

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RAFAEL ALENCAR OLIVEIRA

**INVESTIGAÇÃO DA RELAÇÃO DOS MÉTODOS DE EXTRAÇÃO DE
CARACTERÍSTICAS E OS MOVIMENTOS DAS MÃOS, NO PROCESSAMENTO
DE SINAIS SEMG**

CAMPO MOURÃO

2022

RAFAEL ALENCAR OLIVEIRA

**INVESTIGAÇÃO DA RELAÇÃO DOS MÉTODOS DE EXTRAÇÃO DE
CARACTERÍSTICAS E OS MOVIMENTOS DAS MÃOS, NO PROCESSAMENTO
DE SINAIS SEMG**

**Survey On the Relationship of Feature Extraction Methods and Hands
Movement in sEMG Signal Processing**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Orientador: Prof. Dr. Rodrigo Hübner

Coorientador: Prof. Dr. Lucas Ricken Garcia

CAMPO MOURÃO

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RAFAEL ALENCAR OLIVEIRA

**INVESTIGAÇÃO DA RELAÇÃO DOS MÉTODOS DE EXTRAÇÃO DE
CARACTERÍSTICAS E OS MOVIMENTOS DAS MÃOS, NO PROCESSAMENTO
DE SINAIS SEMG**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Data de aprovação: 06/junho/2022

Osmar Tormena Júnior
Doutorado
Universidade Tecnológica Federal do Paraná

Juliano Henrique Foleiss
Doutorado
Universidade Tecnológica Federal do Paraná

Rodrigo Hübner
Doutorado
Universidade Tecnológica Federal do Paraná

Lucas Ricken Garcia
Doutorado
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2022

Dedico este trabalho à minha família, por viabilizarem e apoiarem meus estudos, desde o princípio.

AGRADECIMENTOS

Um primeiro agradecimento ao meu orientador Prof. Dr. Rodrigo Hübner por, desde o início, prestar todo o apoio necessário para o desenvolvimento do trabalho. Sem nossas conversas no *discord* durante as madrugadas esse trabalho não teria saído. Ao meu professor coorientador, Prof. Me. Lucas Ricken, obrigado por toda solicitude de sempre.

Aos professores do Campus, pela oportunidade de aprender com todos o necessário para viabilizar esse trabalho e, ainda, tornar minha colocação no mercado profissional viável. Obrigado pelo duro trabalho de todos.

Um agradecimento aos meus amigos que me acompanharam durante todo o período de faculdade; o apoio emocional e acadêmico de vocês foi essencial para que eu chegasse até aqui. Bruno, Adriana (vulgo moção), Pedro, Marlon, Isabela, Gabriela, Anderson, Lucas e Angélica...vocês foram e continuam sendo demais! À Nayara, minha melhor amiga virtual que eu poderia ter, obrigado por toda a motivação e por acreditar sempre no meu potencial, desde o ensino médio. Mesmo longe, você foi esplêndida.

À minha família, fundamental para esse marco, meu agradecimento mais especial. Aos meus pais, sei de todo o trabalho duro de vocês para que eu pudesse estudar longe da minha cidade. À minha madrinha e meu irmão, Gabriel, por sempre me ajudarem (principalmente) financeiramente, quando a coisa ficava um pouco apertada. Ao Leo, meu namorado, obrigado por toda motivação prestada, principalmente nessa reta final. Seus puxões de orelha para que eu fizesse o TCC foram essenciais.

RESUMO

Diversas são as técnicas para para medição de potenciais elétricos e magnéticos cuja origem é o corpo humano. Dentre essas técnicas, destaca-se a técnica de eletromiografia, também conhecida como EMG. Os sinais de EMG são comumente retirados do corpo humano através de eletrodos não invasivos (isto é, superficialmente na pele) que são dispostos sobre os músculos de interesse. Ao processar esses sinais, é possível classificá-los, ou seja, identificar os tipos de movimento que o paciente está executando. Uma boa classificação depende de etapas anteriores, como pré-processamento e extração de características. Esse trabalho teve como objetivo investigar quais as características apresentavam melhor e pior desempenho na classificação de diferentes sinais. Para isso, foram processados quatro *datasets* contendo diferentes movimentos das mãos realizados no dia a dia. Ao fim do trabalho, através de análises qualitativas e quantitativas das características implementadas, concluiu-se que algumas características temporais não são interessantes para o processamento, como TM3, TM4 e TM5 enquanto outras, como detector logarítmico, se destacaram. As características na frequência apenas se mostraram interessantes em um *dataset* cuja frequência de amostragem de dados era muito superior às outras.

Palavras-chave: semg; extração de características; classificação.

ABSTRACT

There are several techniques for measuring electric and magnetic potentials that originate in the human body. Among these techniques is the electromyography technique, also known as EMG. EMG signals are commonly taken from the human body using non-invasive electrodes (i.e., superficially on the skin) that are placed over the muscles of interest. By processing these signals, it is possible to classify them, i.e. to identify the types of movement that the patient is performing. A good classification depends on previous steps, such as pre-processing and feature extraction. This work aimed to investigate which features performed best and worst in classifying different signals. For this, four *datasets* containing different hand movements performed in everyday life were processed. At the end of the work, through qualitative and quantitative analysis of the implemented features, it was concluded that some temporal features are not interesting for processing, such as TM3, TM4 and TM5 while others, such as logarithmic detector, stood out. The frequency features only proved interesting on a *dataset* whose data sampling frequency was much higher than the others.

Keywords: semg; feature extraction; classification.

LISTA DE FIGURAS

Figura 1 – Exemplo de um sinal EMG.	17
Figura 2 – Esquema usual de um sistema BCI baseado em sinais EMG	18
Figura 3 – Exemplo de janelamento de um sinal EMG	20
Figura 4 – Conceito Geral de uma SVM	26
Figura 5 – Exemplo de uma matriz de confusão para classificação binária	26
Figura 6 – Fluxograma geral do trabalho	28
Figura 7 – Movimentos realizados e respectivas posições dos membros (P1 até P5) . .	30
Figura 8 – 6 Movimentos realizados no dataset citado	31
Figura 9 – Movimentos do dataset	31
Figura 10 – Disposição dos eletrodos no antebraço	32
Figura 11 – Representação esquemática da musculatura e eletrodos associados.	33
Figura 12 – Disposição dos eletrodos durante a execução dos experimentos.	33
Figura 13 – Espectro de potência para o primeiro <i>Dataset</i>	36
Figura 14 – Espectro de potência para o segundo <i>Dataset</i>	36
Figura 15 – Espectro de potência para o segundo <i>Dataset</i> após filtro notch	37
Figura 16 – Espectro de potência para o segundo <i>Dataset</i> após filtro passa-faixas	37
Figura 17 – Espectro de potência para o terceiro <i>Dataset</i>	37
Figura 18 – Espectro de potência para o quarto <i>Dataset</i>	38
Figura 19 – Espectro de potência para o terceiro <i>Dataset</i> após filtragem	38
Figura 20 – Dimensões do vetor de dados após as sucessivas etapas	41
Figura 21 – Novo <i>shape</i> dos dados.	42
Figura 22 – Métodos para segmentação dos sinais	44
Figura 23 – Parte das amostras de resultados de acurácia - Participante 1	45
Figura 24 – Pequena amostra de resultados de acurácia (frequência) - Participante 1 . .	46
Figura 25 – Performance da característica <code>logdetector</code> para os movimentos do participante 2.	53
Figura 26 – Relatório de classificação para <code>logdetector</code> para os movimentos do participante 2.	54
Figura 27 – Performance da característica <code>logdetector</code> para os movimentos do participante 4.	56

Figura 28 – Performance da característica <code>logdetector</code> para os movimentos do participante 3.	57
Figura 29 – Relatórios de Classificação - <i>Dataset 1</i>	76
Figura 30 – Relatórios de Classificação - <i>Dataset 2</i>	76
Figura 31 – Relatórios de Classificação - <i>Dataset 3</i>	77
Figura 32 – Relatórios de Classificação - <i>Dataset 4</i>	77

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
1.2	Justificativas	15
2	REFERENCIAL TEÓRICO	16
2.1	Eletromiografia e Interfaces Cérebro-Computador	16
2.1.1	Biossinais	16
2.1.2	Princípios de Eletromiografia	16
2.1.3	Interfaces Cérebro-Computador	17
2.2	Aquisição de sinais EMG	18
2.3	Pré-Processamento de Sinais	18
2.3.1	Filtragem do Sinal	19
2.3.2	Janelamento do Sinal	19
2.4	Extração de Características	19
2.4.1	Técnicas de extração de características temporais	20
<u>2.4.1.1</u>	<u>IEMG Integrado (IEMG, do inglês <i>Integrated EMG</i>)</u>	<u>20</u>
<u>2.4.1.2</u>	<u>Valor Absoluto Médio (MAV, do inglês <i>Mean Absolute Value</i>)</u>	<u>21</u>
<u>2.4.1.3</u>	<u>Valor Absoluto Médio Modificado - Tipo 1</u>	<u>21</u>
<u>2.4.1.4</u>	<u>Valor Absoluto Médio Modificado - Tipo 2</u>	<u>21</u>
<u>2.4.1.5</u>	<u>Quadrado Integral Simples (SSI, do inglês <i>Simple Square Integral</i>)</u>	<u>22</u>
<u>2.4.1.6</u>	<u>Variância do EMG</u>	<u>22</u>
<u>2.4.1.7</u>	<u>Valores Absolutos do terceiro, quarto e quinto momentos temporais</u>	<u>22</u>
<u>2.4.1.8</u>	<u>Valor Quadrático Médio</u>	<u>22</u>
<u>2.4.1.9</u>	<u>Detector Logarítmico</u>	<u>23</u>
<u>2.4.1.10</u>	<u>Tamanho da forma de Onda</u>	<u>23</u>
<u>2.4.1.11</u>	<u>Mudança Média de Amplitude</u>	<u>23</u>
<u>2.4.1.12</u>	<u>Valor do Desvio Padrão da Diferença Absoluta</u>	<u>23</u>
2.4.2	Técnicas de Extração no Domínio da Frequência	23
<u>2.4.2.1</u>	<u>Frequência Média</u>	<u>23</u>

