

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

WILLIAN MOREIRA DA SILVA

**REVISÃO E COMPARAÇÃO DE CLASSIFICADORES EM UM
EXPERIMENTO UTILIZANDO INTERFACE
CÉREBRO-COMPUTADOR**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO
2016

WILLIAN MOREIRA DA SILVA

**REVISÃO E COMPARAÇÃO DE CLASSIFICADORES EM UM
EXPERIMENTO UTILIZANDO INTERFACE
CÉREBRO-COMPUTADOR**

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Rodrigo Hübner

**CAMPO MOURÃO
2016**

Resumo

Moreira da Silva, Willian. Revisão e Comparação de Classificadores em um Experimento Utilizando Interface Cérebro-Computador. 2016. 38. f. Trabalho de Conclusão de Curso (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2016.

Uma Interface Cérebro-Computador (ICC) pode ser utilizada para controlar diversos dispositivos eletrônicos, como computadores, smartphone e os movimentos de um robô, por exemplo. A construção de uma ICC pode ser dividida em cinco etapas: aquisição de sinais, pré-processamento, extração de características, classificação e aplicação. Para a etapa de classificação são utilizados classificadores com o objetivo de detectar as intenções do usuário para que sejam executadas na aplicação desejada. Porém, existe uma grande variedade de classificadores disponíveis, sendo que nem todos podem ter um bom desempenho para a aplicação desejada. Nesse contexto, foram selecionados os classificadores *Support Vector Machine* (SVM) e o *Linear Discriminant Analysis* (LDA) para serem avaliados utilizando uma aplicação ICC. A aplicação utilizada foi o P300 **Speller**, que é um soletrador implementado no software **OpenViBE**. Ao longo do trabalho são descritos os hardwares para aquisição de sinais e também os cenários implementados no software **OpenViBE**, utilizados para criar o soletrador seguindo o paradigma P300. Foram realizados dois experimentos denominados Caso 1, em que o algoritmo LDA acertou 7% dos alvos e o Caso 2, em que os algoritmos LDA e SVM acertaram 16% e 8%, respectivamente.

Palavras-chaves: Interfaces Cérebro-Computador, P300, LDA, SVM

Abstract

Moreira da Silva, Willian. Review and Comparison of Classifiers in a experiment using Brain-computer interface. 2016. 38. f. Graduation Project (Bachelor of Computer Science), Federal University of Technology – Paraná. Campo Mourão – PR – Brazil, 2016.

A Brain-Computer Interface (BCI) can be used to control various electronics devices such as computers, smartphones and movements of a robot, for example. The construction of a BCI can be divided into five stages: signal acquisition, preprocessing, feature extraction, classification and application. For the classification stage classifiers are used to detect the user's intentions that are performed in the desired application. However, there is a wide range of classifiers available, and not all classifiers can have a good performance for the desired application. In this context, the Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) classifiers were selected to be evaluated using an ICC application. The application used was P300 Speller, which is a speller implemented in OpenViBE software. Throughout the work, we describe the hardware for signal acquisition and also the scenarios implemented in OpenViBE software used to create the speller following the paradigm P300. Two experiments were performed denominated Case 1, wherein the LDA algorithm hit 7% of targets and Case 2, wherein the LDA and SVM algorithms hit 16% and 8%, respectively.

Keywords: Brain-Computer Interfaces, P300, LDA, SVM

Lista de figuras

1.1 Fluxograma das etapas de uma Aplicação de Interface Cérebro-Computador. Adaptado de: Ramadan <i>et al.</i> (2015)	6
2.1 Sistema 10-20 para o posicionamento dos eletrodos	10
2.2 Ilustração do funcionamento do SVM. Adaptado de: Marshall <i>et al.</i> (2013)	14
2.3 Ilustra a arquitetura de um MLP	15
4.1 Equipamentos de eletrencefalografia	19
4.2 Matriz de caracteres do P300 Speller	20
4.3 Cenário de aquisição de sinais para o treinamento dos classificadores	20
4.4 Cenário utilizado para o treinamento dos classificadores	23
4.5 Parte 1 do cenário de Teste Online	25
4.6 Parte 2 do cenário de Teste Online	25
4.7 Parte 3 do cenário de Teste Online	26
4.8 Cenário utilizado para o teste online dos classificadores	27
4.9 Cenário utilizado para o treinamento do filtro espacial xDAWN	28
4.10 Cenários utilizados para treinamento e teste utilizando o filtro xDAWN	29
4.11 Fluxograma de Execução dos cenários do P300 Speller	29

Lista de tabelas

4.1	Comparativo entre alguns hardwares disponíveis. A diferença entre os as duas versões do <code>OpenBCI</code> é que o <code>OpenBCI R&D Kit</code> vem com a placa expansora para dezesseis canais.	19
4.2	Campos e valores utilizados para a configuração do <code>P300 Speller Stimulator</code>	22
4.3	Campos e valores utilizados para a configuração do <code>Target Letter Generation</code>	22
4.4	Campos e valores utilizados para a configuração do classificador <code>LDA</code>	24
4.5	Campos e valores utilizados para a configuração do classificador <code>SVM</code>	24
4.6	Resultados obtidos para o <code>Caso 1</code>	30
4.7	Resultado do primeiro teste utilizando os cenários descritos no <code>Caso 2</code>	30
4.8	Resultados das execuções das sequências de caracteres <code>BRAIN</code> , <code>LETRA</code> e <code>UTFPR</code>	31
4.9	Resultados das execuções das sequências de caracteres dos arquivos <code>online2</code> e <code>online20</code>	31
4.10	Média das execuções para cada conjunto utilizado no treino	31

Sumário

1	Introdução	6
1.1	Objetivo	7
1.2	Apresentação do texto	7
2	Fundamentação Teórica	8
2.1	Interface Cérebro-Computador	8
2.2	Aquisição do Sinal	9
2.3	Paradigmas Baseados em Interface Cérebro-Computador	11
2.3.1	Atenção Seletiva	11
2.3.2	<i>Motor Imagery</i>	12
2.4	Pré-Processamento	12
2.5	Extração de Características	12
2.6	Classificação	13
2.6.1	<i>Support Vector Machine (SVM)</i>	13
2.6.2	<i>Multilayer Perceptron (MLP)</i>	14
2.6.3	<i>Linear Discriminant Analysis (LDA)</i>	15
2.7	Aplicação de uma ICC	15
3	O Estado da Arte	16
4	Experimentações	18
4.1	Caso 1: Desenvolvimento do cenário P300 Speller	19
4.1.1	Descrição do Cenário de Aquisição de Sinais	20
4.1.2	Descrição do Cenário de Treino dos Classificadores	22
4.1.3	Descrição do Cenário de Teste Online	24
4.2	Caso 2: Desenvolvimento do cenário P300 Speller xDAWN	28
4.3	Discussão dos Resultados	29
5	Conclusões e Trabalhos Futuros	33
	Referências	34

Introdução

A forma como nos comunicamos e interagimos com o mundo é tão natural, que não paramos para pensar a dificuldade de algumas pessoas que, por algum motivo, perderam alguma dessas capacidades. A perda parcial ou total de movimentos do corpo, independente do motivo, torna ações simples do nosso cotidiano em verdadeiros desafios. Maiores são os desafios se pensarmos na dependência que os computadores, *smartphones* e demais dispositivos eletrônicos geram e fazem parte do cotidiano de todas as pessoas.

Uma Interface Cérebro-Computador (ICC) pode ser descrita como uma comunicação direta entre o cérebro e o computador (Ramadan *et al.*, 2015). Em sistemas que utilizam ICC, os sinais gerados pelo cérebro são analisados para determinar as intenções do usuário. Para isso é necessário passar por diversas etapas complexas, desde a aquisição dos sinais até a classificação destes para a geração ou manipulação de uma ação. A Figura 1.1 mostra todas as etapas necessárias para construção de uma ICC.

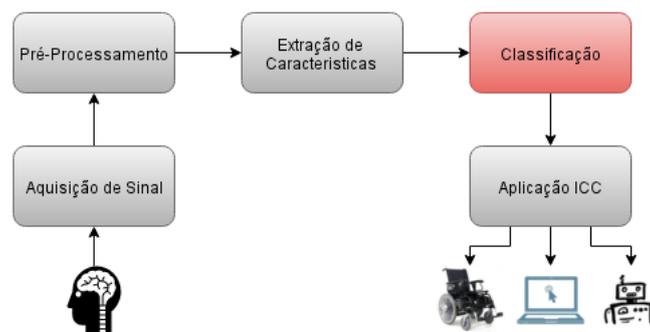


Figura 1.1. Fluxograma das etapas de uma Aplicação de Interface Cérebro-Computador. Adaptado de: Ramadan *et al.* (2015)

Cada uma das etapas apresentadas na Figura 1.1 podem ser expandidas em inúmeros campos de pesquisa. A etapa que terá maior foco nesse trabalho é a de classificação dos sinais gerados por um indivíduo. Essa etapa é responsável por dar sentido aos sinais adquiridos na etapa de aquisição de sinal, detectando por exemplo, a intenção do usuário de acordo com determinados padrões encontrados nos sinais gerados pelo cérebro. Contudo existe uma grande diversidade de algoritmos que podem ser utilizados para classificar o sinal obtido, cada um com suas vantagens, desvantagens e diferentes abordagens.

1.1. Objetivo

No contexto de classificação do sinal mensurado, o desenvolvimento desta monografia tem como objetivo determinar a qualidade dos classificadores *Linear Discriminant analysis* (LDA) (Fisher, 1936) e *Support Vector Machine* (SVM) (Vapnik, 1995). Para isso, foram observados os resultados produzidos por estes classificadores por meio de condições impostas em um experimento prático.

O experimento realizado para esta monografia consiste em implementar todas as etapas necessárias para a construção de uma interface cérebro-computador, porém, com ênfase na fase de classificação. Para isso foram utilizados hardwares e softwares especializados para a concepção dessas etapas. Para aquisição de sinais foram utilizados o equipamento **OpenBCI**¹ e o **Olimex EEG-SMT**², que são hardwares apropriados e de baixo custo, para detecção de atividade elétrica cerebral. Já para as etapas de pré-processamento, extração de características, classificação e aplicação, foi utilizado o **OpenViBE**³, que é uma plataforma de desenvolvimento para ICC.

O **OpenViBE** disponibiliza a implementação de alguns cenários, como exemplo, o **P300 Speller** no qual o objetivo é o usuário se concentrar em caracteres (dispostos em uma matriz na tela) e então o sistema detecta em qual letra o usuário está olhando. A partir desse cenário, foram realizadas modificações para adequá-lo ao experimento proposto. As principais modificações ocorreram na parte que é realizado a classificação, na qual adaptamos o cenário para torná-lo utilizável pelos classificadores que foram avaliados nesta monografia.

O protocolo do experimento consiste em uma sessão para aquisição de sinais para compor o conjunto de treinamento dos classificadores e mais três sessões que compõem o conjunto de testes. Em cada sessão são utilizados conjuntos de letras diferentes, sendo o conjunto de letras escolhido para a sessão de treino é **COMPUTER** e para as sessões de teste são: **UTFPR**, **BRAIN** e **LETRA**. Estas sessões foram realizadas para cada indivíduo testado com diferentes equipamentos.

1.2. Apresentação do texto

Esta monografia está dividida da seguinte forma: o Capítulo 2 aborda a fundamentação teórica, apresentando todo o *background* necessário para o entendimento deste trabalho; o Capítulo 4 apresenta o experimento de ICC realizado e seus resultados; e, por fim, o Capítulo 5 apresenta as conclusões e trabalhos futuros.

¹ openbci.com/

² olimex.com/Products/EEG/OpenEEG/EEG-SMT/open-source-hardware

³ openvibe.inria.fr

Fundamentação Teórica

Com as inovações tecnológicas, a possibilidade de controlar dispositivos computadorizados apenas com o cérebro é uma opção cada vez mais emergente, tornando Interfaces Cérebro-Computador (ICC) uma alternativa interessante para as interfaces atuais. Contudo, ainda é necessário diversos estudos na área para tornar uma ICC realmente utilizável. Este capítulo apresenta a fundamentação teórica necessária para o entendimento deste trabalho: (1) uma breve explicação sobre o que é uma ICC; (2) aquisição de sinais; (3) paradigmas baseados em interface cérebro-computador; (4) pré-processamento; (5) extração características; (6) classificação dos sinais; (7) aplicação da ICC; e, por fim (8) o estado da arte.

2.1. Interface Cérebro-Computador

Interface Cérebro-Computador (ICC) é a comunicação entre o cérebro humano e computadores que não dependem dos meios tradicionais tal como a interação motora de um indivíduo (Wolpaw *et al.*, 2000). Uma ICC infere os pulsos elétricos produzidos pelo cérebro e os traduzem em ações e comandos para um computador (Ramadan *et al.*, 2015). As pesquisas realizadas nesta área possuem o intuito de prover diferentes meios de interface com dispositivos eletrônicos, oferecendo diversas possibilidades de aplicações, tais como:

- Controle de uma cadeira de rodas elétrica (Schuh *et al.*, 2014);
- Auxílio na reabilitação de pessoas (Munoz *et al.*, 2014);
- Interface com jogos digitais (Lalor *et al.*, 2005; Legény *et al.*, 2013).
- Sistema de controle robótico remoto (Bhattacharyya *et al.*, 2012)

De acordo com Zander *et al.* (2010) uma ICC pode ser dividida em três categorias:

- **Ativa:** é aquela que deriva suas respostas de atividades cerebrais que são diretamente controladas pelo usuário, independente de eventos externos;
- **Reativa:** é aquela que deriva suas respostas de atividades cerebrais decorrente de reações a estímulos externos, na qual são indiretamente modulados pelo usuário para controlar uma aplicação. Por exemplo, estímulos visuais, táteis ou auditivos;
- **Passiva:** é aquela que deriva suas respostas por meio de atividades cerebrais arbitrárias, sem o controle voluntário do usuário, para enriquecer a interação humano-computador com informações implícitas. Por exemplo, níveis de concentração, engajamento, humor, etc.

