodos de aquisição de dados de forma automática, com um headset comercial e, depois a criação de rotinas que automatizem o treinamento da BCI mediante a apresentação de novos dados, sendo gravados em tempo real ao mesmo tempo que a saída da BCI passará a ser utilizada como um comando de controle através do computador.

## REFERÊNCIAS

- [1] WOLPAW, Jonathan R et al. **Brain-computer interfaces for communication and control**. *Clinical Neurophysiology*, [s.l.], v. 113, n. 6, p.767-791, jun. 2002. Elsevier BV. [http://dx.doi.org/10.1016/s1388-2457\(02\)00057-3](http://dx.doi.org/10.1016/s1388-2457(02)00057-3).
- [2] NICOLAS-ALONSO, Luis Fernando; GOMEZ-GIL, Jaime. Brain Computer Interfaces, a Review. *Sensors*, [S.L.], v. 12, n. 2, p. 1211-1279, 31 jan. 2012. MDPI AG. <http://dx.doi.org/10.3390/s120201211>.
- [3] PFURTSCHELLER, G.; NEUPER, C.. Motor imagery and direct brain-computer communication. *Proceedings Of The Ieee*, [S.L.], v. 89, n. 7, p. 1123-1134, jul. 2001. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/5.939829>.
- [4] FATOURECHI, Mehrdad et al. EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, [s.l.], v. 118, n. 3, p.480-494, mar. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.clinph.2006.10.019>.

TABELA II  
TÉCNICAS E METODOLOGIAS UTILIZADAS PELOS DEMAIS COMPETIDORES NA IV BCI COMPETITION 2A

Posição	Competidor	Pré-Processamento	Características	Classificador
1.	K. K. Ang	Remoção de artefatos com filtros passa-faixa	FBCSP com paradigma um-contra-todos.	Classificador Naïve Bayes.
2.	L. Guangquan	CSP com sinal filtrado em 8-30Hz.	Variância logarítmica das melhores oito componentes.	LDA (Discriminante Linear) e classificador Bayesiano.
3.	W. Song	Downsampling, remoção de EOG com regressão linear, filtro passa-faixa entre 8-25Hz e seleção de canais baseado em um método recurso de eliminação de canais.	CSP.	Classificador multiclasse utilizando SVM em árvore com duas hierarquias e combinação com estratégia de votação.
4.	D. Coyle	Aplicação de CSP em sinais que passaram pela técnica de <i>Neural Time Series Prediction Preprocessing</i> (NTSPP) espectralmente filtrados.	Variância logarítmica dos sinais em janelas de 1s.	Os dois melhores dentre 3 variantes de LDA e mais duas variantes de SVM para cada sujeito.
5.	J. Wu	Downsampling, seleção dos eletrodos próximos a c3 e c4, filtro passa faixa entre 8-25Hz, remoção de EOG com regressão linear.	CSP.	Conjunto de classificadores multiclasse utilizando três SVM e duas hierarquias de classes.

TABELA III

RESULTADOS DA VI BCI COMPETITION COM RESULTADOS DESTA TRABALHO (LINHA DE ÍNDICE E.T.) E MÉTRICA UTILIZANDO O ÍNDICE KAPPA.

posição	kappa	1	2	3	4	5	6	7	8	9
1.	<b>0.57</b>	0.68	<b>0.42</b>	<b>0.75</b>	<b>0.48</b>	<b>0.40</b>	<b>0.27</b>	<b>0.77</b>	<b>0.75</b>	0.61
2.	0.52	<b>0.69</b>	0.34	0.71	0.44	0.16	0.21	0.66	0.73	<b>0.69</b>
E.T.	0.49	<b>0.69</b>	0.34	0.66	-	0.26	0.26	0.61	0.64	0.46
3.	0.31	0.38	0.18	0.48	0.33	0.07	0.14	0.29	0.49	0.44
4.	0.30	0.46	0.25	0.65	0.31	0.12	0.07	0.00	0.46	0.42
5.	0.29	0.41	0.17	0.39	0.25	0.06	0.16	0.34	0.45	0.37

- [5] VIGÁRIO, Ricardo Nuno. Extraction of ocular artefacts from EEG using independent component analysis. **Electroencephalography And Clinical Neurophysiology**, [s.l.], v. 103, n. 3, p.395-404, set. 1997. Elsevier BV. [http://dx.doi.org/10.1016/s0013-4694\(97\)00042-8](http://dx.doi.org/10.1016/s0013-4694(97)00042-8).
- [6] RAMOSER, H.; MULLER-GERKING, J.; PFURTSCHELLER, G.. Optimal spatial filtering of single trial EEG during imagined hand movement. **Ieee Transactions On Rehabilitation Engineering**, [S.L.], v. 8, n. 4, p. 441-446, dez. 2000. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/86.895946>.
- [7] ANG, Kai Keng; CHIN, Zhang Yang; ZHANG, Haihong; GUAN, Cuntai. Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. **2008 Ieee International Joint Conference On Neural Networks** (Ieee World Congress On Computational In, [S.L.], p. 2390-2397, jun. 2008. IEEE. <http://dx.doi.org/10.1109/ijcnn.2008.4634130>.
- [8] TANGERMANN, Michael; et. al. Review of the BCI Competition IV. **Frontiers In Neuroscience**, [s.l.], v. 6, p. 0-55, 2012. Frontiers Media SA. <http://dx.doi.org/10.3389/fnins.2012.00055>.
- [9] GRAMFORT, Alexandre. MEG and EEG data analysis with MNE-Python. **Frontiers In Neuroscience**, [s.l.], v. 7, p.1-267, jul. 2013. Frontiers Media SA. <http://dx.doi.org/10.3389/fnins.2013.00267>.
- [10] HARRIS, Charles R.; et. al. Array programming with NumPy, **Nature**, 585, 357–362 (2020), DOI:10.1038/s41586-020-2649-2.
- [11] MCKINNEY, Wes. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010).
- [12] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- [13] VIRTANEN, Pauli; et. al. (2020) SciPy 1.0: **Fundamental Algorithms for Scientific Computing in Python**. *Nature Methods*, 17(3), 261-272.
- [14] COHEN, Jacob. A Coefficient of Agreement for Nominal Scales. **Educational And Psychological Measurement**, [S.L.], v. 20, n. 1, p. 37-46, abr. 1960. SAGE Publications. <http://dx.doi.org/10.1177/001316446002000104>.
- [15] LANDIS, J. Richard; KOCH, Gary G.. The Measurement of Observer Agreement for Categorical Data. **Biometrics**, [S.L.], v. 33, n. 1, p. 159-174, mar. 1977. JSTOR. <http://dx.doi.org/10.2307/2529310>.
- [16] PADFIELD, Natasha; ZABALZA, Jaime; ZHAO, Huimin; MASERO, Valentin; REN, Jinchang. EEG-Based Brain-Computer Interfaces Using Motor-Imagery: techniques and challenges. **Sensors**, [S.L.], v. 19, n. 6, p. 1423-1457, 22 mar. 2019. MDPI AG. <http://dx.doi.org/10.3390/s19061423>.