2.4.2.2	Frequência de Pico	24
2.4.2.3	Frequência Mediana	24
2.4.2.4	Energia Média	24
2.4.2.5	Zero, primeiro, segundo e terceiro momentos espectrais	24
2.4.2.6	Variância da Frequência Central	25
2.5	Classificação	25
2.5.1	Algoritmos de Classificação	25
2.5.2	Métricas de Avaliação em Machine Learning	26
3	MATERIAIS E MÉTODOS	28
3.1	Fluxograma Geral do Processo de Análise	28
3.2	Descrição dos <i>Datasets</i> Utilizados	29
3.2.1	<i>Dataset 1: Towards Limb Position Invariant Myoelectric Pattern Recognition Using time-dependent spectral features</i>	29
3.2.2	<i>Dataset 2: EMG based classification of basic hand movements based on time-frequency features</i>	30
3.2.3	<i>Dataset 3: Dataset for multi-channel surface electromyography (sEMG) signals of hand gestures</i>	30
3.2.4	Dataset 4: Dataset disponibilizado Prof. Rodrigo Hübner (Ciência da Computação)	32
3.3	Leitura dos <i>Datasets</i>	33
3.4	Pré Processamento de Sinais	34
3.4.1	Filtragem dos sinais	34
3.4.1.1	<i>Dataset 1</i>	35
3.4.1.2	<i>Dataset 2</i>	36
3.4.1.3	<i>Dataset 3</i>	37
3.4.1.4	<i>Dataset 4</i>	38
3.4.2	Janelamento dos sinais	38
3.5	Extração de Características	40
3.5.1	Implementação	40
3.5.2	Preparação para Classificação	41
3.6	Classificação dos Sinais	43
3.7	Visualização/Análise dos Resultados	43

4	RESULTADOS	45
4.1	<i>Dataset 1</i>	45
4.1.1	Visão Geral das características no domínio do tempo	45
4.1.2	Visão Geral das Características no domínio da frequência	46
4.1.3	Investigação das melhores características e suas relações com os movimentos	47
4.2	<i>Dataset 2</i>	49
4.2.1	Visão Geral das características no domínio do tempo	49
4.2.2	Visão Geral das Características no domínio da frequência	49
4.2.3	Investigação das melhores características e suas relações com os movimentos	50
4.3	<i>Dataset 3</i>	51
4.3.1	Visão Geral das características no domínio do tempo	51
4.3.2	Visão Geral das Características no domínio da frequência	51
4.3.3	Investigação das melhores características e suas relações com os movimentos	52
4.4	<i>Dataset 4</i>	54
4.4.1	Visão Geral das características no domínio do tempo	54
4.4.2	Visão Geral das Características no domínio da frequência	54
4.4.3	Investigação das melhores características e suas relações com os movimentos	55
5	CONCLUSÃO	58
	REFERÊNCIAS	60
	APÊNDICE A CÓDIGO UTILIZADO PARA PROCESSAMENTO DOS DA-	
	DOS	64
	APÊNDICE B RELATÓRIOS DE CLASSIFICAÇÃO DAS MELHORES	
	COMBINAÇÕES	76

1 INTRODUÇÃO

Como consequência dos avanços da engenharia biomédica no século XX, nos dias de hoje, muitas são as técnicas para medição de potenciais elétricos e magnéticos de várias regiões do corpo humano. Dentre essas técnicas, destaca-se a técnica de eletromiografia (EMG), que tem como função monitorar movimentos musculares através dos potenciais elétricos (de aqui em diante chamados de sinais EMG) provindos destes. Os sinais são comumente por meio de através da disposição de eletrodos de superfície sob a pele dos músculos desejados (NEGI; KUMAR; MISHRA, 2016).

Os sinais de EMG estão sendo extensivamente utilizados para dois propósitos: diagnóstico de doenças e reabilitação. No primeiro caso, testes utilizando EMG podem ser conduzidos para auxiliar o diagnóstico de doenças como a síndrome do túnel do carpo (LUKACS; PIETRIKOVA, 2020), doenças musculares (SENGAR; DUTTA; BURGET, 2017) e neuropatias (ULLAH; IQBAL, 2020), além de servir como ferramenta para excluir outras patologias, como a fibromialgia (NEUROCLIN). Já no segundo caso, os sinais de EMG são muito utilizados em sujeitos amputados para o controle próteses. Mesmo que uma pessoa não possua um membro, sua mente enviará impulsos para tentar mover essa parte do corpo caso ele tente mover. Eletrodos de superfície poderiam ser dispostos no ombro do sujeito para tentar captar sinais elétricos responsáveis por mexer seu antebraço e mão, por exemplo. Desta forma, uma prótese pode ser ajustada no paciente e movida apenas com o apoio dos sinais EMG detectados (PATEL *et al.*, 2019).

Extrair e interpretar as informações contidas dos sinais de EMG é fundamental para atingir esses propósitos citados anteriormente. Após a aquisição do sinal de EMG, onde detecta-se o início e o fim da atividade eletromiográfica, é possível determinar quais os tipos de movimentos que o usuário está fazendo, bem como a região corpórea de origem destes, por meio de técnicas de processamento de sinais, que usualmente envolve reconhecimento de padrões. Essas técnicas são agrupadas em três grupos: pré-processamento, extração de características e classificação (SPIEWAK, 2018). A primeira etapa compreende técnicas de organização dos dados, filtragem e janelamento de sinais, enquanto a segunda, a extração de características, consolida os dados temporais em padrões identificáveis pelos algoritmos da próxima etapa, a classificação, que tem como objetivo central nesse contexto identificar os movimentos desempenhados (ou a intenção destes) por um sujeito (ZARDOSHTI-KERMANI *et al.*, 1995).

O cerne da etapa de extração de características, objeto desse estudo, consiste no fato de que é, muitas vezes, mais conveniente mapear um determinado conjunto de dados de entrada de um sistema em outro espaço com informações mais relevantes, de forma que o trabalho com estes dados se torne mais eficiente. (SIJIANG; VUSKOVIC, 2004). Segundo a literatura, três são as principais abordagens para se extrair características de um conjunto de dados EMG: no domínio do tempo, no domínio da frequência e tempo-frequência. As técnicas de extração de características temporais são mais rapidamente calculadas pois os dados utilizados não precisam

ser transformados. Já para as técnicas utilizando o domínio da frequência, as características são geralmente calculadas a partir da densidade espectral de potência dos dados temporais segmentados no domínio da frequência (ZECCA *et al.*, 2002).

A extração de características de sinais EMG através de técnicas no domínio tempo-frequência geralmente acontecem por Transformadas Wavelet (PHINYOMARK; LIMSAKUL; PHUKPATTARANONT, 2011a). Enquanto a análise em frequência através da transformada de Fourier não permite a análise do sinal transformado no domínio do tempo, a transformada Wavelet permite a análise de bandas de frequência preservando a resolução temporal (WEIDERPASS *et al.*, 2013).

A extração de características utilizando Transformadas Wavelet não será considerada no escopo desse estudo devido sua alta capacidade de processamento envolvida e, ainda, a necessidade de algoritmos de redução de dimensionalidade do vetor de característica antes da etapa de classificação, visto que essa abordagem fornece um grande vetor de características (WANG *et al.*, 2006). Portanto, no presente trabalho, serão consideradas apenas técnicas no domínio do tempo e no domínio da frequência para a extração de características.

1.1 Objetivos

Nesse tópico serão explanados os objetivos gerais e específicos realizados durante o desenvolvimento do trabalho.

1.1.1 Objetivo Geral

O objetivo deste trabalho é investigar a relação entre diferentes técnicas de extração de características na classificação de movimentos úteis no dia a dia de uma pessoa amputada através do levantamento de *datasets* disponíveis na literatura.

1.1.2 Objetivos Específicos

- Fazer um estudo teórico das condições de aquisição de dados de EMG para cada *dataset* utilizado;
- Realizar o pré-processamento e processamento dos dados de cada *dataset*, o que inclui a implementação das técnicas clássicas temporais e de frequência para a extração de características e algoritmos de classificação;
- Investigar quais características são mais apropriadas para um conjunto de movimentos realizados no do dia-dia de modo geral analisando-se a acurácia;

- Investigar quais características são mais apropriadas para quais tipos de movimentos do dia a dia singularmente, por meio da matriz de confusão e/ou graficamente

1.2 Justificativas

Alguns trabalhos já buscaram investigar a relação entre diferentes métodos de extração de características de sinais EMG e suas implicações na acurácia no modelo de classificação, ou seja, na capacidade do classificador acertar o movimento realizado para um determinado conjunto de dados de entrada. Zardoshti-Kermani *et al.* (1995), seleciona algumas características temporais e no domínio da frequência para avaliar a separabilidade máxima das classes fornecidas pelas características, o que pode ser traduzida na acurácia do modelo. Os movimentos feitos realizados em diferentes níveis de força empregados na contração de bíceps e tríceps. Phinyomark, Limsakul e Phukpattaranont (2011a) usaram 37 técnicas diferentes de extração de características para avaliar dois *datasets*, onde cada *dataset* tinha um propósito diferente; o primeiro para avaliar a redundância das características e o segundo para buscar as características ótimas. Os movimentos feitos foram mão aberta, mão fechada, extensão do pulso, flexão do pulso, pronação e supinação do braço. Lolure e Thool (2015), em seu trabalho de extração de características baseado em transformada Wavelets, utilizou movimentos dos dedos, cotovelo e pulso em seus estudos.