Como ilustrado na Figura 1.1, uma ICC pode ser dividida em cinco etapas: (1) aquisição de sinais, (2) pré-processamento, (3) extração de características, (4) classificação e (5) aplicação. Cada uma dessas etapas é responsável por realizar algum processamento e enviar os resultados a próxima etapa. As seis próximas seções descreverão em detalhes cada uma dessas etapas juntamente com os paradigmas baseados em ICC.

2.2. Aquisição do Sinal

Existem duas formas para realizar a leitura das atividades cerebrais. A primeira forma é a invasiva, no qual utiliza-se eletrodos intracranianos para aquisição dos dados, o que permite um sinal com menos ruído e um alto nível de controle sobre os dispositivos (Faria, 2014). Ainda é possível dividir a forma invasiva em duas categorias (Lebedev; Nicolelis, 2006), os *Single Units*, na qual cada eletrodo faz a leitura de um grupo de neurônios em específico, e os *Multi-Units*, na qual cada eletrodo faz a leitura de uma área que abrange vários neurônios simultaneamente, aproveitado o fato de que o cérebro realiza o processamento distribuído da informação em vários neurônios. O segundo tipo é o não-invasivo e também o mais comum. Nesse tipo os eletrodos são colocados sobre o couro cabeludo, e os sinais mensurados são mais fracos devido a resistência natural do crânio, no entanto, o custo é menor e os riscos para o usuário são menores, pois não há a necessidade de uma cirurgia para posicionar os eletrodos (Marshall *et al.*, 2013; Lebedev; Nicolelis, 2006).

As formas não-invasivas de aquisição dos sinais cerebrais podem ser realizadas por meio de diversas técnicas, são elas:

- **Magnetoencephalography (MEG):** é uma técnica que registra a atividade magnética do cérebro por meio da indução magnética. MEG mede o campo magnético gerado pelas correntes intracelulares que fluem através dos dendritos (Nicolas-alonso; Gomez-gil, 2012).
- **Functional Near-Infrared Spectrography (fNIRS):** é uma técnica de espectroscopia óptica que utiliza luz infravermelha para caracterizar flutuações no metabolismo cerebral durante a atividade neuronal (Nicolas-alonso; Gomez-gil, 2012).
- **Functional Magnetic Resonance Imaging (fMRI):** é uma técnica que detecta alterações no volume e fluxo sanguíneo cerebral e os níveis de oxigenação durante a ativação neuronal por meio de campos eletromagnéticos (Waldert *et al.*, 2009).
- **Electroencephalography (EEG):** utiliza eletrodos dispostos no couro cabeludo, para medir a atividade elétrica do cérebro provocada pelas sinapses dos neurônios (Baillet *et al.*, 2001). A qualidade do sinal é baixa devido a resistência gerada pelo couro cabeludo, crânio e outras camadas entre os neurônios e o eletrodo. Além de captar ruídos interno e externos ao cérebro.

Os eletrodos do EEG tem posições padronizadas pela federação internacional de eletroencefalografia e neurofisiologia clínica (Klem *et al.*, 1999). Essa padronização foi realizada para possibilitar a repetição de estudos ao longo do tempo. Esse sistema é conhecido como sistema internacional 10-20 e designa por meio de letras e números, o posicionamento dos eletrodos, sendo que os números ímpares designam o hemisfério esquerdo e os pares o hemisfério direito, as letras designam áreas anatômicas, sendo elas: frontal (F), central (C), parietal (P) e occipital (O). A Figura 2.1 ilustra a posição dos eletrodos seguindo o sistema 10-20.

2.3. Paradigmas Baseados em Interface Cérebro-Computador

Medir atividades cerebrais eficientemente é uma tarefa difícil em uma ICC, uma vez que não é possível inferir pensamentos, mas sim detectar e classificar padrões específicos nos sinais cerebrais associados a uma tarefa ou evento específico (Graumann *et al.*, 2010). Tais tarefas ou eventos são predefinidos em paradigmas (ou estratégias) mentais. Os paradigmas determinam o que o usuário necessita fazer para produzir sinais cerebrais com padrões específicos para que uma ICC consiga interpretá-los corretamente. Nas próximas subseções serão explicados os paradigmas de Atenção Seletiva e de *Motor Imagery*.

2.3.1. Atenção Seletiva

Paradigmas baseados em atenção seletiva requerem estímulos externos que resultam em certos padrões de resposta pelo cérebro (Graumann *et al.*, 2010). Esses estímulos podem ser visuais, auditivos ou táteis, sendo que o visual é o mais utilizado nas pesquisas em ICC que utilizam esse modelo. Existem diversos paradigmas que utilizam estímulos visuais, alguns exemplos são o *Steady State Visually Evoked Potentials* (SSVEP) e o P300 que serão apresentados a seguir.

O paradigma SSVEP consiste em enviar estímulos visuais modulados em frequências específicas para o usuário, podendo ser luzes piscando por exemplo. O cérebro então responde a esses estímulos com a região do córtex visual (próximo a região Oz, Figura 2.1), produzindo sinais na mesma frequência ou em múltiplos dessa frequência (Soman *et al.*, 2015).

Essa técnica é menos susceptível a artefatos (dados não desejados pelo manipulador), tornando-se relativamente fácil a detecção desses estímulos utilizando EEG e ainda possui a vantagem de não necessitar de grandes sessões de treinamento, que é o tempo gasto para o usuário aprender a manipular a ICC.

O SSVEP possui algumas desvantagens, uma delas é que para o sistema conseguir detectar a intenção do usuário, este necessita se concentrar em um símbolo ou região da tela que esteja piscando em uma frequência configurada pela aplicação. Isso ao longo do tempo pode tornar a utilização desconfortável e irritante (Marshall *et al.*, 2013), o que torna essa técnica inviável em sistemas no qual o usuário utilizará por muito tempo.

Já o paradigma P300 consiste em uma série de picos positivos no EEG (uma variação radical na amplitude do sinal durante um pequeno espaço de tempo) que ocorrem aproximadamente 300 milissegundos após o aparecimento do estímulo alvo (Fazel-rezai; Ahmad, 2011), ou seja o estímulo alvo consiste em uma amplitude infrequente que se encontra entre várias frequentes. Estes estímulos podem ser visuais, auditivos ou sensoriais.

Um exemplo de utilização do paradigma P300 é o soletrador (*speller*), caracterizado como uma matriz de caracteres que são mapeados para comandos no computador ou em dispositivos específicos. Os estímulos são os piscares sucessivos de tais caracteres que estimulam o cérebro a produzir padrões denominados P300. Esse padrão ocorre na região do lobo parietal (próximo a região Pz, Figura 2.1) (Graumann *et al.*, 2010). Em uma aplicação de ICC é possível detectar e determinar o símbolo que o usuário está focado e então executar o comando correspondente.

O P300 tem sido um dos principais paradigmas de pesquisas em modelos de ICC reativas, pois ele possui diversas características atraentes, por exemplo, a eficácia na utilização pela maioria dos usuários, por ser um método relativamente rápido e que requer quase nenhuma sessão de treinamento com o usuário (Guger *et al.*, 2009).

O experimento realizado nesta monografia utiliza o paradigma P300, não só pelas vantagens já apresentadas, mas também por ser o paradigma mais simples dos apresentados nessa fundamentação teórica, além de já estar disponível aplicações que utilizam esse paradigma, por exemplo o software OpenViBE, que

implementa como exemplo três aplicações baseadas no paradigma P300.

2.3.2. *Motor Imagery*

Motor imagery pode ser vista como um ensaio mental de um movimento do corpo, sem que o movimento realmente aconteça (Pfurtscheller; Neuper, 2001). Tais movimentos resultam em uma ativação do córtex motor que, por sua vez, promove oscilações que podem ser detectadas por meio do EEG. Essas oscilações são manipuladas pela aplicação ICC e são transformadas em comandos para a aplicação específica. Essa técnica geralmente exige muitas sessões de treinamento antes de conseguir resultados aceitáveis (Marshall *et al.*, 2013), o que pode se tornar um problema dependendo da aplicação desejada.

Motor Imagery depende muito da concentração do usuário, sendo afetada por distrações ou frustrações, além de que muitas pessoas têm dificuldades em se acostumar com a sensação de imaginar movimentos motores, uma vez que dificilmente elas sabem imaginar uma movimentação de uma parte do corpo, principalmente quando não as possui. O que acabam fazendo é imaginar imagens das partes do corpo se movendo (Neuper *et al.*, 2005).

2.4. Pré-Processamento

Nem todos os dados mensurados em uma ICC são úteis para a construção de uma aplicação. O pré-processamento é realizado para atenuar eventuais ruídos e artefatos oriundos da etapa de aquisição. Ainda nessa etapa é realizado as primeiras operações necessárias para destacar ou ocultar possíveis elementos que caracterizam um determinado padrão nos sinais (Lotte, 2014). Alguns filtros que podem ser utilizados nessa etapa são:

- **Filtro passa baixa:** permite a passagem apenas de baixas frequências, atenuando apenas as que estão acima de uma frequência de corte indicada. É utilizado para remover as frequências que não são relevantes para os extratores de características, explicado na próxima seção, por exemplo em ICC, as frequências úteis geralmente vão até 50 Hz.
- **Filtro rejeita faixa:** permite a passagem da maioria das frequências, porém atenua uma faixa determinada pelo filtro. Utilizado por exemplo na remoção de ruídos e frequências que não são relevantes para os extratores de características utilizados posteriormente;
- **Filtro passa faixa:** permite a passagem de frequências de uma certa faixa e atenua as frequências fora dessa faixa. No experimento realizado neste trabalho é utilizado o filtro passa faixa configurado para as frequências entre 1 e 20 Hz, devido os padrões procurados nos sinais estarem dentro dessa faixa de frequência.

Os ruídos e artefatos podem ter diversas origens, por exemplo, no momento da aquisição por meio de um EEG, o cérebro está realizando diversas atividades, portanto os sinais captados podem conter informações de diversas atividades além das de interesse. Outras possíveis origens de artefatos são os ruídos externos provocados por mal funcionamento no hardware ou interferências externas, como é o caso da rede elétrica ou outros equipamentos eletroeletrônicos.

2.5. Extração de Características

Esta etapa é realizada para extrair características relevantes na construção de uma aplicação de ICC. Tais características serão utilizadas, posteriormente, pelos classificadores nas etapas de treinamento e teste. Uma característica é uma particularidade ou atributo que é distinguível em um objeto, portanto, as características extraídas devem representar adequadamente as informações ocultas no sinal cru do EEG (Bashashati *et al.*, 2015). Existem diversos tipos de características que podem ser utilizadas para destacar padrões nos sinais.

Lotte *et al.* (2007) realizou um levantamento de algumas características possíveis que podem ser utilizadas, são elas: *Amplitude Values of EEG Signals*, *Band Powers*, *Power Spectral Density*, *AutoRegressive*, *Adaptive AutoRegressive*, *Time-Frequency* e *Inverse Model-Based*.

A saída desta etapa é um vetor de características, que então é utilizado como entrada para a próxima etapa, que é a de classificação, no qual a partir das características oriundas dessa etapa é realizada a classificação das intenções de um indivíduo. No experimento realizado nesta monografia a característica utilizada foi *Amplitude Values of EEG Signals*, em que após o sinal mensurado passar por filtros é gerado um vetor de características contendo tais valores. O tamanho desse vetor depende da taxa de amostragem e também do pré-processamento realizado no sinal. A próxima seção apresenta em maiores detalhes a etapa de classificação.

2.6. Classificação

A etapa de classificação é que faz o mapeamento de uma amostra em ações ou comandos, ou seja, a cada nova amostra, o classificador decide qual é a ação ou comando deverá ser executado. Neste trabalho as amostras são o resultado do processamento de um intervalo de tempo do sinal pela etapa de extração de características, sendo representadas por um vetor de características. O classificador utiliza as informações contidas nesse vetor para realizar a classificação. Michie *et al.* (1994) define um classificador como um modelo formal capaz de tomar decisões repetidamente para diferentes situações. No contexto de ICC os classificadores utilizam as informações obtidas dos sinais neurais para reconhecer e executar comandos do usuário em computadores e dispositivos eletrônicos.

Os classificadores que seguem a abordagem de aprendizagem supervisionada utilizam amostras de exemplos rotuladas, denominadas conjunto de treinamento (Kotsiantis *et al.*, 2006). Este conjunto é formado por diversas amostras rotuladas de cada classe. Neste trabalho o conjunto de treinamento é dividido em duas classes, a primeira é a classe alvo, que é composta pelas amostras que contém a componente P300 e a segunda classe é a não-alvo que é composta pelas amostras que não contém a componente P300. Após o treinamento do classificador ele será capaz de reconhecer novas amostras e classificá-las em alguma das classes que compõem o conjunto de treinamento.

Existem diversos algoritmos de classificação, dentre eles foram selecionados dois para serem avaliados neste trabalho. São eles: *Support Vector Machine* (SVM) e *Linear Discriminant Analysis* (LDA). As próximas subseções descreverão alguns algoritmos que seguem a abordagem de aprendizagem supervisionada, juntamente com os dois algoritmos que foram utilizados nos experimentos.