Como pode-se notar em alguns trabalhos apresentados, não há uma padronização de movimentos utilizados para o estudo das diversas técnicas de extração de características e, ainda, muitos movimentos acabam não sendo úteis para uma aplicação de prótese para um amputado, por exemplo. Além disso, nenhum deles estuda o desempenho de cada característica utilizada na classificação em relação a cada movimento específico, dizendo se determinada característica ou conjunto de características seria adequado para um conjunto específico de movimentos. Este trabalho pretende investigar esses pontos através da análise de *datasets* que possuem movimentos úteis realizados no dia a dia.

2 REFERENCIAL TEÓRICO

Este capítulo elucidada o tema abordado, contextualizando o sinal EMG e suas características. Depois, adentra nas técnicas de pré-processamento e processamento do sinal, dando foco especial no assunto de extração de características, assunto principal deste trabalho.

2.1 Eletromiografia e Interfaces Cérebro-Computador

Os sinais biológicos, também chamados de biossinais, têm sido protagonistas em diversas áreas, como diagnóstico médico, reabilitação e monitoramento passivo de saúde. Um profundo entendimento destes tipos de sinais permite que diversas aplicações possam ser desenvolvidas nas áreas mencionadas. Como exemplo, pode-se citar o uso de sinais EMG em aplicações de interfaces cérebro computador para reabilitação de pessoas amputadas através do controle de próteses (SUPRATAK *et al.*, 2016).

2.1.1 Biossinais

O termo biossinal se aplica a qualquer sinal que pode ser adquirido de um ser vivo, de forma que possa ser medido e monitorado. Supratak *et al.* (2016) classificam os biossinais em dois tipos: bioelétricos e não-bioelétricos. A primeira classe de sinais reside em tecidos eletricamente ativos, como músculos e nervos, e geralmente possuem sua origem de mudanças na corrente elétrica geradas por diferentes potenciais entre os tecidos. Exemplos de sinais elétricos são sinais provindos da atividade cerebral (sinais de eletroencefalografia), sinais provindos do coração (sinais de eletrocardiografia) e sinais provindos de músculos, EMG, objeto deste trabalho. Já no grupo dos sinais não-bioelétricos, podem ser citados os sinais acústicos (fonocardiograma, som da respiração), sinais mecânicos (sinais de MMG, conhecidos como mecanomiografia (ISLAM *et al.*, 2013), sinais magnéticos (sinais do campo magnético cardíaco, por exemplo, conhecido como magnetocardiografia – MCG (GESELOWITZ, 1979) e sinais ópticos (sinais de fotopletismografia, PPG), (KAVYA *et al.*, 2020).

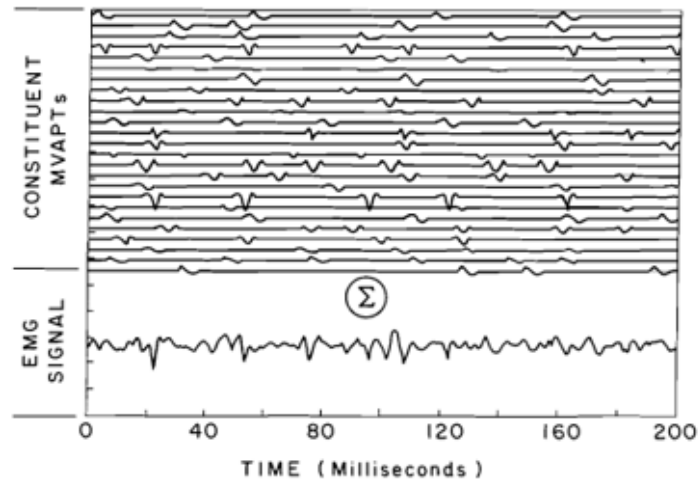
2.1.2 Princípios de Eletromiografia

Para que as contrações musculares aconteçam, várias componentes nervosas e fibras precisam trabalhar em conjunto, ou seja, constantemente ativadas. Cada componente recebe o nome de unidade motora. Uma unidade motora é composta por neurônios motores periféricos e o conjunto de suas fibras musculares associadas. Sempre que acontece uma despolarização do neurônio motor, o potencial gerado por esta despolarização se propaga através de seu axônio e, como consequência, todas as fibras ligadas a este neurônio também se des-

polarizam gerando potenciais, que são somados e captados por eletrodos específicos (FARINA; HOLOBAR, 2016).

A Figura 1 abaixo mostra sinais produzidos por 25 unidades motoras durante 0,2 segundos. A Soma dos sinais é o sinal EMG

Figura 1 – Exemplo de um sinal EMG.



Fonte: Adaptado de Basmajian e Luca (1985).

Matematicamente e idealmente, Basmajian e Luca (1985) dizem que o sinal EMG pode ser definido como a soma dos potenciais gerados por cada unidade motora:

$$m_p(T,F) = \sum_{i=1}^p u_i(f,P) \quad (1)$$

onde $f(t,F)$ é o sinal de EMG observável, p é o número de unidades motoras, i simboliza cada unidade motora e $u_i(t)$ cada trem de potenciais motores de ação (MUAPS - *Motor Unit Action Potential Trains*). A equação mostra que o sinal EMG é uma função do tempo (t) e força (F). Na aquisição do sinal, o sinal ainda sofrerá a adição de ruído elétrico e, ainda, sofrerá por propriedades filtradoras de eletrodos $r(t)$, resultando no sinal $m(t,F)$, o sinal EMG não-ideal

A amplitude do sinal EMG varia entre 1 mV e 10 mV e apresenta comportamento estocástico, ou seja, aleatório em condições normais, e pode razoavelmente ser apresentado por uma função gaussiana de distribuição. Além disso, o espectro de frequência varia entre 0 e 500 Hz para eletrodos de superfície e de 0 a 10 kHz para eletrodos invasivos (BASMAJIAN; LUCA, 1985).

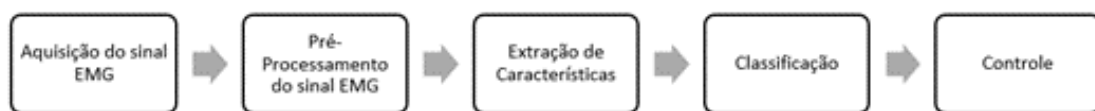
2.1.3 Interfaces Cérebro-Computador

Uma interface cérebro-computador, BCI, (do inglês, *Brain Computer Interface*) pode ser definida como uma conexão entre um computador e o cérebro humano, substituindo dispositivos de entrada comuns como teclado e mouse. Em grande parte dos trabalhos realizados, a inter-

face cérebro computador utiliza ondas cerebrais coletadas a partir de um eletroencefalograma e realiza um processamento nos sinais coletados para identificar padrões e, então, dependendo do padrão identificado é associado algum comando em um atuador, como alguma prótese ou cadeira de rodas (RAMADAN *et al.*, 2015). Spiewak (2018) relaciona o conceito de BCI e sinais EMG, vistos que estes, apesar de não serem coletados na região cerebral, tem suas origens neurológicas.

Baseado em Spiewak (2018), pode-se sumarizar o controle de algum dispositivo com interface humana baseada em sinais EMG de acordo com a figura 2 abaixo:

Figura 2 – Esquema usual de um sistema BCI baseado em sinais EMG



Fonte: Adaptado de Spiewak (2018).

Cada etapa mostrada na figura 1 será fundamentada nas sessões seguintes, especialmente a etapa de extração de característica, onde reside os objetivos de estudo deste trabalho.

2.2 Aquisição de sinais EMG

A aquisição dos sinais EMG pode ser feita, geralmente, pela disposição de dois tipos de eletrodos nos músculos das regiões de interesse: eletrodos de superfície ou eletrodos de inserção. Os eletrodos de superfície, dispostos sob a pele, fornecem maior facilidade para aquisição dos sinais, porém não é possível obter sinais provindos de músculos menores. Também são muito utilizados quando se desejam obter sinais para o controle de próteses. O segundo grupo de eletrodos, de inserção, são divididos em dois grupos: eletrodos do tipo agulha e do tipo fio. Eletrodos do tipo de inserção são utilizados quando se deseja detectar MUAPS individuais e, além disso, permite a aquisição desses sinais durante pequenas contrações musculares. Entretanto, esses tipos de eletrodos causam desconforto no indivíduo e, portanto, não são adequados para usos frequentes (PANCHOLI; AGARWAL, 2016).

2.3 Pré-Processamento de Sinais

O propósito da etapa de pré-processamento de sinais é transformar o sinal EMG cru, da etapa de aquisição, e torná-lo adequado para a etapa e extração de características e, posteriormente, ser classificado (reconhecimento de padrões).

2.3.1 Filtragem do Sinal

O propósito da etapa de filtragem do sinal EMG é melhorar sua relação sinal-ruído e, conseqüentemente, aumentar a acurácia na etapa de classificação dos sinais. Usualmente, em sinais EMG, são aplicados filtros seletivos em frequência como o filtro *Notch* para a retirada de ruídos provindos da rede elétrica, com frequência de corte centrada em 50 Hz ou 60 Hz. Usa-se também um filtro passa-faixa para limitar a banda do sinal apenas à útil, geralmente entre 10 e 500 Hz.

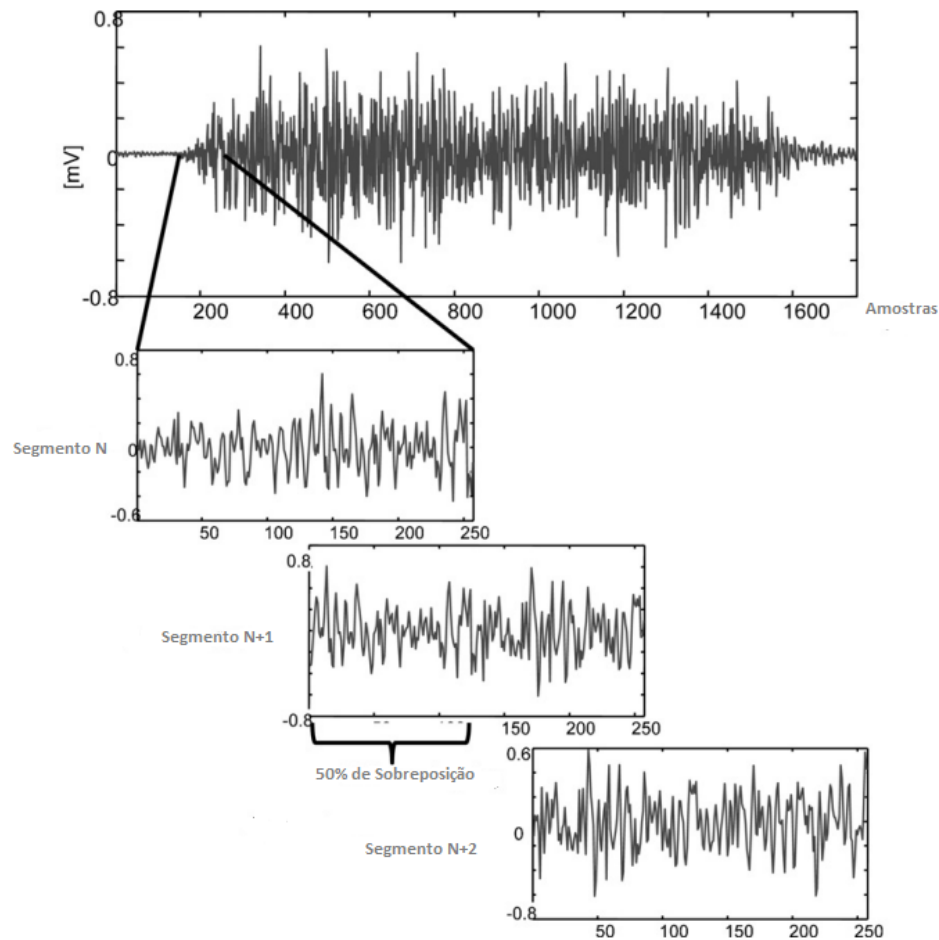
2.3.2 Janelamento do Sinal

Devido a natureza aleatória do sinal EMG, os valores instantâneos adquiridos na etapa de aquisição do sinal não apresentam informações relevantes para que possam servir de conjuntos de entrada para os algoritmos de classificação. Portanto, os dados de entrada devem ser segmentados de forma que características relevantes sejam extraídas destas janelas e, posteriormente, alimentadas em um algoritmo de classificação (SMITH *et al.*, 2011). Quanto maior o tamanho da janela considerada, menor será a variância desta, o que implicará em uma maior acurácia na classificação. Entretanto, a etapa seguinte, a extração de características, será mais demorada. O janelamento pode ser feita com ou sem sobreposição entre as janelas, a depender da aplicação. A sobreposição entre as janelas permite que nenhum dado seja perdido. Em sistemas online, ou seja, em sistemas de tempo real, a sobreposição pode ser custosa computacionalmente e gerar atrasos, a depender do tamanho da sobreposição (SAMUEL *et al.*, 2017). Um exemplo de janelamento é mostrado na Figura 3 - exemplo de janelamento. $W_1, W_2 \dots W_n$ são as janelas e T são as sobreposições entre as janelas.

2.4 Extração de Características

Como dito anteriormente, segmenta-se o sinal de EMG obtido na fase de aquisição, pois seus valores instantâneos de forma isolada não possuem relevância para a etapa de classificação, dada à natureza randômica do sinal. Além disso, o número de pontos de entrada no classificador seria um gargalo para a diferenciação dos movimentos. Desta forma, é coerente mapear todos os pontos do sinal EMG em um vetor de dimensão menor, chamado de vetor de características. O vetor de características consegue melhor definir os padrões contidos no sinal original, porém de forma mais compacta (SIJIANG; VUSKOVIC, 2004). Três são as abordagens para se realizar a extração de características de sinais EMG: no domínio temporal, da frequência, e tempo-frequência. A última está fora de escopo deste trabalho. Phinyomark, Limsakul e Phukpattaranont (2011a)) apresentam uma série de técnicas de extração de características temporais e no domínio da frequência utilizadas pela literatura até a data de sua publicação. Al-

Figura 3 – Exemplo de janelamento de um sinal EMG



Fonte: Adaptado de Orosco, Lopez e Sciascio (2013).

gumas das características apresentadas por esse estudo serão implementadas nesse trabalho e, portanto, serão definidas a seguir.

2.4.1 Técnicas de extração de características temporais

As técnicas de extração de características temporais são amplamente utilizadas pois, como usam os pontos do sinal de EMG sem realizar nenhuma transformação, são facilmente implementáveis e, ainda, apresentam custo computacional relativamente baixo (OSKOEI; HU, 2006).

2.4.1.1 IEMG Integrado (IEMG, do inglês *Integrated EMG*)

O EMG Integrado pode ser definido como a soma de todos os valores de um segmento do sinal EMG:

$$IEMG_i = \sum_{k=1}^N |x_k|, \quad i = 1, \dots, I \quad (2)$$

Onde N é o número total de pontos (amostras), k_i cada amostra e I o número de segmentos do sinal.

2.4.1.2 Valor Absoluto Médio (MAV, do inglês *Mean Absolute Value*)

O valor absoluto médio apresenta semelhança com o IEMG, porém é dividido pelo número de pontos do segmento:

$$MAV_i = \frac{1}{N} \sum_{k=1}^N |x_k|, \quad i = 1, \dots, I \quad (3)$$

2.4.1.3 Valor Absoluto Médio Modificado - Tipo 1

O valor Absoluto Médio Modificado Tipo 1 é uma característica derivada da MAV. Nessa característica, o sinal no segmento é multiplicado por uma janela w_k :

$$MAV1_i = \frac{1}{N} \sum_{k=1}^N w_k |x_k| \quad i = 1, \dots, I \quad (4)$$

onde w_k é definido por:

$$\begin{cases} 1, & \text{para } 0,25N \leq i \leq 0,75N \\ 0,5, & \text{caso contrário} \end{cases} \quad (5)$$

2.4.1.4 Valor Absoluto Médio Modificado - Tipo 2

O valor Absoluto Médio Modificado Tipo 2 apresenta uma modificação na função de janelamento w_k , que é mais suave:

$$MAV2_i = \frac{1}{N} \sum_{k=1}^N w_k |x_k| \quad i = 1, \dots, I \quad (6)$$

onde w_k é definido por:

$$\begin{cases} 1, & \text{para } 0,25N \leq i \leq 0,75N \\ \frac{4i}{N}, & i < 0,25N \\ 0,5, & \text{caso contrário} \end{cases} \quad (7)$$

2.4.1.5 Quadrado Integral Simples (SSI, do inglês *Simple Square Integral*)

O SSI é uma característica que usa a energia do sinal no segmento. A característica é obtida elevando-se cada elemento do segmento ao quadrado e, depois, as somando:

$$SSI_i = \sum_{k=1}^N x_k^2, \quad i = 1, \dots, I \quad (8)$$

2.4.1.6 Variância do EMG

Como a média de um sinal EMG é praticamente 0, pode-se definir a variância de um sinal em um segmento como sendo:

$$SSI_i = \frac{1}{N-1} \sum_{k=1}^N x_k^2, \quad i = 1, \dots, I \quad (9)$$

2.4.1.7 Valores Absolutos do terceiro, quarto e quinto momentos temporais

Propostas para o controle de uma prótese de um braço, os valores absolutos de terceiro, quarto e quinto momento (do inglês *Absolute value of the 3rd, 4th and 5th temporal moment*) são definidos por:

$$TM3_i = \left| \frac{1}{N} \sum_{k=1}^N x_k^3 \right|, \quad i = 1, \dots, I \quad (10)$$

$$TM4_i = \frac{1}{N} \sum_{k=1}^N x_k^4, \quad i = 1, \dots, I \quad (11)$$

$$TM5_i = \left| \frac{1}{N} \sum_{k=1}^N x_k^5 \right|, \quad i = 1, \dots, I \quad (12)$$

2.4.1.8 Valor Quadrático Médio

O valor quadrático médio, RMS (do inglês *Root Mean Square*), uma característica muito utilizada na literatura em sinais EMG, é definida por:

$$RMS_i = \sqrt{\frac{1}{N} \sum_{k=1}^N x_k^2} \quad i = 1, \dots, I \quad (13)$$

2.4.1.9 Detector Logarítmico

Essa característica fornece uma estimativa da força de contração e é definida por:

$$LOG_i = e^{\frac{1}{N} \sum_{k=1}^N \log(|x_k|)} \quad i = 1, \dots, I \quad (14)$$

2.4.1.10 Tamanho da forma de Onda

O tamanho da forma de onda, ou WL (do inglês *waveform length*), é definido como o tamanho cumulativo da forma de onda no segmento de sinal considerado:

$$WL_i = \sum_{k=1}^{N-1} |x_{k+1} - x_k| \quad i = 1, \dots, I \quad (15)$$

2.4.1.11 Mudança Média de Amplitude

A mudança média de amplitude, ou AAC (do inglês *Average Amplitude Change*) se diferencia da WL apenas pela divisão pelo número de amostras considerado no segmento:

$$AAC_i = \frac{1}{N} \sum_{k=1}^{N-1} |x_{k+1} - x_k| \quad i = 1, \dots, I \quad (16)$$

2.4.1.12 Valor do Desvio Padrão da Diferença Absoluta

A mudança média de amplitude, ou AAC (do inglês *Average Amplitude Change*) se diferencia da WL apenas pela divisão pelo número de amostras considerado no segmento:

$$DASDV_i = \sqrt{\frac{1}{N-1} \sum_{k=1}^{N-1} (x_{k+1} - x_k)^2} \quad i = 1, \dots, I \quad (17)$$

2.4.2 Técnicas de Extração no Domínio da Frequência

As características no domínio da frequência também são extensivamente utilizadas pela literatura para a extração de características. A maioria delas são definidas a partir da densidade espectral de potência do sinal, a PSD (do inglês *Power Spectral Density*).