2.6.1. *Support Vector Machine* (SVM)

Support Vector Machine é uma técnica de classificação de padrões proposto por Vapnik (1995). Essa técnica constrói a partir de um conjunto de exemplos de treino contendo classes categorizadas, um modelo capaz de atribuir a novos exemplos uma das categorias contidas no conjunto de treinamento. Para isso o SVM utiliza funções lineares para distinção das classes, selecionando hiperplanos que maximizam as margens (Lotte *et al.*, 2007). A Figura 2.2 ilustra o funcionamento do SVM, no qual a partir de diversos possíveis hiperplanos não otimizados, o SVM escolhe o hiperplano otimizado

Raramente temos problemas que podem ser linearmente separáveis, portanto o SVM mapeia implicitamente os dados de entrada para um espaço de alta dimensionalidade, pois nessa dimensão é possível separar linearmente os conjuntos, sendo que a função responsável por esse mapeamento é denominado *kernel*.

Originalmente o SVM foi idealizado para classificações binárias, não suportando problemas

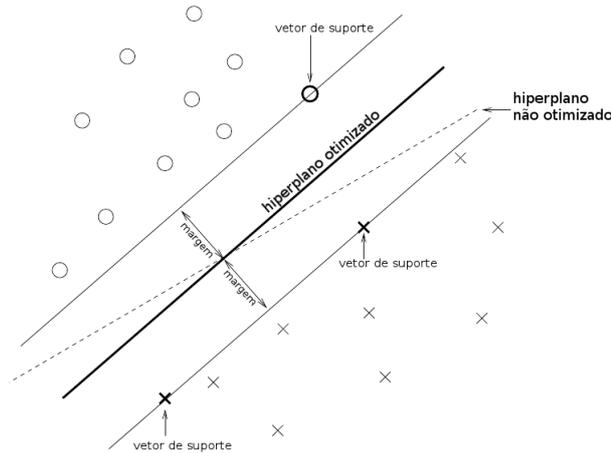


Figura 2.2. Ilustração do funcionamento do SVM. Adaptado de: Marshall *et al.* (2013)

multi-classe, porém existem duas abordagens que estendem suas funcionalidades dando suporte a esses problemas (Weston; Watkins, 1998). A primeira abordagem é a *one-versus-rest*, que consiste em construir M classificadores para resolver o problema de M classes, ou seja, cria-se um classificador que rotula a classe M como positiva e as demais como negativa. Enquanto a segunda abordagem é o *one-versus-one*, que cria um classificador binário entre todas as classes, resultando em $M \times M$ classificadores. A abordagem utilizada neste trabalho é a *one-versus-one*, pois é a abordagem disponível no software `OpenViBE`.

2.6.2. *Multilayer Perceptron* (MLP)

Multilayer Perceptron (MLP) é uma rede de unidades de processamento simples, dispostos em um modelo hierárquico de camadas (Gallagher, 2000). Essa rede é composta de várias camadas de neurônios, sendo uma camada de entrada, uma ou várias camadas intermediárias (ocultas) e uma camada de saída. Cada entrada de neurônio está ligado a saída da camada anterior, exceto a camada de entrada, que é alimentada com um vetor de características (Lotte *et al.*, 2007). As ligações possuem um valor de peso numérico associado a elas e o sinal transmitido, por meio de uma ligação, é multiplicado pelo valor do peso. Cada unidade calcula alguma função da soma de suas entradas ponderadas, e transmite o resultado por meio de suas conexões de saída (Gallagher, 2000). A Figura 2.3 ilustra a arquitetura da rede de neurônios de um MLP.

Seja qual for a estrutura da rede, o fluxo de informação é realizado em uma única direção, começando na da entrada da rede e terminando ao final dela, sem ciclos ou conexões para nós anteriores (Gallagher, 2000). Redes neurais, assim como MLP, tem a capacidade de aproximar funções contínuas, adicionando o fato de que podem classificar qualquer número de classes. Isto faz de redes neurais classificadores muito flexíveis que podem se adaptar a uma grande variedade de problemas (Lotte *et al.*, 2007).

Para a realização do treinamento da rede, é utilizado o algoritmo *backpropagation* (Rumelhart *et al.*, 1986). Esse algoritmo fornece um meio de ajustar os pesos na rede, dado um conjunto de dados de treinamento (Gallagher, 2000). O algoritmo funciona em duas etapas, na primeira etapa é apresentado um padrão à camada de entrada da rede e os resultados são propagados por meio da rede, camada por camada, até que a resposta seja produzida pela camada de saída. Na segunda etapa, a saída obtida é comparada com a saída desejada, se não estiver correto, o erro é calculado e propagado a partir da camada de saída até a camada de entrada. Os pesos das conexões das unidades das camadas internas vão sendo modificados conforme o erro é retropropagado. Essas etapas são repetidas até que o erro atinja um nível aceitável.

Uma desvantagem do classificador MLP é que, por ser um aproximador universal, ele é sensível a *overtraining* (Lotte *et al.*, 2007), ou seja, ele pode ficar muito especialista em classificar certa classe, tornando

² cse22-iiith.vlabs.ac.in/exp4/images/structure.png

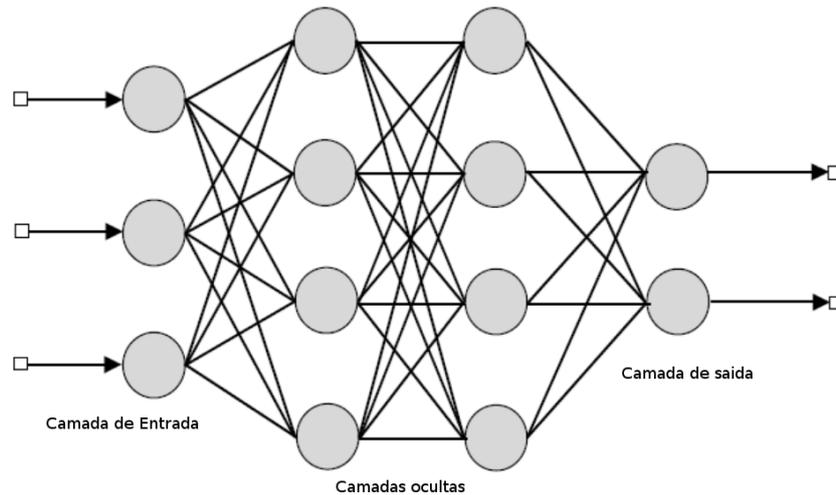


Figura 2.3. Ilustra a arquitetura de um MLP. Adaptado de: Virtual Labs²

a generalização ruim. Isso deve ser observado no momento do treinamento, evitando, por exemplo, treinar excessivamente com uma única classe, oferecendo poucos exemplos das demais classes.

2.6.3. *Linear Discriminant Analysis (LDA)*

Linear Discriminant Analysis (LDA) é uma técnica de classificação proposto por Fisher (1936). Essa técnica utiliza hiperplanos para separar duas classes, a separação por hiperplano é obtida buscando projeções que maximizam a distância entre as classes (Fukunaga, 1990).

O algoritmo LDA tem uma baixa exigência computacional, o que o torna adequado para sistemas em que o tempo para a classificação é importante (Lotte *et al.*, 2007), além disso, esse classificador é simples de usar e geralmente proporciona bons resultados (Bostanov, 2004; Hoffmann *et al.*, 2008)

2.7. Aplicação de uma ICC

A etapa de aplicação de uma ICC consiste na interface entre o classificador e aplicação real, por exemplo, para controlar os movimentos de uma cadeira de rodas elétrica, controlar os movimentos e cliques do cursor do mouse em um computador, controlar os movimentos do braço de um robô, etc. É nessa etapa que é executada a intenção do usuário por meio de um comando ou ação em um computador ou dispositivo eletrônico.

Cada aplicação tem um conjunto finito de ações que são executáveis, sendo que esse conjunto representa as possíveis respostas do classificador. Ao final dessa etapa a intenção do usuário será atendida e o ciclo será reiniciado com o objetivo de detectar a próxima intenção do usuário.

Neste trabalho foi utilizado a aplicação P300 Speller para realizar a avaliação nos classificadores. O P300 Speller é um soletrador que utiliza o paradigma P300 e é utilizado para soletrar palavras. Nesta aplicação é utilizada uma matriz 6x6 contendo caracteres alfanuméricos, no qual o usuário se concentra no caractere que deseja soletrar. Toda vez que a ICC reconhece uma letra ela é selecionada e o ciclo é reiniciado até que a palavra desejada esteja completa.

Na próxima seção serão apresentados alguns trabalhos relacionados atuais que compõe a fundamentação, além de demonstrar como são aplicadas as técnicas para construção de uma aplicação de ICC.

O Estado da Arte

Pesquisas em ICC vem crescendo bastante nas últimas décadas. A maior parte destas pesquisas estão relacionadas a aplicações voltadas para pessoas que possuem alguma deficiência motora (Schuh *et al.*, 2014), no qual uma ICC pode ser utilizada para restaurar ou ampliar funções cognitivas e motoras de indivíduos normais ou que possuem alguma deficiência. Por outro lado, aplicações com outro propósito também estão ganhando foco, por exemplo, promovendo a interação com dispositivos eletrônicos (Perakakis; Potamianos, 2013) ou aplicações utilizando jogos virtuais (Marshall *et al.*, 2013). Essa seção descreverá algumas das mais recentes aplicações encontradas na literatura, com o objetivo de demonstrar o que é possível realizar e as tecnologias envolvidas com estas aplicações.

Resumidamente, problemas que podem ser resolvidos utilizando uma ICC são no geral: (1) reabilitação de pessoas que perderam totalmente ou parcialmente os movimentos (Daly; Wolpaw, 2008), (2) aplicações que ajudam pessoas com alguma deficiência a interagirem com dispositivos eletrônicos (Halder *et al.*, 2015), (3) aplicações que detectam a atenção do usuário para serem utilizadas como informação adicional em outras aplicações ou monitoramento (Perakakis; Potamianos, 2013) e (4) aplicações que promovem novas formas de interação em jogos digitais (Finke *et al.*, 2009).

Em Mirghasemi *et al.* (2006) foi realizada a comparação de cinco algoritmos de classificação, sendo eles: *Linear Support Vector Machine* (LSVM), *Gaussian Support Vector Machine* (RSVM), *Multilayer Perceptron* (MLP), *Fisher Linear Discriminant* (FLD) e *Kernel Fisher Discriminant* (KFD). Os testes realizados utilizaram uma base de dados da competição BCI 2003 gravados em uma aplicação P300 *Speller*. Essa base contém três sessões com vários caracteres, sendo as duas primeiras utilizadas para treino e a terceira para teste. Os dados foram gravados utilizando sessenta e quatro canais, porém foram utilizados apenas três para o primeiro teste e dez para o segundo. Para o MLP foram realizados testes utilizando duas (MLP2) e três (MLP3) camadas ocultas. Os resultados obtidos para três canais foram: FLD - 97%, KFD - 97%, LSVM - 94%, RSVM - 94%, MLP2 - 94%, MLP3 - 87%. Enquanto que para dez canais os resultados foram: FLD - 100%, KFD - 100%, LSVM - 100%, RSVM - 94%, MLP2 - 97%, MLP3 - 90%.

No trabalho realizado em Halder *et al.* (2015), foi desenvolvido duas aplicações controláveis através de ICC, um navegador *web* e um reprodutor multimídia. Ambas as aplicações são controladas através de matrizes de símbolos, utilizando o paradigma P300. O navegador sinaliza as possíveis ações do usuário com símbolos, por exemplo, em uma página *web* todo *hyperlink* disponível é associado a uma letra do alfabeto. A matriz de símbolos é então gerada dinamicamente contendo os símbolos necessários para interagir com a página em foco. O reprodutor multimídia possui ações pré-definidas, como aumentar volume, diminuir volume,

avancar, retroceder, pausar, iniciar, parar, etc. Cada uma dessas ações possui um símbolo correspondente na matriz.

A avaliação de ambas as aplicações foram realizadas a partir de testes com dez participantes saudáveis e três com deficiências motoras. A acurácia, dentre os indivíduos saudáveis, do navegador *web* foi de 90% e do reprodutor de música 85%. Enquanto para os três indivíduos com deficiências motoras a acurácia foi de 62% para o navegador e 58% para o reprodutor multimídia. O classificador utilizado foi o *Stepwise Linear Discriminant Analysis (SWLDA)*.

Resalat e Saba (2015) apresentam um detector de sonolência para motoristas. O objetivo é determinar a atenção do motorista, detectando a sonolência através de uma ICC. Para a construção de tal sistema foi utilizado o paradigma SSVEP, no qual a partir de quatro LEDS dispostos a alguns centímetros do usuário, dois diretamente a frente e dois em um ângulo de 45° a direita e a esquerda, conseguem detectar a atenção do usuário. Tais experimentos utilizaram dois tipos de característica, ambas retiradas do domínio da frequência. Os classificadores utilizados foram LDA, SVM e Max. Os testes foram realizados com cinco indivíduos, sendo que no melhor caso foi obtido uma acurácia de 98,2%.

Em Finke *et al.* (2009) foi desenvolvido um jogo, denominado *MindGame*, baseado no paradigma P300. Nesse jogo o usuário controla um personagem em um ambiente tridimensional que é composto por um tabuleiro de damas com 28x18 campos e doze árvores randomicamente posicionadas no tabuleiro. Os campos com árvores são os que transmitem o estímulo para o usuário. A tarefa do jogador é mover o personagem de árvore em árvore até todas as árvores serem visitadas. Os experimentos foram realizados em 11 pessoas, 9 homens e 2 mulheres, com idades de 24 a 39 anos. Foram utilizados dez eletrodos nas posições Fz, Cz, Pz, Oz, C3, C4, P3, P4, PO7 e PO8, seguindo o sistema internacional 10-20. Para classificação foram utilizados dois algoritmos, o *Fisher's Linear Discriminant Analysis (FLDA)* e o *Quadratic Discriminant Analysis (QDA)*, em que a acurácia dos algoritmos foram de aproximadamente 84% para o FLDA e 73% para o QDA.

Manyakov *et al.* (2011) comparou algoritmos de classificação para o paradigma P300 utilizando a aplicação P300 Speller. O experimento foi realizado em um grupo de pessoas que possuem alguma deficiência, como síndrome da artéria cerebral média (MCA), esclerose lateral amiotrófica (ELA), Hemorragia subaracnóidea (SAH) ou sofrendo de alguma deficiência motora ou da fala. Foram analisados sete tipos de classificadores, lineares e não-lineares, e foi investigado a acurácia para cada paciente considerando sua doença. Os experimentos foram realizados com 16 pessoas, e os algoritmos comparados foram: *Fisher's Linear Discriminant Analysis (LDA)*, *Stepwise Linear Discriminant Analysis (SWLDA)*, *Bayesian Linear Discriminant Analysis (BLDA)*, *Linear Support Vector Machine (SVM)*, *Nonlinear Support Vector Machine (nSVM)*, *Artificial Neural Network (NN)* e *Method Based on Feature Extraction (FE)*. Na comparação feita, os algoritmos lineares obtiveram uma performance melhor ou igual aos algoritmos não-lineares, sendo que o BLDA obteve os melhores resultados, seguido pelo SVM e nSVM em segundo, SWLDA e LDA em terceiro e os piores resultados foram para o FE e o NN.

Bhattacharyya *et al.* (2012) desenvolveu um sistema de controle para um robô utilizando ICC. Esse sistema é capaz de detectar dezesseis movimentos diferentes, por exemplo ir para frente, virar à direita, virar à esquerda, etc. Para a implementação foi utilizado o paradigma *motor imagery*, sendo assim, cada movimento do robô foi mapeado em uma sequência específica de movimentos imaginados pelo usuário. Os membros do corpo que foram mapeados são os dedos, ombros e cotovelos. A classificação é realizada em dois estágios, o primeiro classifica qual é o lado do membro imaginado (direito ou esquerdo), no segundo estágio é classificado qual é o membro da intenção (dedo, cotovelo ou ombro).

Foram realizados experimentos utilizando dois tipos de características, *Hjorth* e *Adaptive Autoregressive (AAR)*. E para a classificação o algoritmo KNN. As taxas de acerto foram superiores a 80% em todos os experimentos, tanto para detectar qual lado foi a intenção (direito ou esquerdo) quanto para detectar para qual membro foi a intenção (dedo, cotovelo ou ombro).

O próximo capítulo apresenta as experimentações realizadas com os resultados obtidos.

Experimentações

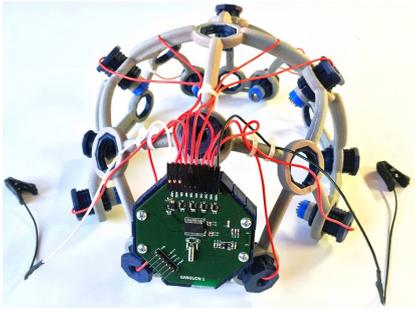
Até o momento, foi levantado a Fundamentação Teórica sobre o desenvolvimento de aplicações utilizando Interfaces Cérebro-Computador (ICC), além dos trabalhos mais recentes da área que utilizam os classificadores que foram aplicados neste trabalho. Neste capítulo, será apresentado como foram realizados os experimentos que norteiam os objetivos deste estudo.

Contextualizando os objetivos deste trabalho de conclusão de curso, foi utilizada uma aplicação ICC baseado em estímulos visuais externos (reativa). O experimento consiste em projetar sinais visuais para um indivíduo para que ele tome decisões. A tomada de decisão será obtida por meio da ICC e é onde entra o objetivo principal do trabalho: caracterizar o sinal obtido, aplicar alguns classificadores e comparar os resultados de cada um deles, com a intenção de determinar qual deles obtém a melhor acurácia em determinadas condições. O experimento contribui principalmente para a escolha do melhor algoritmo de classificação para aplicações ICC que utilizam este tipo de estímulo visual.

Para a aquisição dos sinais foram utilizados dois hardwares: O primeiro é o **OpenBCI** que é uma interface neural *open-source* versátil e acessível, que pode ser usado para medir a atividade elétrica cerebral (EEG), atividade muscular (eletromiografia - EMG) e frequência cardíaca (eletrocardiograma - ECG). O **OpenBCI** possui oito canais mais um de referência e um terra, porém pode ser expandido para dezesseis canais utilizando uma placa expansora chamada **daisy module**. Os eletrodos usualmente utilizados são os passivos, que não possuem filtro de ruído no eletrodo, podendo ser de prata-cloreto de prata, que devido as propriedades do material descarta a necessidade de utilização da pasta eletrolítica, ou eletrodos revestidos de ouro, que necessitam da pasta eletrolítica. O **Olimex EEG-SMT** também é *open-source*, ele é baseado no projeto **OpenEEG**¹. O **Olimex EEG-SMT** foi desenvolvido para a utilização de eletrodos passivos, porém é possível a utilização com eletrodos ativos, que possuem filtros à prova de ruídos externos diretamente implantados no eletrodo, o que pode melhorar sua performance. Ambos os eletrodos utilizados no **Olimex EEG-SMT** necessitam da utilização da pasta eletrolítica. As Figuras 4.1a e 4.1b ilustram o **OpenBCI** e o **Olimex EEG-SMT**, respectivamente.

Além desses dois hardwares apresentados, existem no mercado diversos outros que são capazes de realizar a medição de atividade elétrica cerebral. A Tabela 4.1 mostra alguns dos hardwares disponíveis no mercado, destacando a quantidade de canais e o valor aproximado de cada um. As versões do **OpenBCI** e também o **Olimex EEG-SMT** é possível movimentar os eletrodos, ou seja, podem ser posicionados em qualquer lugar da cabeça do usuário, já o **EMOTIV Epoc+**, **EMOTIV Insight** e o **Muse** possuem os eletrodos posicionados

¹ <http://openeeg.sourceforge.net/>



(a) Equipamento com o capacete Ultra Cortex Mark III e o Open BCI board adaptado.



(b) Equipamento Olimex EEG-SMT com eletrodos (passivo/ativos) plugados.

Figura 4.1. Equipamentos de eletrencefalografia

em regiões pré-definidas, impossibilitando sua movimentação.

Hardware	Canais+Referência	Preço
EMOTIV EPOC+ ²	14+2	USD 799,00
EMOTIV Insight 5 ³	5+2	USD 299,00
OpenBCI R&D Board ⁴	16 + 1 + 1 bias	USD 899,00
OpenBCI Board ⁵	8 + 1 + 1 bias	USD 499,00
Olimex EEG-SMT ⁶	2+1	EUR 99,00
Muse ⁷	4+1	USD 249,00

Tabela 4.1. Comparativo entre alguns hardwares disponíveis. A diferença entre os as duas versões do OpenBCI é que o OpenBCI R&D Kit vem com a placa expansora para dezesseis canais.

Para o desenvolvimento da aplicação usada nos experimentos, foi utilizado o software OpenViBE que é uma plataforma gratuita e *open-source* para o desenvolvimento, teste e utilização de ICC (Renard *et al.*, 2010). O OpenViBE tem diversos exemplos de experimentos utilizando ICC. Dentre esses experimentos foram selecionados e adaptados dois para a realização deste trabalho. Os escolhidos foram:

- P300 Speller: é um experimento de um soletrador que utiliza o paradigma P300.
- P300 Speller xDAWN: utiliza cenários bem parecidos com o P300 Speller, porém utiliza filtros espaciais para melhorar a detecção da componente P300 quando são utilizados mais de três canais na experimentação.

As seções seguintes descreverão em detalhes ambos os experimentos, apresentando sua construção e configuração.

4.1. Caso 1: Desenvolvimento do cenário P300 Speller

A aplicação P300 Speller utiliza o paradigma P300, descrito na seção 2.3.1, que é um pico positivo no EEG que ocorre 300 milissegundos após um evento esperado. Tal aplicação pisca randomicamente colunas e linhas em uma matriz 6x6 de caracteres alfanuméricos (veja Figura 4.2), e o usuário é instruído a focar na letra que ele deseja soletrar. Após uma quantidade predeterminada de repetições o sistema seleciona a letra desejada.

O P300 Speller é composto por três cenários no OpenViBE, o de aquisição, o de treinamento e o de teste. As próximas subseções descreverão esses cenários.

² emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/

³ emotiv.com/product/emotiv-insight-5-channel-mobile-eeg/

⁴ shop.openbci.com/collections/frontpage/products/openbci-16-channel-r-d-kit

⁵ shop.openbci.com/collections/frontpage/products/openbci-32-bit-board-kit

⁶ olimex.com/Products/EEG/OpenEEG/EEG-SMT/open-source-hardware

⁷ choosemuse.com/



Figura 4.2. Matriz de caracteres do P300 Speller

4.1.1. Descrição do Cenário de Aquisição de Sinais

O primeiro cenário é utilizado para a aquisição dos sinais que compõem o conjunto de treinamento utilizado pelos classificadores. Tal cenário gera um arquivo contendo os sinais neurais do indivíduo acompanhado com anotações sobre o estímulo corrente naquele momento do tempo. As anotações dizem qual estímulo foi enviado para o usuário naquele instante de tempo, isso é necessário para descobrir o exato momento do estímulo para que seja possível detectar a componente P300.

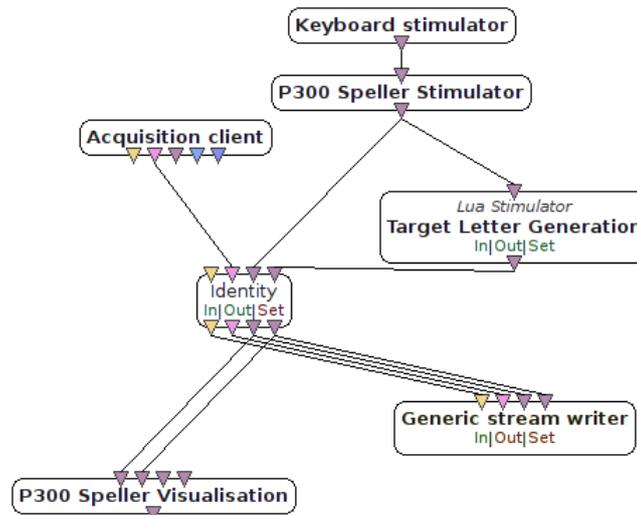


Figura 4.3. Cenário de aquisição de sinais para o treinamento dos classificadores

A Figura 4.3 ilustra o cenário descrito no `OpenViBE` para aquisição de sinais para o treinamento dos classificadores, no qual é possível visualizar as caixas utilizadas para a construção desse cenário. Cada uma das caixas são utilizadas para:

- **Acquisition Client:** é utilizado para se conectar com o `Acquisition Server`. É desta caixa que o `OpenViBE` recebe o sinal recebido pelo hardware. O `Acquisition Server` é uma ferramenta desenhada para se comunicar com vários dispositivos de aquisição de sinal. A tarefa do servidor é de transmitir os sinais adquiridos e outras informações do experimento para aplicações `OpenViBE` em um formato padronizado.

- **Identity:** apenas replica as entradas nas saídas.
- **Keyboard stimulator:** gera estímulos através do teclado. É utilizado, por exemplo, para configurar um comando para iniciar o experimento.
- **P300 Speller Stimulator:** utilizado para realizar a configuração do gerador de estímulos, por exemplo, a duração do experimento, a duração de cada piscada, o número de repetições para cada letra, etc. Essa caixa tem como saída a sequência de estímulos que será enviado para o `Generic stream writer` para gravar o sinal obtido com o tempo dos estímulos no arquivo e também é utilizado pela caixa `P300 Speller Visualisation` para exibir o soletrador ao usuário.
- **Target Letter Generation:** utilizado para realizar a configuração dos caracteres alvos, ou seja, a sequência de caracteres que serão marcadas como alvo pelo gerador de estímulos.
- **P300 Speller Visualization:** essa caixa é utilizada para mostrar os estímulos gerados pelo `P300 Speller Stimulator` para o usuário. É essa caixa que realiza a criação e o gerenciamento da matriz de caracteres e modula os estímulos para o usuário.
- **Generic stream writer:** essa caixa é utilizada para gravação dos sinais recebidos do `Acquisition client` acompanhado dos estímulos gerados pelo `P300 Speller Stimulator` e pelo caracter alvo gerado pelo `Target Letter Generation`.

Cada uma dessas caixas tem parâmetros que são configuráveis e como esse cenário foi adaptado de um exemplo do `OpenViBE`, todos vieram com seus campos preenchidos com valores pré-estabelecidos. Sendo assim, para o experimento foram mantidos as configurações originais da maioria dessas caixas, sendo alterado apenas as que refletem os caracteres que são marcadas como alvo. Para o cenário de aquisição de sinais foram necessários a alteração das caixas `Target Letter Generator` e `P300 Speller Stimulator`. A seguir serão explicados em detalhes cada parâmetro dessas caixas e quais foram as alterações realizadas para o experimento.