2.4.2.1 Frequência Média

Define-se a Frequência Média, MNF (do inglês *Mean Frequency*) como a soma do produto da PSD com a frequência dividido pela soma da PSD:

$$MNF_i = \frac{\sum_{j=1}^M f_{ij} P_{ij}}{\sum_{j=1}^M P_{ij}} \quad i = 1, \dots, I \quad (18)$$

Onde f_{ij} é a frequência do espectro em um ponto de frequência (bin) j , P_{ij} é a densidade PSD do sinal em um bin j para o i -ésimo segmento, M é o tamanho da janela de frequência e I a quantidade de janelas em frequência do sinal EMG.

2.4.2.2 Frequência de Pico

A frequência de pico, PKF (do inglês *Peak Frequency*) é a frequência cuja PSD é máxima, no segmento:

$$PKF_i = f_i(\max(P_{ij})), \quad i = 1, \dots, I; \quad (19)$$

2.4.2.3 Frequência Mediana

A frequência mediana, MDF (do inglês *Median Frequency*) é a frequência que divide o espectro do segmento considerado ao meio, onde cada metade possui amplitude igual a outra. Pode ser definida como:

$$MDF_i = \frac{1}{2} \sum_{j=1}^M P_{ij}, \quad i = 1, \dots, I; \quad (20)$$

2.4.2.4 Energia Média

A energia média, MNP (do inglês *Mean Power*), é uma característica que pode ser definida como:

$$MNP_i = \frac{\sum_{j=1}^M P_{ij}}{M}, \quad i = 1, \dots, I; \quad (21)$$

2.4.2.5 Zero, primeiro, segundo e terceiro momentos espectrais

Phinyomark, Limsakul e Phukpattaranont (2011b) definem os momentos espectrais como sendo uma alternativa estatística para extrair características do espectro EMG:

$$SM0_i = \sum_{j=1}^M P_{ij}, \quad i = 1, \dots, I; \quad (22)$$

$$SM1_i = \sum_{j=1}^M P_{ij} f_{ij}, \quad i = 1, \dots, I; \quad (23)$$

$$SM2_i = \sum_{j=1}^M P_{ij} f_{ij}^2, i = 1, \dots, I; \quad (24)$$

$$SM3_i = \sum_{j=1}^M P_{ij} f_{ij}^3, i = 1, \dots, I; \quad (25)$$

2.4.2.6 Variância da Frequência Central

A variância da frequência central, VCF (do inglês *Variance of Central Frequency*) é uma importante característica da PSD e pode ser definida como:

$$VCF_i = \left(\frac{SM1}{SM2} \right)^2, i = 1, \dots, I; \quad (26)$$

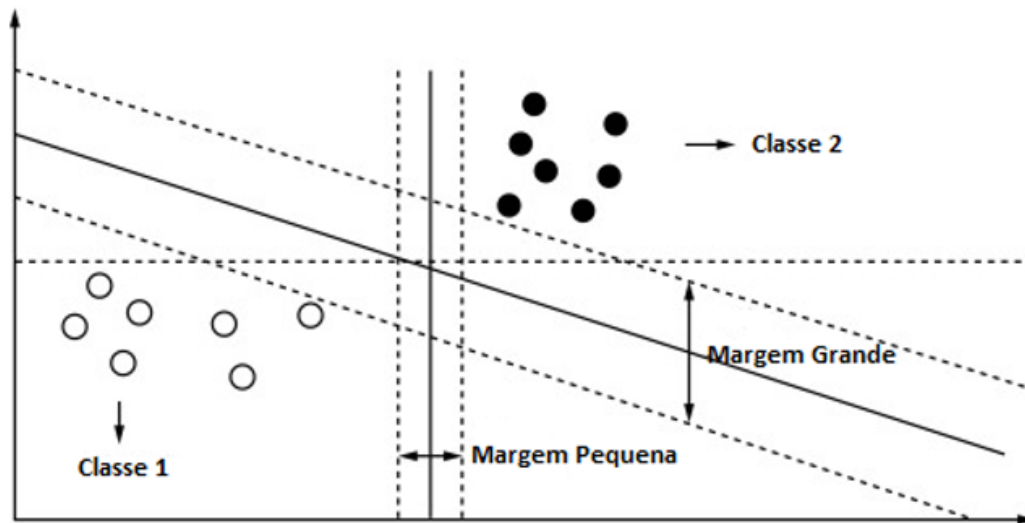
2.5 Classificação

A função do classificador é utilizar o vetor de características resultante da etapa de extração de características, aprender a generalizar a partir desse conjunto de dados e, então, ser capaz de associar classes à novos conjuntos de dados de entrada (DUDA; HART; STORK, 2000). No contexto de sinais EMG, o classificador deve ser capaz de identificar qual movimento específico o usuário está realizando a partir dos dados musculares adquiridos, dentro um universo de vários movimentos possíveis. No contexto de um amputado com uma prótese, o classificador identificaria o sinal EMG adquirido que contém o movimento intencional, classificaria este em alguma classe e, então, o atuador (prótese) seria acionado (SUDARSAN; SEKARAN, 2012). Vários são os classificadores utilizados pela literatura para classificação de sinais EMG, como Redes Neurais Artificiais, algoritmo de Naive Bayes, Análise Discriminante Linear, K-vizinhos mais próximos, e máquina de vetores de suporte (SVM) (NAZMI *et al.*, 2016). Neste trabalho será utilizado a SVM (do inglês *Support Vector Machine*).

2.5.1 Algoritmos de Classificação

A SVM pode ser definida como um classificador que gera hiperplanos que deve separar diferentes classes, de modo que o espaçamento entre as classes sejam o maior possível. Em outras palavras, fornecendo dados previamente classificados na etapa de treinamento, o SVM fornece um hiperplano ótimo que permite classificar novos dados provindos do conjunto de teste (KO *et al.*, 2014). Caso os dados de entrada não sejam separáveis em sua dimensão original, esses são mapeadas em dimensões maiores por meio de funções denominadas kernels, arranjando os dados de forma que se tornem separáveis e, conseqüentemente, classificáveis (HOSSAIN; MIAH, 2015).

Figura 4 – Conceito Geral de uma SVM



Fonte: Autoria Própria (2022).

2.5.2 Métricas de Avaliação em Machine Learning

A performance do vetor de características que serve de entrada para o modelo de classificação é avaliada a partir de métricas de desempenho chamadas acurácia, precisão e recall. Essas três métricas são derivadas da matriz de confusão, que é uma tabela que relaciona a quantidade de valores classificados, sendo eles verdadeiros positivos (TP), falso positivos (FP), Falso negativos (FN) e verdadeiros negativos (TN). A matriz de confusão, terá dimensão $N \times N$ para uma classificação com N classes diferentes (JEPPESEN *et al.*, 2019). Abaixo, na figura 5, é apresentada uma matriz de confusão para uma classificação binária (2x2):

Figura 5 – Exemplo de uma matriz de confusão para classificação binária

		Valores Verdadeiros	
		+	-
Valores Preditos	+	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	-	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Fonte: Autoria Própria (2022).

A partir da matriz de confusão, define-se as seguintes métricas: Acurácia: é uma medida que indica o desempenho geral do classificador, ou seja, o total de acertos sobre o número de

acertos e erros. É definida por:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (27)$$

Precisão: é uma medida que indica quantas classes foram classificadas corretamente como positivas em relação ao número total de classes preditas como positivas (corretamente ou incorretamente):

$$Precisão = \frac{VP}{VP + FP} \quad (28)$$

Recall: Diferente da precisão, a métrica chamada recall considera as classificações positivas perdidas, ou seja, os falsos negativos:

$$Recall = \frac{VP}{VP + FN} \quad (29)$$

F1 Score: o F1 Score pode ser interpretado como a média harmônica da precisão e do recall e é útil quando se há um desbalanceamento de classes no conjunto de dados de entrada do classificador. É definido como:

$$F1 = 2 \frac{Precisão * Recall}{Precisão + Recall} \quad (30)$$

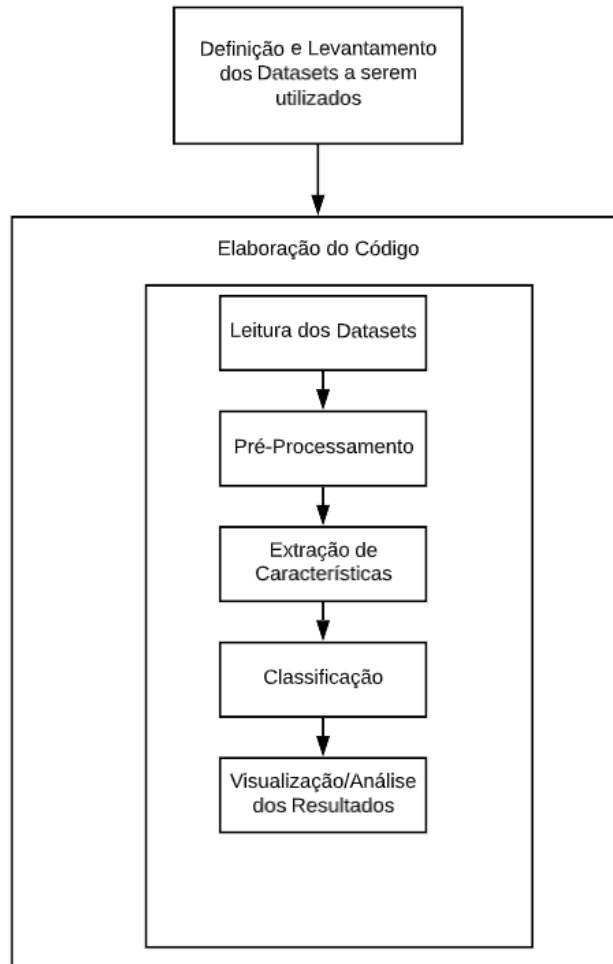
3 MATERIAIS E MÉTODOS

Com o objetivo de apoiar e justificar os resultados que serão mostrados posteriormente, neste capítulo apresentamos a descrição dos conjuntos de dados utilizados, os algoritmos e critérios utilizados na análise, baseados na literatura e experimentação.