Na caixa `P300 Speller Stimulator`, que é utilizada para configurar o gerador de estímulos, o único parâmetro alterado foi o `Number of trials`, que configura a quantidade de alvos que aparecerão na tela e os valores que esse campo assumiu de acordo com a sequência de caracteres utilizados no `Target Letter Generator`. Os parâmetros utilizados no experimento e seus respectivos valores podem ser vistos na Tabela 4.2 e a definição de cada parâmetro é detalhada da seguinte forma:

- **Number of rows:** utilizado para informar a quantidade de linhas que a matriz de caracteres possui. Deve coincidir com a quantidade de linhas da matriz gerada pelo `P300 Speller Visualization`.
- **Number of columns:** utilizado para informar a quantidade de colunas que a matriz de caracteres possui. Deve coincidir com a quantidade de colunas da matriz gerada pelo `P300 Speller Visualization`.
- **Number of repetitions:** utilizado para configurar a quantidade de vezes que serão repetidos os estímulos para cada alvo.
- **Number of Trials:** utilizado para configurar a quantidade de vezes que serão enviados estímulos ao usuário. Deve coincidir com a sequência de caracteres gerados pelo `Target Letter Generator`.
- **Flash duration:** utilizado para configurar o tempo que cada linha ou coluna da matriz que ficará em destaque.
- **No flash duration:** utilizado para configurar quanto tempo ficará sem nenhum estímulo na tela (entre um estímulo e outro).
- **Inter-repetition delay:** utilizado para configurar o tempo que a matriz não produzirá estímulos entre duas repetições de piscada.
- **Inter-trial delay:** utilizada para configurar quanto tempo a matriz não irá gerar estímulos durante a troca do alvo.

- **Avoid neighbor flashing:** utilizado para configurar se o gerador enviará estímulos para colunas ou linhas imediatamente a suas adjacentes que tenham acabado de receber um estímulo.

Campo	Valor
Start stimulation	OVTk_StimulationId_Label_00
Row stimulation base	OVTk_StimulationId_Label_01
Collumn stimulation base	OVTk_StimulationId_Label_07
Number of rows	6
Number of columns	6
Number of repetitions	12
Number of trials	8
Flash duration (in sec)	0.2
No flash duration (in sec)	0.1
Inter-repetition delay (in sec)	1.0
Inter-trial delay (in sec)	3.0
Avoid neighbor flashing	false

Tabela 4.2. Campos e valores utilizados para a configuração do `P300 Speller Stimulator`

Enquanto para a caixa `Target Letter Stimulator`, que é utilizada para configurar os caracteres que serão marcadas como alvo, o único parâmetro alterado foi o `Word to Spell`, que configura a sequência de caracteres que serão marcados como alvo. O valor que esse campo assumiu nessa etapa foi `COMPUTER`. Os parâmetros utilizados no experimento e seus respectivos valores podem ser vistos na Tabela 4.3 e a definição de cada um deles são detalhados a seguir:

- **Lua Script:** utilizado para configurar o caminho que contém o *script* escrito na linguagem Lua que gera cada letra.
- **Word to Spell:** utilizado para configurar a sequência de caracteres que serão marcados como alvo no `P300 Speller Visualization`. Caso esse campo esteja vazio são gerados caracteres aleatórios.
- **Row Stimulation Base:** é utilizada pelo *script* escrito na linguagem Lua para calcular em qual linha da matriz está a letra que será marcada como alvo.
- **Column Stimulation Base:** é utilizada pelo *script* escrito na linguagem Lua para calcular em qual coluna da matriz está a letra que será marcada como alvo.

Campo	Valor
Lua Script	<code>\${Player_ScenarioDirectory}/p300-speller-target.lua</code>
Word to Spell	<code>computer</code>
Row Stimulation Base	OVTk_StimulationId_Label_01
Collumn Stimulation Base	OVTk_StimulationId_Label_07
Delay Before Sending (in sec)	2

Tabela 4.3. Campos e valores utilizados para a configuração do `Target Letter Generation`

4.1.2. Descrição do Cenário de Treino dos Classificadores

Este cenário é utilizado para realizar a configuração e treinamento dos classificadores. O conjunto de treinamento para os classificadores de aplicações que utilizam o paradigma P300, são compostos por duas classes: a classe alvo, que são amostras de sinais que possuem a componente P300 e a classe não alvo, que são as amostras de sinais que não contém essa componente. A Figura 4.4 ilustra o cenário no software `OpenViBE` que foi utilizado para o treinamento do classificador e mostra quais caixas foram utilizadas para sua construção. A definição das caixas são detalhadas a seguir:

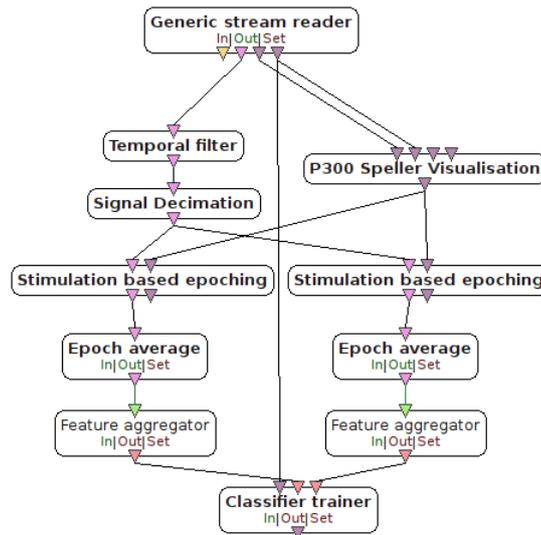


Figura 4.4. Cenário utilizado para o treinamento dos classificadores

- **Generic stream reader:** essa caixa é utilizada para realizar a leitura de arquivos gravados pelo `Generic stream writer`, tendo como saída os sinais neurais gravados do usuário acompanhado dos estímulos e caracteres que foram gravados no cenário de aquisição de sinais.
- **Temporal filter:** essa caixa é utilizada para realizar a filtragem no sinal. Os filtros disponíveis são: passa baixa, passa alta, passa faixa, rejeita faixa.
- **Signal Decimation:** essa caixa reduz a taxa de amostragem do sinal de entrada, tem como parâmetro o `decimation factor` que define o quanto a taxa de amostragem será reduzida.
- **Stimulation based epoching:** o objetivo dessa caixa é selecionar sinais próximos a eventos. O evento é enviado para a caixa e ela seleciona um intervalo configurável de tempo do sinal perto desse evento.
- **Epoch Average:** é utilizado para realizar a média em matrizes de sinal, espectro ou vetores de característica. Nessa caixa é possível configurar os seguintes tipos de média:
 - **Moving Average:** neste caso, o cálculo da média é realizado a cada chegada de uma nova entrada, começando logo que o número suficiente de entradas foram recebidas.
 - **Moving average (Immediate):** neste caso, o cálculo da média é realizado a cada chegada de uma nova entrada, iniciando-se imediatamente. Quando o número de entradas recebidas é menor do que o número desejado, a média é calculada sobre esse número intermediário de entradas.
 - **Epoch block average:** neste caso, a média é feita quando o número suficiente de entradas é alcançado. Uma vez que este número exato de entradas é recebida, a média é calculada e o resultado é enviado à saída.
 - **Cumulative average:** neste caso, o cálculo da média é realizado em um número infinito de entradas a partir da primeira recebida até a última.
- **Feature aggregator:** essa caixa é utilizada para concatenar cada entrada em um vetor de características e o enviar à saída.
- **Classifier Trainer:** essa caixa expõe uma interface genérica para classificadores, podendo selecionar o classificador que deseja utilizar. Para cada classificador selecionado, essa caixa muda sua estrutura para que seja possível gerenciar os possíveis parâmetros dos algoritmos. O `OpenViBE` traz três classificadores já implementados: SVM, MLP e LDA.

Para o experimento, a caixa alterada no cenário de Treinamento dos Classificadores foi a `Classifier Trainer` que realiza a configuração dos classificadores. Cada classificador possui seus próprios parâmetros e para o experimento foram utilizados os padrões fornecidos pelo software `OpenViBE`. Os parâmetros e os

valores correspondentes dos classificadores LDA e SVM podem ser vistos nas Tabelas 4.4 e 4.5, respectivamente.

Campo	Valor
Train trigger	OVTk_StimulationId_Train
Multiclass strategy to apply	OneVsOne
Pairwise Decision Strategy	PKPD
Class 1 label	OVTk_StimulationId_Target
Class 2 label	OVTk_StimulationId_NonTarget
Use shrinkage	false
Shrinkage coeficiente (-1 == auto)	-1.0
Shrinkage: Force Diagonal cov (DDA)	false
Number of Partitions for k-fold cross-validation test	12

Tabela 4.4. Campos e valores utilizados para a configuração do classificador LDA

Campo	Valor
Multiclass strategy to apply	OneVsOne
Pairwise Decision Strategy	PKPD
Class 1 label	OVTk_StimulationId_Target
Class 2 label	OVTk_StimulationId_NonTarget
Epsilon	0.100000
SVM Type	C-SVC
Degree	3
Kernel Type	Linear
Epsilon tolerance	0.001000
Cost	1.000000
Cache Size	100.000000
Gamma	0.000000
Nu	0.500000
Shrinking	true
Coef 0	0.000000
Number of Partitions for k-fold cross-validation test	12

Tabela 4.5. Campos e valores utilizados para a configuração do classificador SVM

Após o treinamento do classificador, o `Classifier Trainer` gera um arquivo de configuração contendo as informações resultantes do treinamento, esse arquivo será utilizado pelo `Classifier processor`, que é uma interface genérica para utilização dos classificadores no `OpenViBE`.

4.1.3. Descrição do Cenário de Teste Online

Esse cenário é utilizado para realizar a avaliação dos classificadores e pode ser dividido em três partes. Na primeira parte, ilustrada na Figura 4.5, é possível observar que no início do cenário, algumas caixas estão desconexas das demais (área destacada em vermelho), essas caixas são utilizadas para a configuração do experimento online. Como já foi explicado, o `Target Letter Generator` é utilizado para configurar a sequência de caracteres que serão marcados como alvo e o `P300 Speller Stimulator` é utilizado para a configuração dos estímulos que serão enviados para o usuário, essas caixas e também o `Acquisition client` estão ligados a um `Generic stream writer` para a gravação dos estímulos e sinais captados. Esse cenário grava os sinais e estímulos essencialmente pela necessidade de realizar novamente o treinamento dos classificadores. Isso é recomendável devido ao modo em que o `OpenViBE` executa os cenários.

O `OpenViBE` opera utilizando um relógio simulado, isso significa que ao iniciar o núcleo de execução, que é um laço de repetição contendo todas as caixas do cenário, um tempo é iniciado e cada caixa executará como se esse tempo fosse o tempo corrente do experimento. Isso significa que o tempo real está em progresso

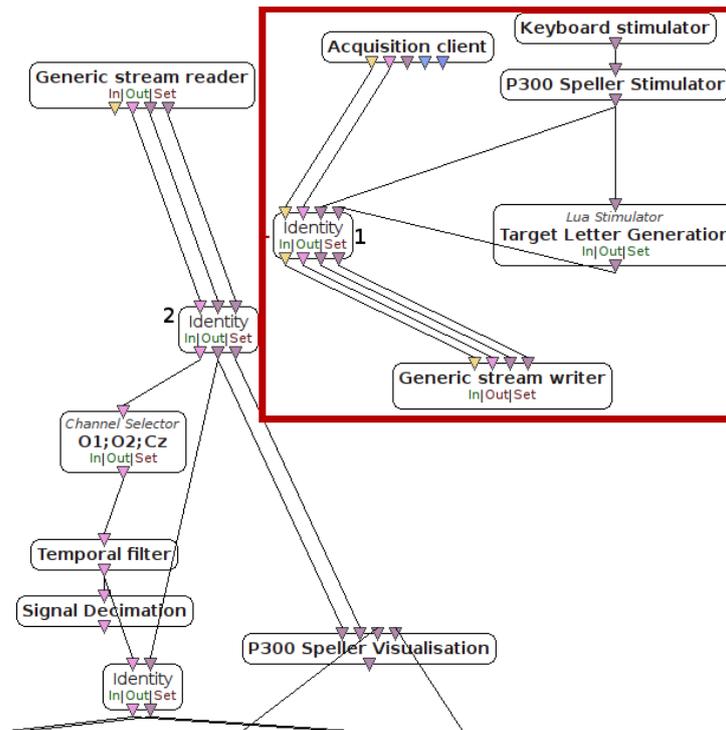


Figura 4.5. Parte 1 do cenário de Teste Online

e ambos os tempos divergem. O tempo que leva para a execução de cada iteração do laço é constante, sendo assim o OpenViBE consegue complementar esse atraso, porém cenários diferentes resultam em constantes diferentes, gerando assim a necessidade da reexecução do cenário de treinamento utilizando um arquivo de dados gravados no cenário de testes.

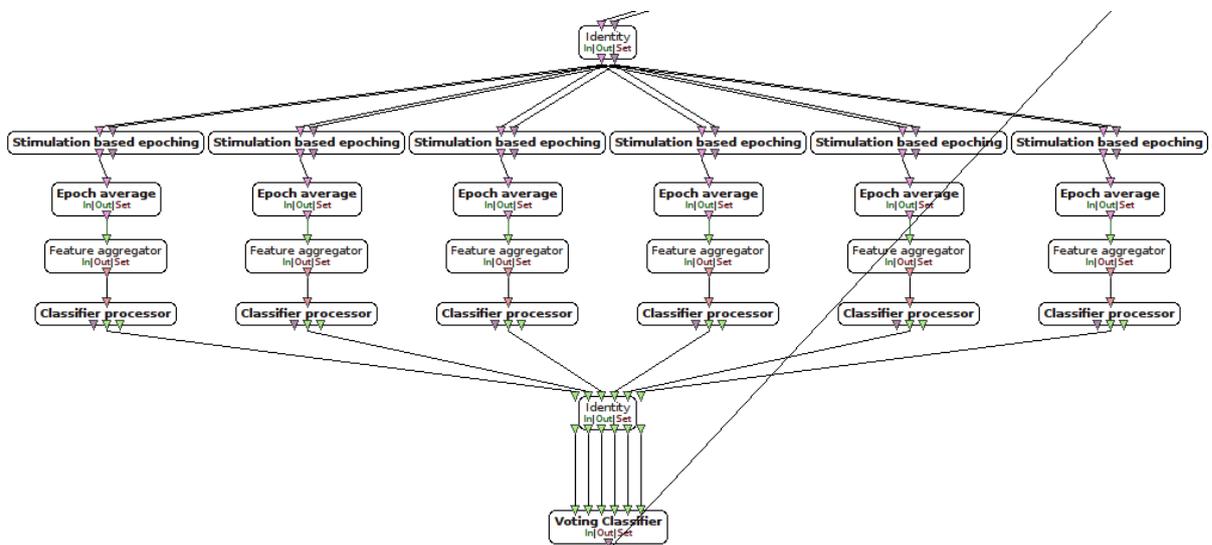


Figura 4.6. Parte 2 do cenário de Teste Online

A Figura 4.5 ilustra o cenário configurado para executar a partir de um arquivo. Para fazer com que esse cenário opere em cima dos dados captados pelo eletroencefalograma é necessário ligar a caixa Identity (1) na caixa Identity (2), assim serão classificados os sinais oriundos do acquisition client e não do arquivo.

Esse cenário apresenta mais três caixas que não foram descritas anteriormente, são elas:

- **Classifier processor:** essa caixa trabalha em conjunto com o **Classifier Trainer** e é responsável pela classificação de novas entradas (vetores de características). Ele tem como entrada um vetor de característica e a saída depende do classificador selecionado, podendo ter três formatos: (1) rótulos, (2) distância entre hiperplanos e (3) valores de probabilidade.
- **Matrix Display:** exibe a matriz fornecida na entrada em uma tabela.
- **Voting Classifier:** o objetivo desta caixa é escolher, dentre as respostas dos classificadores, qual é a mais correta. Por exemplo, se temos N estados e cada um deles podem assumir o valor de ativo ou inativo e cada classificador poderá responder a probabilidade de um estado estar ativo ou inativo, o **Voting Classifier** escolherá qual desses estados é o mais ativo.

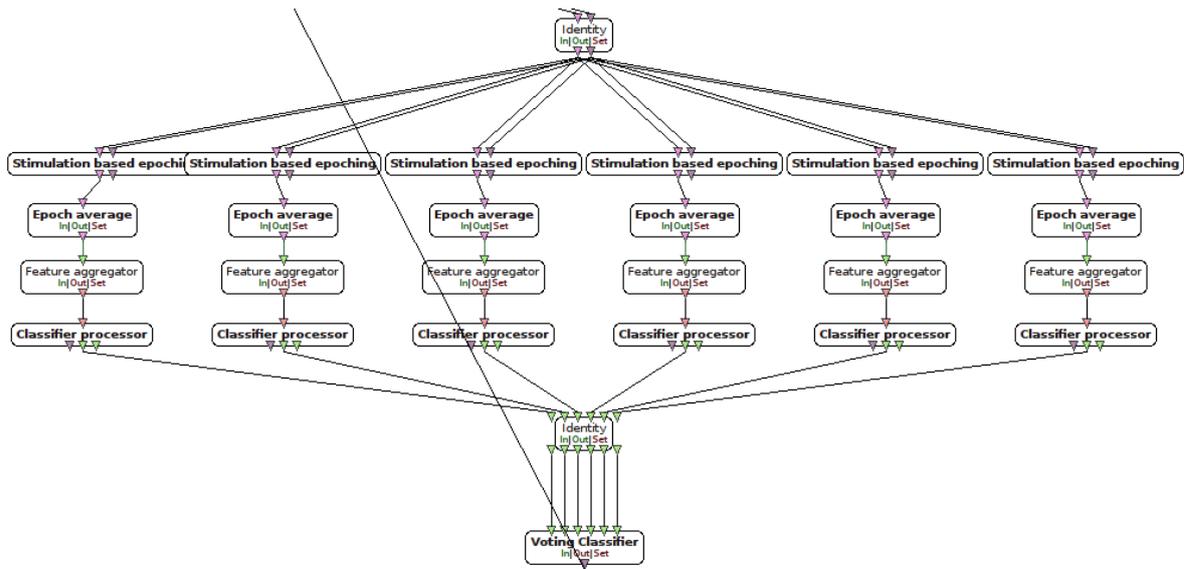


Figura 4.7. Parte 3 do cenário de Teste Online

As partes 2 e 3, vistas nas Figuras 4.6 e 4.7, respectivamente, são semelhantes. A diferença entre elas é que a primeira mostra a classificação das linhas da matriz e a segunda mostra a classificação das colunas. Cada parte possui seis classificadores, que são representados pelas caixas **Classifier processor**, um para cada linha e coluna da matriz e o **Voting Classifier** é utilizado para dizer qual dos seis classificadores tem mais voto de que ocorreu o estímulo P300 naquele instante do tempo, ou seja, cada um dos classificadores dirá a probabilidade do usuário estar olhando no momento em que sua respectiva linha ou coluna piscou, e o **Voting Classifier** dirá qual dos classificadores tem mais chance de estar certo. A Figura 4.8 ilustra o cenário completo, unindo as três partes descritas anteriormente.

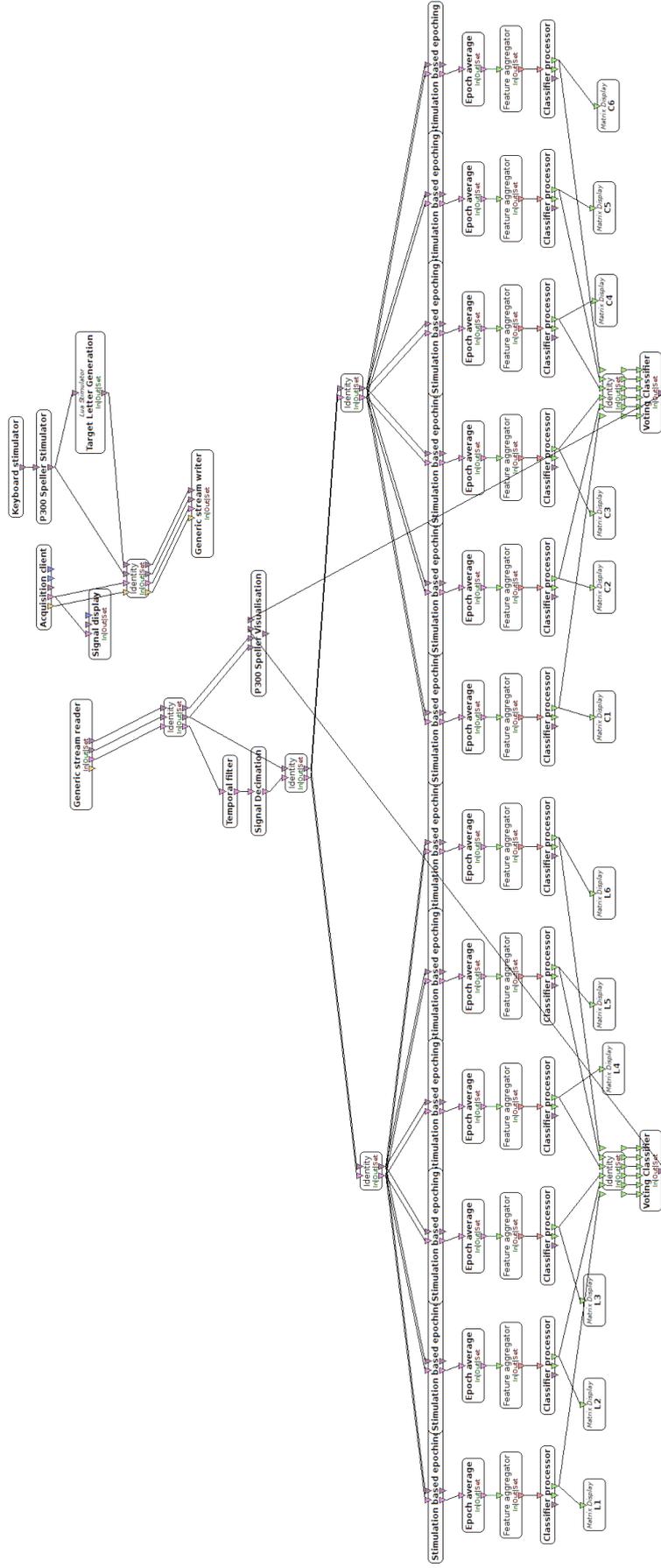


Figura 4.8. Cenário utilizado para o teste online dos classificadores

4.2. Caso 2: Desenvolvimento do cenário P300 Speller xDAWN

O P300 Speller xDAWN possui cenários muito semelhantes ao P300 Speller, explicado na seção anterior. Os cenários são: aquisição de sinais, treinamento dos classificadores, teste *online* e o treinamento do xDAWN que é o cenário adicional utilizado para realizar o cálculo do filtro espacial xDAWN. Esse filtro seleciona, entre os canais utilizados, quais estão detectando mais eficientemente a componente P300. A Figura 4.9 ilustra o cenário utilizado para realizar o treinamento do filtro espacial xDAWN.

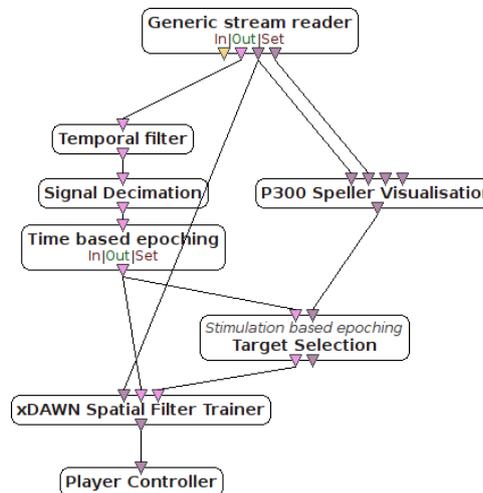


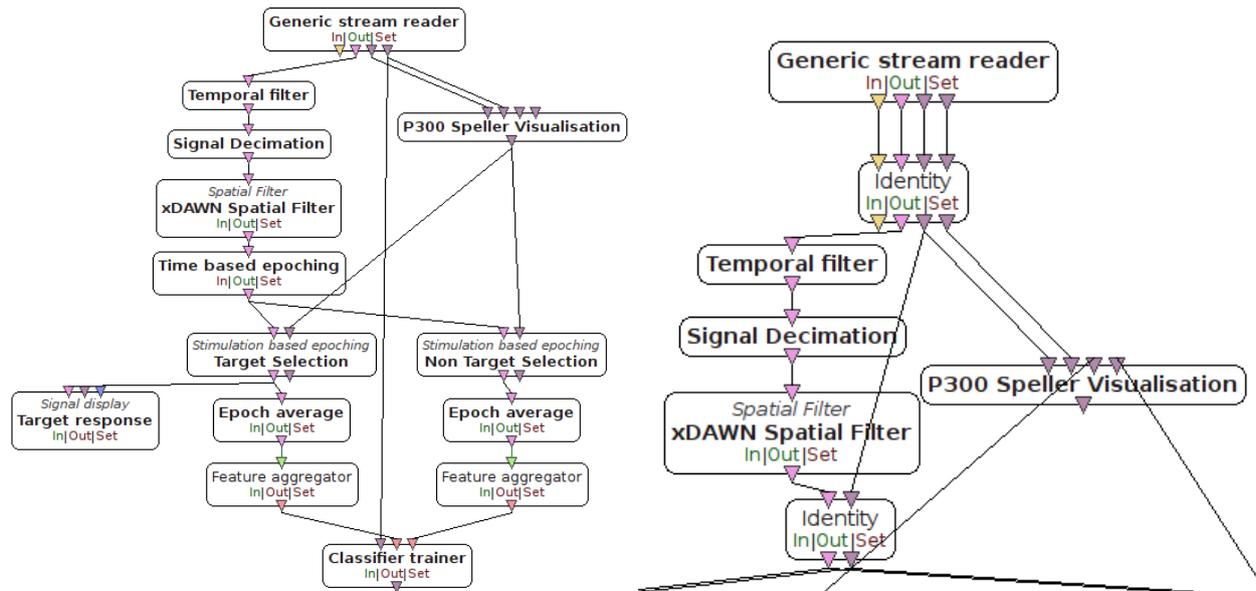
Figura 4.9. Cenário utilizado para o treinamento do filtro espacial xDAWN

No cenário de treino do filtro é adicionado mais duas caixas que não foram utilizadas no Caso 1, são elas:

- **xDAWN Spatial Filter Trainer:** essa caixa é utilizada para calcular o filtro espacial xDAWN, a fim de melhorar a detecção da componente P300, diminuindo a complexidade e a quantidade dos sinais a serem observados para detecção do P300 (Rivet *et al.*, 2009, 2010). Ele é necessário neste experimento para escolher os três melhores canais em cada amostra do sinal obtido. Como no experimento realizado são utilizados seis canais, este filtro busca em cada amostra de cada canal algo próximo ao pico P300 no domínio da frequência. As três melhores amostras (com os melhores picos P300 encontrados) são selecionadas para serem utilizadas na extração de características e classificação do sinal. Este filtro recebe dois parâmetros: o primeiro contendo intervalos de sinais que não possuem a componente P300 e o segundo contendo intervalos de sinais que possuem a componente P300. Em seguida o filtro é calculado.
- **Time based epoching:** essa caixa seleciona amostras do sinal, podendo configurar a duração do intervalo e também a duração entre intervalos.