3.1 Fluxograma Geral do Processo de Análise

O diagrama abaixo mostra um resumo das atividades executadas para alcançar os objetivos deste trabalho:

Figura 6 – Fluxograma geral do trabalho



Fonte: Autoria Própria (2022)..

Neste estudo, foram 4 os *Datasets* escolhidos, que foram processados através de métodos implementados em `python` através de `jupyter notebooks`, um ambiente que permite explorar conjunto de dados, gerar gráficos e realizar análises quantitativas e qualitativas de forma iterativa. O `jupyter notebook` é *open source* e, por isso, não é necessário uma licença para utilizá-lo. Para a implementação dos códigos, as principais ferramentas utilizadas são:

- `python`: O projeto unifica um conjunto de bibliotecas escritas em `python` com o objetivo de analisar biosinais, como dados provindos de eletroencefalografia, magnetoencefalografia, eletromiografia, entre outros.
- A biblioteca `Scikit Learn` para rotinas baseadas em Inteligência Computacional, também escrita na linguagem de programação `python`.
- A biblioteca `SciPy` que unifica diversas funções científicas disponíveis em `Python`, tais como análises estatísticas, construção de gráficos, etc.

A utilização dessas três ferramentas em conjunto permitiram a execução de todas as etapas de programação, desde a aquisição dos dados brutos dos *datasets* até o processamento de dados.

3.2 Descrição dos *Datasets* Utilizados

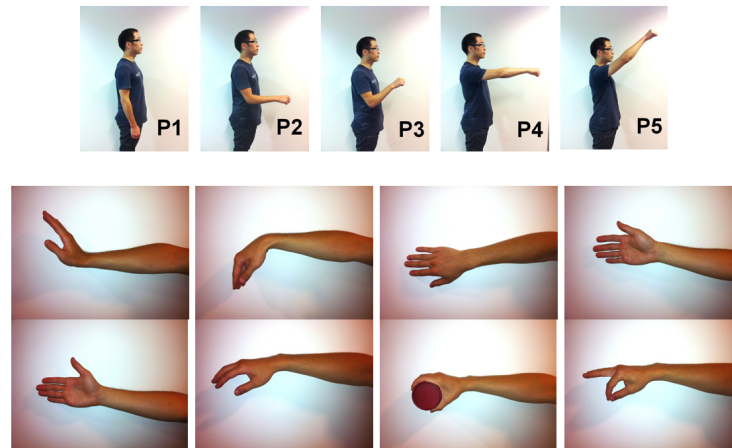
Uma vez que a problemática do trabalho estava bem definida, o próximo passo foi encontrar na internet bases de dados abertas que pudessem ser processadas. Além disso, cada base de dados devia conter movimentos que são considerados úteis no dia-a-dia, ou seja, movimentos que seriam comumente feitos por uma pessoa utilizando um braço prostético com controle a partir de um sistema baseado em eletromiografia. Também foi utilizada uma base de dados EMG gravada nas dependências da UTFPR Campo Mourão, cedida pelo professor Rodrigo Hübner, do Departamento Acadêmico de Ciência da Computação (DACOM).

3.2.1 *Dataset 1: Towards Limb Position Invariant Myoelectric Pattern Recognition Using time-dependent spectral features*

Khushaba *et al.* (2014) apresenta a relação de algumas rotinas de extração de características com a localização do membro (neste caso, mão) quando um determinado movimento é iniciado. Neste estudo, nove indivíduos, sete homens e duas mulheres, com idades entre 20 e 37 anos foram recrutados para performar 8 classes de movimentos. Nenhum dos participantes apresentava desordens neurológicas ou motoras. O *dataset* foi gravado usando 7 canais EMG dispostos sobre a circunferência do antebraço e processados através de um programa desktop chamado Bagnoli (Delsys Inc.). Durante o experimento, foram realizados um grupo de 8 movimentos com as mãos. Cada grupo de movimentos contendo os 8 movimentos foram realizados em 5 posições diferentes do braço, como mostrados na figura 7:

Os movimentos realizados foram: 1) flexão de Pulso, 2) extensão do Pulso, 3) pronação, 4) supinação, 5) punho fechado, 6) movimento de pinça, 7) Mão aberta, 8) descanso. Os sinais EMG coletados foram amplificados usando um Delsys Bagnoli-8 com um ganho de 1000. Um conversor analógico-digital foi usado para amostrar o sinal com uma frequência de amostragem de 4000 Hz.

Figura 7 – Movimentos realizados e respectivas posições dos membros (P1 até P5)



Fonte: Khushaba *et al.* (2014).

Os participantes realizaram os movimento durante 5 segundos, repetindo-os 5 vezes (daqui pra frente cada repetição será chamada de *trial*), para as 5 posições definidas. Neste trabalho, a primeira posição foi desconsiderada do trabalho devido dados temporais insuficientes no *dataset*. Além disso, os dados de todas as tentativas para cada posição foram concatenadas.

3.2.2 *Dataset 2: EMG based classification of basic hand movements based on time-frequency features*

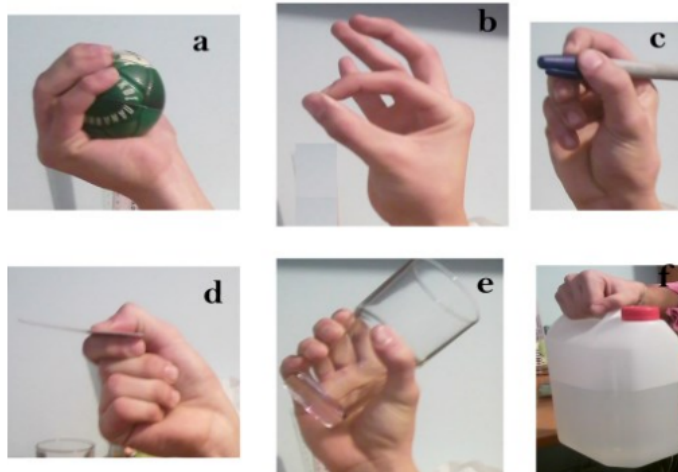
Sapsanis, Georgoulas e Tzes (2013) apresentam em seu estudo uma combinação de algoritmos da etapa de processamento para classificar dados de movimentos diários a partir da aquisição de sinais EMG. Neste estudo, seis indivíduos saudáveis (dois homens e quatro mulheres) com idades variando entre 20 e 22 foram orientados a realizar os seguintes movimentos: a) esférico, b) pinça, c) palmar, d) lateral, e) cilíndrico e f) gancho. O *hardware* de aquisição utilizado foi um conversor analógico-digital N1 USB-6009, com o sinal sendo coletado a uma frequência de amostragem de 500 Hz através de dois eletrodos de superfície sendo transmitidos para um dispositivo de coleta de sinais da Delsys (2 canais). Os movimentos que foram realizados são mostrados abaixo na figura 8:

Os participantes executaram cada movimento por 6 segundos, repetindo-os 30 vezes. Logo, para cada indivíduo, foram adquiridos 180 sinais de 6 segundos, em dois canais.

3.2.3 *Dataset 3: Dataset for multi-channel surface electromyography (sEMG) signals of hand gestures*

Ozdemir *et al.* (2022) também apresentam, em seu *dataset* disponibilizado, movimentos que são utilizados no cotidiano. Neste *dataset*, 40 participantes com idades entre 18 e 29 anos foram convidados para realizar 10 movimentos das mãos: 1) Descanso, 2) Extensão, 3) Flexão,

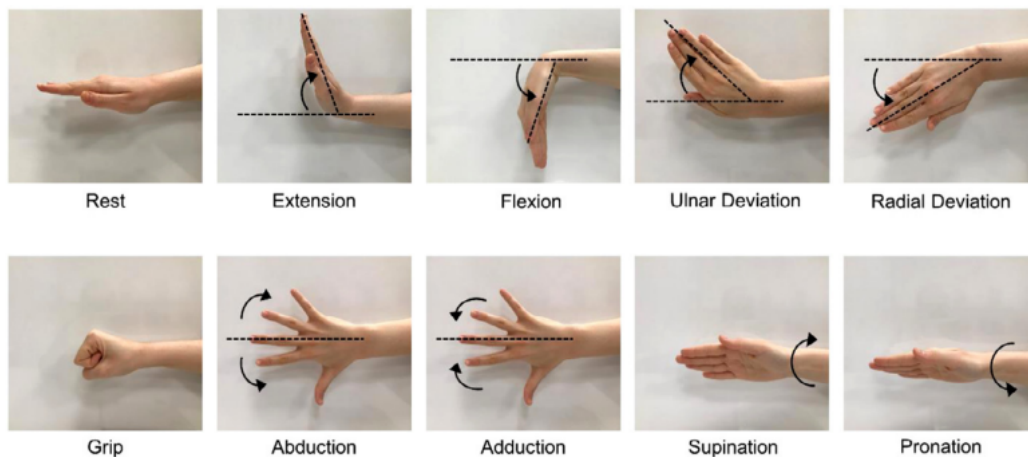
Figura 8 – 6 Movimentos realizados no dataset citado



Fonte: Sapsanis, Georgoulas e Tzes (2013).