O cenário de treinamento do filtro xDAWN utiliza o arquivo gravado no cenário de aquisição de sinais e gera um arquivo de configuração contendo o filtro espacial xDAWN que será utilizado pelos demais cenários. Os outros cenários, de treinamento dos classificadores e teste *online*, tem algumas modificações para suportar a adição do filtro.

As alterações presentes nos cenários de treinamento dos classificadores e teste *online* podem ser vistas na Figura 4.10a e 4.10b no qual em ambos são introduzidos a caixa xDAWN Spatial Filter, que é utilizada para a aplicação do filtro espacial no sinal. Na primeira também é introduzida a caixa Time based epoching que seleciona partes dos sinais apenas enquanto está havendo estímulos, removendo as partes do sinal em que não foram enviados nenhum estímulo ao usuário.



(a) Cenário de treinamento dos classificadores com o filtro xDAWN

(b) Cenário de teste online com o filtro xDAWN

Figura 4.10. Cenários utilizados para treinamento e teste utilizando o filtro xDAWN

As configurações das caixas apresentadas nos cenários do Caso 2 são iguais às apresentadas no Caso 1, com exceção da caixa xDAWN Spatial Filter que é calculada automaticamente pelo cenário treinamento do filtro xDAWN

4.3. Discussão dos Resultados

Nas Seções 4.1 e 4.2 foram apresentados os casos P300 Speller e P300 Speller xDAWN respectivamente, em que foram explicados seu funcionamento e as configurações utilizadas na experimentação. Nessa seção será apresentado os resultados obtidos nas execuções.



Figura 4.11. Fluxograma de Execução dos cenários do P300 Speller

Para a execução do Caso 1 foi utilizado o hardware Olimex EEG-SMT com eletrodos nas seguintes posições: Fp1, P3, Fp2 e P4, sendo os dois primeiros para o primeiro canal e os dois últimos para o segundo canal, mais o eletrodo de referência colocado na orelha do usuário. O Caso 1 foi realizado em uma pessoa, na qual foi previamente instruída sobre como seria conduzido o experimento. Primeiramente foi configurado no cenário de Aquisição de Sinais a sequência de caracteres COMPUTER como alvo e iniciado o experimento. Após concluído, o cenário para treinamento do classificador foi executado e o classificador LDA foi treinado. Em seguida foi executado o cenário Teste Online com a mesma sequência de caracteres para a reexecução do treinamento do classificador, devido ao problema apresentado na Seção 4.2. Após o classificador retreinado, foi configurado as sequências de caracteres: BRAIN, LETRA, UTFPR e 20 caracteres aleatórios (que gerou o arquivo denominado aleatorio20), e executados pelo cenário Teste Online gerando seus respectivos arquivos.

A Figura 4.11 ilustra a sequencia de execuções dos cenários, em que exemplifica a repetição do treinamento com o arquivo gerado pelo cenário de teste online.

Os arquivos gerados pelo cenário Teste Online contém os sinais acompanhados das anotações dos estímulos e caracteres alvos, sendo que para cada caractere é feito o reconhecimento três vezes devido a quantidade de repetições configuradas na caixa P300 Speller Stimulator. O arquivos correspondentes a palavra **COMPUTER** e aos vinte caracteres aleatórios foram utilizados como treino e os demais arquivos utilizados para o teste. O experimento foi executado em um computador com processador Intel® Core™ I5 480M de 2.67GHz com 6GB de memória RAM. E o monitor utilizado foi o Philips 196V4LSB2/57 com 18.5 polegadas e resolução de 1366 x 768 píxeis.

	BRAIN		LETRA		UTFPR	
	Acerto	Meio acerto	Acerto	Meio acerto	Acerto	Meio acerto
COMPUTER	0,00	0,33	0,00	0,40	0,00	0,20
aleatorio20	0,07	0,47	0,07	0,33	0,07	0,27

Tabela 4.6. Resultados obtidos para o Caso 1

Os resultados obtidos no Caso 1 podem ser vistos na Tabela 4.6, sendo que a primeira coluna representa os arquivos utilizados para o treinamento, a primeira linha significa os arquivos utilizados para teste e a segunda linha significa o algoritmo utilizado. O Caso 1 foi executado apenas com o classificador LDA, pois quando utilizado o SVM, o cenário de treinamento dos classificadores não funcionou corretamente, quando executado para treinar o SVM o cenário não terminava e também não respondia aos comandos enviados, sendo necessário reiniciar o software **OpenViBE**. Sendo assim o Caso 1 foi executado apenas para o LDA e foram contabilizados a taxa de acertos dos caracteres e também a taxa de “meio acerto”, que significa que o acerto foi apenas para a linha ou coluna do caractere alvo.

Para ambos os casos de teste, os resultados não foram satisfatórios, no qual a melhor taxa de acerto foi de 7%, que corresponde a 1 caractere correto dentre os quinze selecionados. Para o Caso 1, um dos motivos do experimento ter falhado está no sinal mensurado, em que não foi possível configurá-lo de forma que, a partir dos sinais lidos, fosse possível a detecção da componente P300 no sinal. Outro problema foi em relação a quantidade de canais que o hardware possui, tendo apenas dois canais para realizar o mapeamento do lobo parietal, que é a região do cérebro que ocorre a resposta desse padrão.

Com o Caso 2 foram realizados dois testes, no primeiro foi utilizado o hardware **OpenBCI** com eletrodos de prata-cloreto de prata e as possíveis posições foram: P8, P7, P3, P4, Pz, Cz, O1, O2, mais os de referência e terra nas orelhas do usuário. Foram gravados dois arquivos de dez caracteres aleatórios, o **aquisição** e o **online**. O primeiro arquivo foi utilizado como treino para os classificadores e o segundo foi utilizado como teste. Os resultados obtidos nesse primeiro experimento podem ser observado na Tabela 4.7 em que os melhores resultados foram de 60% e 70% para os classificadores LDA e SVM respectivamente.

No segundo teste também foi utilizado o hardware **OpenBCI**, porém com eletrodos passivos revestidos de ouro. Os eletrodos foram dispostos nas posições Cz, O1 e O2, mais os eletrodos de referência colocados em cada uma das orelhas do usuário. Os experimentos foram executados no mesmo computador do Caso 1, porém com um monitor diferente. O monitor utilizado foi um Samsung SyncMaster 940B de 19” e resolução de 1280x1024 píxeis.

Treino	LDA		SVM	
aquisição-2	0,6	0,3	0,5	0,5
aquisição-5	0,6	0,4	0,7	0,3

Tabela 4.7. Resultado do primeiro teste utilizando os cenários descritos no Caso 2

Primeiramente foi executado o cenário de aquisição de sinais para gravar o primeiro arquivo de

treinamento denominado `adquisicao-2`. Para a geração desse arquivo o ambiente foi configurado para gerar dez caracteres aleatórios como alvo. Após a gravação o arquivo `adquisicao-2` foi utilizado para o cálculo do filtro `xDAWN` e também para o treinamento do algoritmo LDA. Em seguida foram gravados mais cinco arquivos utilizando o cenário Teste *Online* com as seguintes configurações: `LETRA`, `UTFPR`, `BRAIN`, 10 caracteres aleatórios (`online-2`) e 20 caracteres aleatórios (`online-20`).

Treino	BRAIN				LETRA				UTFPR			
	LDA		SVM		LDA		SVM		LDA		SVM	
<code>adquisicao2-2</code>	0,20	0,20	0,20	0,60	0,00	0,40	0,00	0,40	0,00	0,40	0,00	0,40
<code>adquisicao2-5</code>	0,40	0,20	0,00	0,33	0,20	0,33	0,13	0,27	0,00	0,33	0,13	0,27
<code>online2-2</code>	0,40	0,20	0,13	0,47	0,00	0,40	0,00	0,27	0,00	0,40	0,00	0,27
<code>online2-5</code>	0,33	0,47	0,07	0,53	0,00	0,20	0,07	0,47	0,00	0,20	0,00	0,27

Tabela 4.8. Resultados das execuções das seqüências de caracteres `BRAIN`, `LETRA` e `UTFPR`.

Treino	online-2				online-20			
	LDA		SVM		LDA		SVM	
<code>adquisicao2-2</code>	0,40	0,20	0,30	0,20	0,05	0,47	0,05	0,47
<code>adquisicao2-5</code>	0,50	0,30	0,17	0,43	0,15	0,40	0,03	0,28
<code>online2-2</code>	-	-	-	-	0,05	0,45	0,15	0,52
<code>online2-5</code>	-	-	-	-	0,20	0,50	0,03	0,20

Tabela 4.9. Resultados das execuções das seqüências de caracteres dos arquivos `online2` e `online20`.

Os resultados do segundo teste realizado com o Caso 2 podem ser vistos nas Tabelas 4.8 e 4.9, no qual a primeira apresenta os resultados para os conjuntos de teste: `BRAIN`, `LETRA` e `UTFPR`, enquanto a segunda tabela apresenta os resultados para os conjuntos de teste: `online2` e `online20`. A primeira coluna representa os conjuntos utilizados para treino dos classificadores, em que foram variados o `decimation factor` da caixa `Signal Decimation`, essa variação resultou nos treinos `adquisicao2-2` e `online2-2` para o `decimation factor` igual a dois e `adquisicao2-5` e `online2-5` para o `decimation factor` igual a cinco. A média geral de acertos podem ser observadas na Tabela 4.10, em que representa o total de acertos para cada caso de teste dividido pela soma total das amostras contidas nos casos de teste.

Treino	LDA		SVM	
<code>adquisicao2-2</code>	0,13	0,36	0,11	0,41
<code>adquisicao2-5</code>	0,24	0,34	0,08	0,32
<code>online2-2</code>	0,09	0,40	0,10	0,44
<code>online2-5</code>	0,16	0,41	0,04	0,30

Tabela 4.10. Média das execuções para cada conjunto utilizado no treino

Para o Caso 2, alguns dos resultados obtidos foram bons, sendo que o melhor caso a taxa de acertos foi de 60% e 70% para o LDA e o SVM, respectivamente. Porém, se observarmos a média geral, esses valores se apresentam bem menores. O LDA acerta mais que o SVM quando o `decimation factor` da caixa `Signal Decimation` é igual a cinco, mas quando abaixamos para dois o SVM e o LDA possuem uma taxa semelhante.

O Caso 2, diferentemente do Caso 1, para alguns testes obteve acertos consideráveis, porém não o suficiente para serem considerados como bons resultados. A única execução que obteve bons resultados, considerando que os caracteres errados foram contabilizados pelos “meio acertos”, foi o primeiro teste descrito utilizando os cenários do Caso 2 (Tabela 4.7). Porém se for observado a média geral, somando os dois testes executados no Caso 2, os valores são baixos, não sendo possível afirmar com certeza a eficiência dos métodos

de classificação utilizados. Para determinar o motivo das baixas taxas obtidas no experimento, deverá ser realizado um estudo mais aprofundado, isolando os possíveis problemas que possam ter acontecido durante a experimentação, sendo necessário realizar mais testes utilizando outras configurações de cenário, outras características para detecção do P300 e até mesmo outros softwares, para assim afirmar os motivos que influenciaram as baixas taxas nos resultados do experimento.

O próximo capítulo apresenta as conclusões e possíveis trabalhos futuros acerca desta monografia.

Conclusões e Trabalhos Futuros

Inicialmente o intuito desse trabalho era determinar dentre os classificadores selecionados qual poderia oferecer a melhor acurácia no experimento proposto, porém ao decorrer do desenvolvimento do experimento, notou-se diversas dificuldades, desde o manuseio com o hardware até as configurações dos cenários no `OpenViBE`. Sendo assim, a maior dificuldade do trabalho foi em entender e construir o experimento proposto.

No decorrer da experimentação foram testados diversas configurações dos cenários, além das apresentadas nesta monografia, com a intenção de descobrir a configuração correta do experimento, definindo parâmetros que o torna-se facilmente replicável, porém não foi possível estabelecer um padrão entre os parâmetros utilizados e os resultados obtidos. Sendo que o motivo pelo qual as taxas não atingiram valores próximos aos encontrados na literatura ainda é desconhecido, necessitando uma maior análise para determinar os fatores que influenciaram negativamente os resultados.

Um dos problemas foi em relação a escolha do software `OpenViBE` para a implementação da aplicação, que apesar de ser um software robusto, não possui uma documentação clara das ferramentas implementadas e também sobre os cenários de exemplos que estão pouco explicativos, o que resultou na demora na adaptação dos cenários para o experimento proposto. Houve também problemas com a utilização dos hardwares, sendo que os primeiros ensaios foram realizados utilizando o `Olimex EEG-SMT` e, como visto na seção de resultados, não obtivemos resultados satisfatórios. Também não foram encontrados trabalhos semelhantes que utilizavam esse hardware para comparação, o que fez com que insistíssemos nas tentativas de conseguir bons resultados. Após os experimentos com o `Olimex EEG-SMT` não funcionar corretamente, iniciamos as tentativas com o `OpenBCI`, que em alguns casos os resultados foram razoáveis, porém distantes dos encontrados na literatura.