4) Desvio Ulnar, 5) Desvio Radial, 6) Punho fechado, 7) Abdução, 8) Adução, 9) Supinação e 10) pronação. Os movimentos são mostrados na figura 9:

Figura 9 – Movimentos do dataset

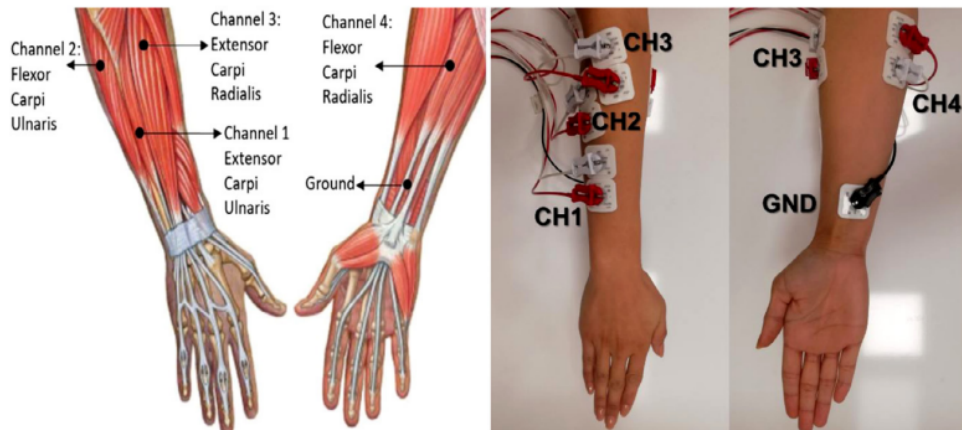


Fonte: Ozdemir *et al.* (2022).

Os dados foram coletados a partir de eletrodos bipolares de Ag/AgCl que foram dispostos sobre 4 músculos do antebraço através do dispositivo de 4 canais BIOPAC MP36 cuja frequência de amostragem é de 2000 Hz. A figura 10 mostra a localização exata dos eletrodos:

Toda a coleta de dados para cada participante teve uma duração de 640 segundos, que incluíam 5 *trials* de 104 segundos. Cada *trial* era separado do outro por um tempo de descanso de 30 segundos. Dentro de cada *trial*, o indivíduo realizava cada um dos 10 movimentos por 6 segundos, onde cada movimento era seguido por um tempo de descanso de 4 segundos.

Figura 10 – Disposição dos eletrodos no antebraço



Fonte: Ozdemir *et al.* (2022).

3.2.4 Dataset 4: Dataset disponibilizado Prof. Rodrigo Hübner (Ciência da Computação)

A base de dados pública disponibilizada pelo professor Rodrigo Hübner do departamento da ciência da computação da UTFPR Campus Campo Mourão foi gravada com dez indivíduos¹, onde desempenharam movimentos úteis ao cotidiano. Cada indivíduo realizou o seguinte conjunto de movimentos: 1) Flexão de Punho (Fechar a Mão), 2) Extensão de Punho (Abrir a mão), 3) Pinça (segurar objeto pequeno), 4) Segurar Cartão, 5) Segurar Caneta, 6) Segurar objeto esférico, 7) Segurar copo e 8) Gancho. O *dataset* apresenta arranjo de movimentos muito parecido com o *dataset* anterior, com o acréscimo dos movimentos 1 e 2.

Foram posicionados 8 eletrodos referentes a 4 canais mais 1 de referência combinada nas seguintes regiões do membro superior:

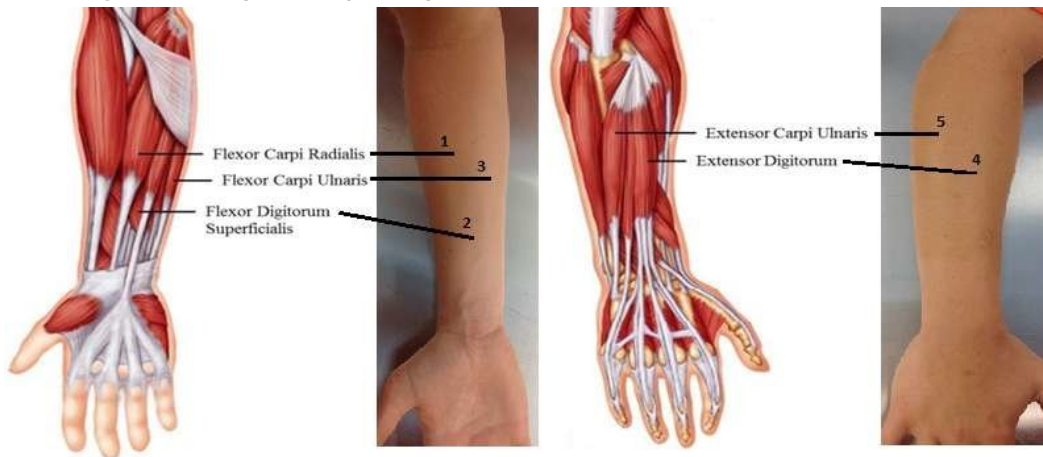
- Canal 1 - músculo extensor dos dedos;
- Canal 2 - músculo extensor ulnar do carpo;
- Canal 3 - Músculo flexor superficial dos dedos;
- Canal 4 - Músculo flexor Radial do carpo
- Referência - Cotovelo.

A figura 11 mostra um esquemático da disposição dos eletrodos enquanto a 12 mostra a disposição dos mesmos durante a execução experimental:

O hardware utilizado para a aquisição dos sinais foi um OpenBCI Ganglion, que possui um conversor analógico-digital com frequência de amostragem de 200 Hz, 4 canais de alta impedância e comunicação *bluetooth Low Energy*. Cada movimento tem uma duração de 8 segundos, sendo que o movimento permanecia até o término do tempo. Entre cada movimento,

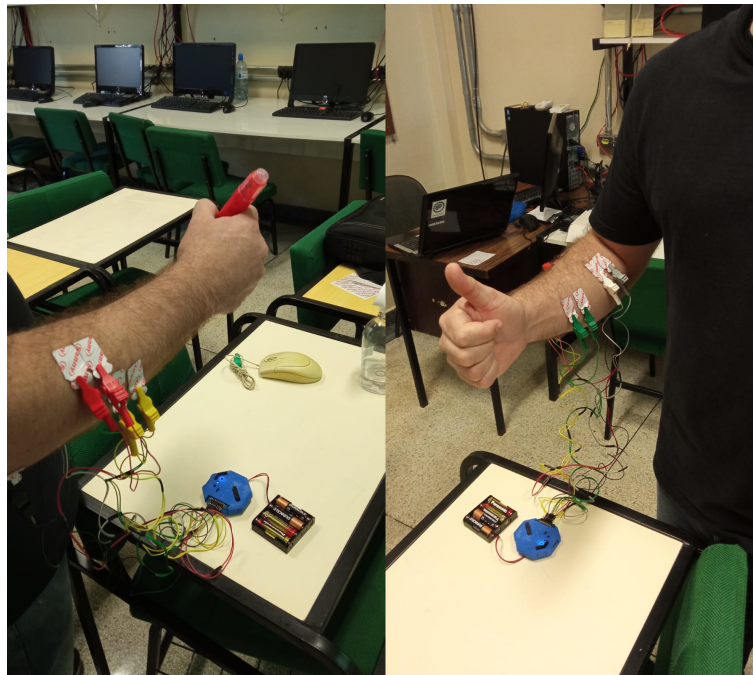
¹ A análise pelo CEP será finalizada à tempo da publicação do trabalho em uma revista e/ou congresso.

Figura 11 – Representação esquemática da musculatura e eletrodos associados.



Fonte: Cedido pelo professor Rodrigo Hübner. Acervo Pessoal.

Figura 12 – Disposição dos eletrodos durante a execução dos experimentos.



Fonte: Cedido pelo professor Rodrigo Hübner. Acervo Pessoal.

o indivíduo descansava por 10 segundos. A sequência de movimentos e descanso foi repetida 3 vezes para cada participante, com um intervalo de aproximadamente 2 minutos entre cada sessão. Foi utilizado o software OpenBCI GUI para encaminhar os dados recebidos pela placa para um script de organização e rotulação dos dados usando um protocolo de comunicação LSL (camada de streaming de laboratório, do inglês *Lab Streaming Layer*).

3.3 Leitura dos *Datasets*

Após a definição dos datasets, foi necessário importá-los no ambiente jupyter notebook para que o processamento de sinais fosse realizado. Como cada dataset estava estruturado de

forma diferente, diferentes foram as rotinas implementadas em `python` para a leitura de cada um. O *dataset* 1 foi lido a partir de arquivos de texto disponibilizados (.txt), os *datasets* 2 e 4 foram lidos a partir de arquivos MATLAB (.MAT) e, por fim, o *dataset* 3 foi lido a partir de arquivos numpy (.npy). Todos os *datasets*, ao sair da etapa de leitura, eram comprimidos em um vetor multidimensional numpy (biblioteca adequada para trabalhar com dados multidimensionais em `python`) que possuía a seguinte dimensão:(NúmeroDeClasses x NúmeroDeTrials x NúmeroDeCanais x DadosTemporais). Após a leitura, cada vetor ficou com as seguintes dimensões:

- *Dataset* 1: (8,24,7,2000), que representa os 8 movimentos a serem classificados, 24 trials executados, 7 canais de dados e, por fim, os dados.
- *Dataset* 2: (6,30,2,3000), que representa os 6 movimentos a serem classificados, 30 trials executados, 2 canais de dados e, por fim, os dados.
- *Dataset* 3:(10,5,4,12000), que representa os 10 movimentos a serem classificados, 5 trials executados, 4 canais de dados e, por fim, os dados.
- *Dataset* 4: (8,3,4,1600), que representa os 8 movimentos a serem classificados, 3 trials executados, 4 canais de dados em por fim, os dados.