Na análise dos resultados, mesmo apresentando baixas taxas de acerto, pode ser observado que para a aplicação `P300 Speller`, o classificador sozinho não é o ponto mais importante, mais sim o conjunto do cenário, sendo que a baixa taxa de acerto do classificador sobre as classes alvo e não-alvo não influenciam diretamente na taxa de acerto dos caracteres, visto que o mais importante é o modo como é feita a junção das respostas dos classificadores. Porém é claro que se o classificador conseguir acertar, com muita certeza, qual a classe a amostra de sinal pertence, influenciará positivamente nos resultados finais.

Com a realização deste trabalho foi possível compreender o funcionamento de uma aplicação que utiliza ICC, realizando a construção de todas as suas etapas, desde a manipulação do hardware para aquisição de sinais até a execução dos comandos na aplicação desejada. Em cada etapa foram encontradas diversas dificuldades na sua construção, o que tornou o trabalho desafiador. Ao final, foi possível compreender o funcionamento tanto dos softwares e dos hardwares utilizados, mesmo que não foi atingido o objetivo de

avaliar os classificadores, os resultados obtidos deram direções para melhorar as taxas de acerto, tais como reavaliar os filtros utilizados no pré-processamento e também as características extraídas, sendo necessários mais testes modificando os componentes utilizados nessas etapas.

Por fim, como trabalhos futuros, poderá ser realizado:

- **Comparação utilizando outros classificadores:** realizar a verificação de como outros classificadores se comportam no experimento.
- **Comparar os classificadores para outros paradigmas:** realizar a comparação dos classificadores em aplicações que utilizam outros paradigmas, como o SSVEP e o *Moto Imagery*, que possuem características diferentes do P300.
- **Analisar outras plataformas de desenvolvimento para as aplicações:** nessa monografia, todos os testes foram realizados utilizando o OpenViBE, porém existem diversos softwares que implementam ferramentas para auxiliar na criação de aplicações que utilizam ICC, tais como BCI2000¹, BCILAB², Tobi³, dentre outros.

¹ <http://www.schalklab.org/research/bci2000>

² <http://sccn.ucsd.edu/wiki/BCILAB>

³ <http://www.tobi-project.org/>

Referências

- BAILLET, Sylvain; MOSHER, John C; LEAHY, Richard M. Electromagnetic brain mapping. *Signal Processing Magazine, IEEE, IEEE*, v. 18, n. 6, p. 14–30, 2001.
- BASHASHATI, Hossein; WARD, Rabab K.; BIRCH, Gary E.; BASHASHATI, Ali. Comparing different classifiers in sensory motor brain computer interfaces. *PLoS ONE*, v. 10, n. 6, p. 1–17, 2015. ISSN 19326203.
- BHATTACHARYYA, Saugat; SENGUPTA, Abhronil; CHAKRABORTI, Tathagatha; BANERJEE, Dhrubojyoti; KHASNOBISH, Anwasha; KONAR, Amit; TIBAREWALA, D. N.; JANARTHANAN, R. EEG controlled remote robotic system from motor imagery classification. *2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012*, n. July, 2012.
- BOSTANOV, Vladimir. Bci competition 2003-data sets ib and iib: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *Biomedical Engineering, IEEE Transactions on, IEEE*, v. 51, n. 6, p. 1057–1061, 2004.
- DALY, Janis J; WOLPAW, Jonathan R. Brain-computer interfaces in neurological rehabilitation. *The Lancet Neurology*, v. 7, n. 11, p. 1032 – 1043, 2008. ISSN 1474-4422.
- FARIA, Tiago João Fernandes da Silva. *Interfaces Cérebro-Computador Utilização do Emotiv EPOC para controlar software lúdico*. Dissertação (Mestrado), 2014.
- FAZEL-REZAI, Reza; AHMAD, Waqas. *P300-based Brain-Computer Interface paradigm design*. California, USA: INTECH Open Access Publisher, 2011.
- FINKE, Andrea; LENHARDT, Alexander; RITTER, Helge. The mindgame: A p300-based brain-computer interface game. *Neural Networks*, v. 22, n. 9, p. 1329 – 1333, 2009. ISSN 0893-6080. Brain-Machine Interface.
- FISHER, Ronald A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.
- FUKUNAGA, Keinosuke. Introduction to statistical pattern recognition. Academic Press Professional, Inc., 1990.
- GALLAGHER, Marcus. *Multi-layer perceptron error surfaces: visualization, structure and modelling*. Tese (Doutorado) — University of Queensland, 2000.
- GRAIMANN, Bernhard; ALLISON, Brendan Z; PFURTSCHELLER, Gert. *Brain-computer interfaces: Revolutionizing human-computer interaction*. Berlin, Germany: Springer Science & Business Media, 2010.
- GUGER, Christoph; DABAN, Shahab; SELLERS, Eric; HOLZNER, Clemens; KRAUSZ, Gunther; CARABALONA, Roberta; GRAMATICA, Furio; EDLINGER, Guenter. How many people are able to control a p300-based brain-computer interface (bci)? *Neuroscience Letters*, v. 462, n. 1, p. 94 – 98, 2009. ISSN 0304-3940.

- HALDER, Sebastian; PINEGGER, Andreas; KÄTHNER, Ivo; WRIESSNEGGER, Selina C; FALLER, Josef; ANTUNES, João B Pires; MÜLLER-PUTZ, Gernot R; KÜBLER, Andrea. Brain-controlled applications using dynamic p300 speller matrices. *Artificial intelligence in medicine*, Elsevier, v. 63, n. 1, p. 7–17, 2015.
- HOFFMANN, Ulrich; VESIN, Jean-Marc; EBRAHIMI, Touradj; DISERENS, Karin. An efficient p300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, v. 167, n. 1, p. 115 – 125, 2008. ISSN 0165-0270. Brain-Computer Interfaces (BCIs).
- KLEM, George H; LÜDERS, Hans Otto; JASPER, HH; ELGER, C. The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, v. 52, n. 3, 1999.
- KOTSIANTIS, S. B.; ZAHARAKIS, I. D.; PINTELAS, P. E. Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, v. 26, n. 3, p. 159–190, 2006. ISSN 02692821.
- LALOR, E; KELLY, SP; FINUCANE, C; BURKE, R; SMITH, R; REILLY, RB; MCDARBY, G. Steady-state vep-based brain-computer interface control in an immersive 3d gaming environment. *EURASIP Journal on Advances in Signal Processing*, v. 2005, n. 19, p. 706906, 2005. ISSN 1687-6180.
- LEBEDEV, Mikhail A.; NICOLELIS, Miguel A L. Brain-machine interfaces: past, present and future. *Trends in Neurosciences*, v. 29, n. 9, p. 536–546, 2006. ISSN 01662236.
- LEGÉNY, J.; VICIANA-ABAD, R.; LÉCUYER, A. Toward contextual ssvpep-based bci controller: Smart activation of stimuli and control weighting. *IEEE Transactions on Computational Intelligence and AI in Games*, v. 5, n. 2, p. 111–116, June 2013. ISSN 1943-068X.
- LOTTE, Fabien. Guide to brain-computer music interfacing. In: _____. London: Springer London, 2014. cap. A Tutorial on EEG Signal-processing Techniques for Mental-state Recognition in Brain-Computer Interfaces, p. 133–161. ISBN 978-1-4471-6584-2.
- LOTTE, F.; CONGEDO, M.; LÉCUYER, A.; LAMARCHE, F.; ARNALDI, B. A review of classification algorithms for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, v. 4, n. 2, p. R1–R13, 2007. Cited By 0.
- MANYAKOV, Nikolay V; CHUMERIN, Nikolay; COMBAZ, Adrien; HULLE, Marc M Van. Comparison of classification methods for p300 brain-computer interface on disabled subjects. *Computational intelligence and neuroscience*, Hindawi Publishing Corp., v. 2011, p. 2, 2011.
- MARSHALL, D.; COYLE, D.; WILSON, S.; CALLAGHAN, M. Games, gameplay, and bci: The state of the art. *IEEE Transactions on Computational Intelligence and AI in Games*, v. 5, n. 2, p. 82–99, 2013. Cited By 0.
- MICHIE, Donald; SPIEGELHALTER, David J; TAYLOR, Charles C. Machine learning, neural and statistical classification. Citeseer, 1994.
- MIRGHASEMI, H.; FAZEL-REZAI, R.; SHAMSOLLAHI, M. B. Analysis of p300 classifiers in brain computer interface speller. In: *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006. p. 6205–6208. ISSN 1557-170X.
- MUNOZ, John E.; CHAVARRIAGA, Ricardo; LOPEZ, David S. Application of hybrid bci and exergames for balance rehabilitation after stroke. In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*, 2014. (ACE '14), p. 67:1–67:4. ISBN 978-1-4503-2945-3.

- NEUPER, Christa; SCHERER, Reinhold; REINER, Miriam; PFURTSCHELLER, Gert. Imagery of motor actions: Differential effects of kinesthetic and visual-motor mode of imagery in single-trial {EEG}. *Cognitive Brain Research*, v. 25, n. 3, p. 668 – 677, 2005. ISSN 0926-6410.
- NICOLAS-ALONSO, Luis Fernando; GOMEZ-GIL, Jaime. Brain computer interfaces, a review. *Sensors, Molecular Diversity Preservation International*, v. 12, n. 2, p. 1211–1279, 2012.
- PERAKAKIS, Manolis; POTAMIANOS, Alexandros. An affective evaluation tool using brain signals. In: *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion*, 2013. (IUI '13 Companion), p. 105–106. ISBN 978-1-4503-1966-9.
- PFURTSCHELLER, G.; NEUPER, C. Motor imagery direct communication. *Proceedings of the IEEE*, v. 89, n. 7, p. 1123–1134, 2001. Cited By 0.
- RAMADAN, Rabie A; REFAT, S; ELSHAHED, Marwa A; ALI, Rasha A. Basics of brain computer interface. In: *Brain-Computer Interfaces*. Cairo, Egypt: Springer, 2015. p. 31–50.
- RENARD, Yann; LOTTE, Fabien; GIBERT, Guillaume; CONGEDO, Marco; MABY, Emmanuel; DELANNOY, Vincent; BERTRAND, Olivier; LÉCUYER, Anatole. Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, MIT Press, v. 19, n. 1, p. 35–53, 2010.
- RESALAT, SeyedNavid; SABA, Valiallah. A practical method for driver sleepiness detection by processing the eeg signals stimulated with external flickering light. *Signal, Image and Video Processing*, Springer London, v. 9, n. 8, p. 1751–1757, 2015. ISSN 1863-1703.
- RIVET, B.; CECOTTI, H.; PLYPO, R.; BERTRAND, O.; MABY, E.; MATTOUT, J. Eeg sensor selection by sparse spatial filtering in p300 speller brain-computer interface. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010. p. 5379–5382. ISSN 1094-687X.
- RIVET, Bertrand; SOULOUMIAC, Antoine; ATTINA, Virginie; GIBERT, Guillaume. xDAWN algorithm to enhance evoked potentials: application to brain-computer interface. *IEEE transactions on bio-medical engineering*, v. 56, n. 8, p. 2035–2043, 2009. ISSN 0018-9294.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: RUMELHART, David E.; MCCLELLAND, James L.; GROUP, CORPORATE PDP Research (Ed.). Cambridge, MA, USA: MIT Press, 1986. cap. Learning Internal Representations by Error Propagation, p. 318–362. ISBN 0-262-68053-X.
- SCHUH, Anderson R.; LIMA, Alessandro; MORCHE, Gustavo; MOSSMANN, João; BEZ, Marta R. Control of a wheelchair simulator in a three-dimensional environment using eye blink detection through non-invasive brain-computer interface. In: *Proceedings of the XV International Conference on Human Computer Interaction*, 2014. (Interacci#243;n '14), p. 76:1–76:2. ISBN 978-1-4503-2880-7.
- SOMAN, Sumit; SRIVASTAVA, Siddharth; SRIVASTAVA, Saurabh; RAJPUT, Nitendra. Brain computer interfaces for mobile apps: State-of-the-art and future directions. *CoRR*, abs/1509.01338, 2015.
- VAPNIK, Vladimir N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.
- WALDERT, Stephan; PISTOHL, Tobias; BRAUN, Christoph; BALL, Tonio; AERTSEN, Ad; MEHRING, Carsten. A review on directional information in neural signals for brain-machine interfaces. *Journal of Physiology-Paris*, Elsevier, v. 103, n. 3, p. 244–254, 2009.

WESTON, Jason; WATKINS, Chris. *Multi-class support vector machines*. Egham, UK, 1998.

WOLPAW, J.R. Brain-computer interfaces: signals, methods, and goals. In: Ieee. *Neural Engineering, 2003. Conference Proceedings. First International IEEE EMBS Conference on*, 2003. p. 584–585.

WOLPAW, Jonathan R; BIRBAUMER, Niels; HEETDERKS, William J; MCFARLAND, Dennis J; PECKHAM, P Hunter; SCHALK, Gerwin; DONCHIN, Emanuel; QUATRANO, Louis A; ROBINSON, Charles J; VAUGHAN, Theresa M *et al.* Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, Citeseer, v. 8, n. 2, p. 164–173, 2000.

ZANDER, Thorsten O; KOTHE, Christian; JATZEV, Sabine; GAERTNER, Matti. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. In: *Brain-computer interfaces*. Berlin, Germany: Springer, 2010. p. 181–199.