É importante mencionar que cada carregamento foi feito de modo a pegar os dados de uma pessoa por vez. Para carregar diferentes *datasets* de participantes diferentes, a referência para o *dataset* do respectivo participante foi feita manualmente.

3.4 Pré Processamento de Sinais

3.4.1 Filtragem dos sinais

Após os arranjos dos dados nos vetores multidimensionais, a filtragem dos dados foi realizada através de métodos provindos da biblioteca `scipy`. Foram implementados dois filtros, um filtro butterworth de quarta ordem para atenuar frequências de 50 Hz e/ou 60 Hz (a depender da localidade de gravação do *dataset*) e um passa-faixas butterworth de quarta ordem para limitar a banda desejável dos sinais e, ainda, para a remoção de componentes DCs dos sinais e outros possíveis ruídos contaminadores).O código a seguir mostra os métodos de filtragem implementados.

Listing 3.1 – Métodos de filtragem e plotagem das PSDs

```

1 # definições de filtros
2
3 def butter_bandpass(data, lowcut, highcut, fs=2000, order=4):
4     nyq = fs * 0.5

```

```

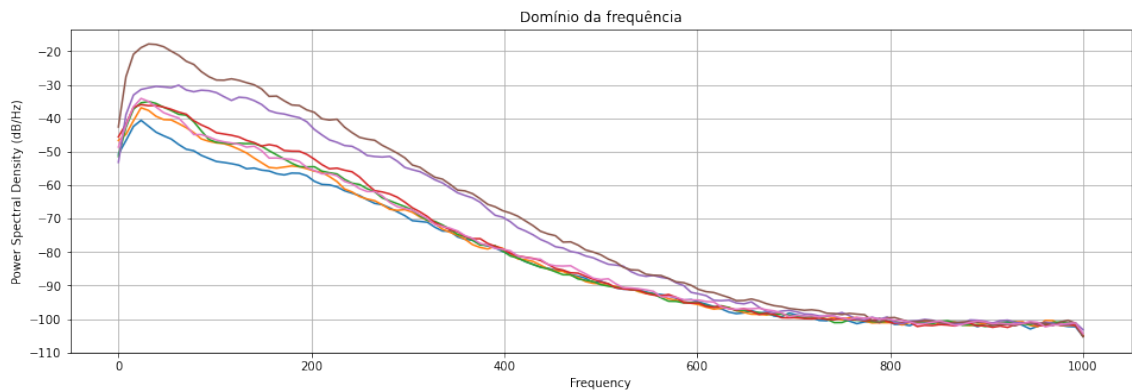
5     low = lowcut / nyq
6     high = highcut / nyq
7     b, a = signal.butter(order, [low, high], btype='bandpass')
8     return signal.filtfilt(b, a, data)
9
10
11 def butter_lowpass(data, lowcut, fs=2000, order=4):
12     nyq = fs * 0.5
13     low = lowcut / nyq
14     b, a = signal.butter(order, low, btype='lowpass')
15     return signal.filtfilt(b, a, data)
16
17
18 def butter_highpass(data, highcut, fs=2000, order=4):
19     nyq = fs * 0.5
20     high = highcut / nyq
21     b, a = signal.butter(order, high, btype='highpass')
22     return signal.filtfilt(b, a, data)
23
24
25 def butter_notch(data, cutoff, var=1, fs=2000, order=4):
26     nyq = fs * 0.5
27     low = (cutoff - var) / nyq
28     high = (cutoff + var) / nyq
29     b, a = signal.iirfilter(order, [low, high], btype='bandstop', ftype="butter")
30     return signal.filtfilt(b, a, data)

```

3.4.1.1 Dataset 1

Os dados do primeiro *Dataset* foram fornecidos já filtrados pelo autor, onde um filtro passa-faixas foi aplicado entre 20 e 450 Hz e, então, aplicou-se um filtro notch em 50 Hz para remover o ruído provindo da rede elétrica local. É possível ver o comportamento da densidade espectral de potência no gráfico abaixo, na figura 13 para os 7 canais do sinal (mantendo-se fixos um movimento e um trial específico):

Figura 13 – Espectro de potência para o primeiro Dataset

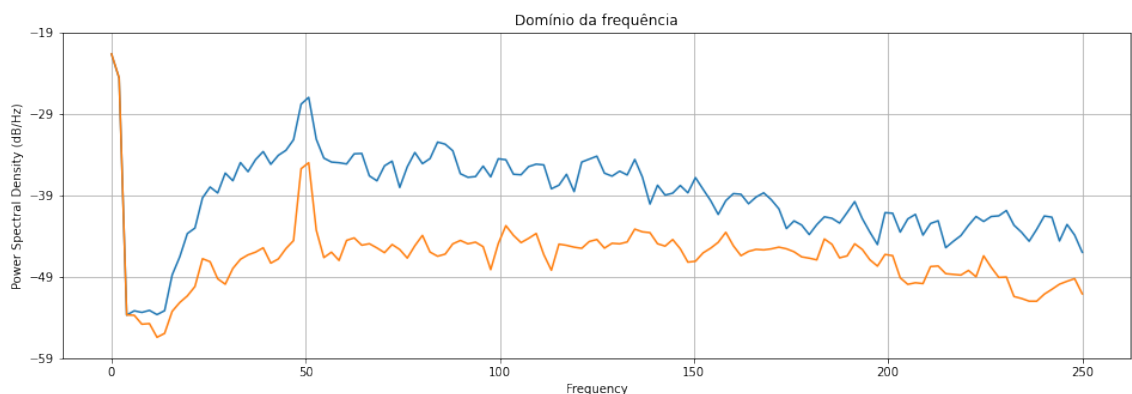


Fonte: Autoria Própria (2022).

3.4.1.2 Dataset 2

Segundo os autores do segundo *Dataset*, os dados foram filtrados com um passa-faixas entre 15 Hz e 500 Hz e, depois, foi aplicado um filtro em 50 Hz para eliminar ruído da rede elétrica local. A densidade espectral de potência para esse sinal é mostrada abaixo, na figura 14:

Figura 14 – Espectro de potência para o segundo Dataset

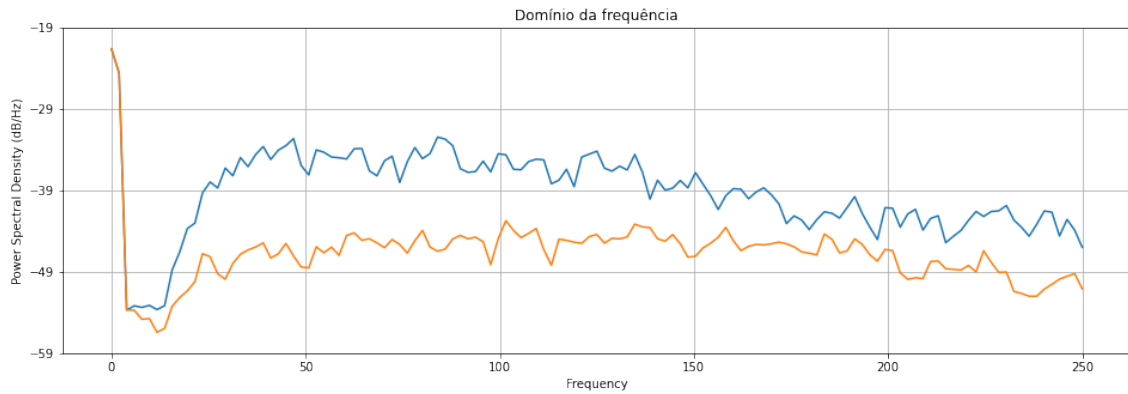


Fonte: Autoria Própria (2022).

Através do gráfico da PSD, é possível notar um pico em 50 Hz, o que significa que a presença por contaminação da rede elétrica ainda é alta. Por isso, um filtro Butterworth de quarta ordem foi aplicado com essa frequência de corte. Após a aplicação do filtro, têm-se a PSD como mostrada na figura 15 a seguir, cuja redução da amplitude na frequência de corte é diminuída:

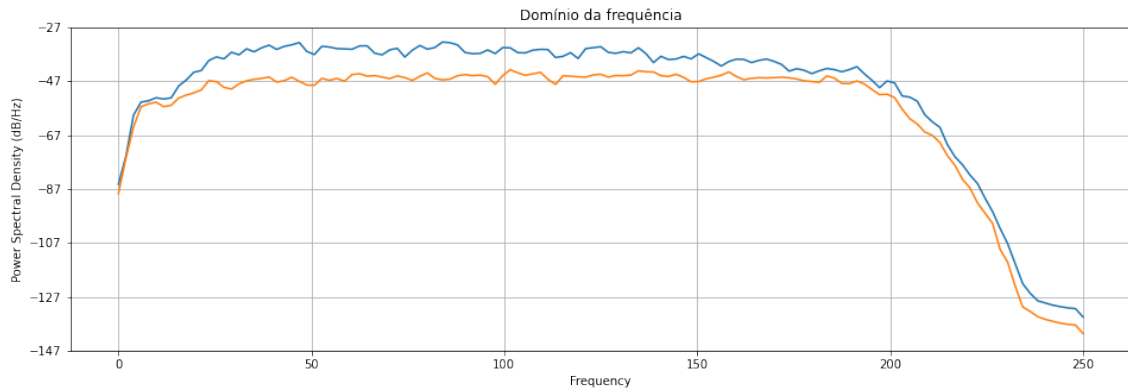
Também foi aplicado um filtro passa-faixas butterworth de quarta ordem para a retirada das componentes de frequência não desejáveis (como componentes DC), cujas frequências de corte entre 5 e 200 Hz, respectivamente. O sinal resultante é mostrado na figura 16 abaixo:

Figura 15 – Espectro de potência para o segundo *Dataset* após filtro notch



Fonte: Autoria Própria (2022).

Figura 16 – Espectro de potência para o segundo *Dataset* após filtro passa-faixas

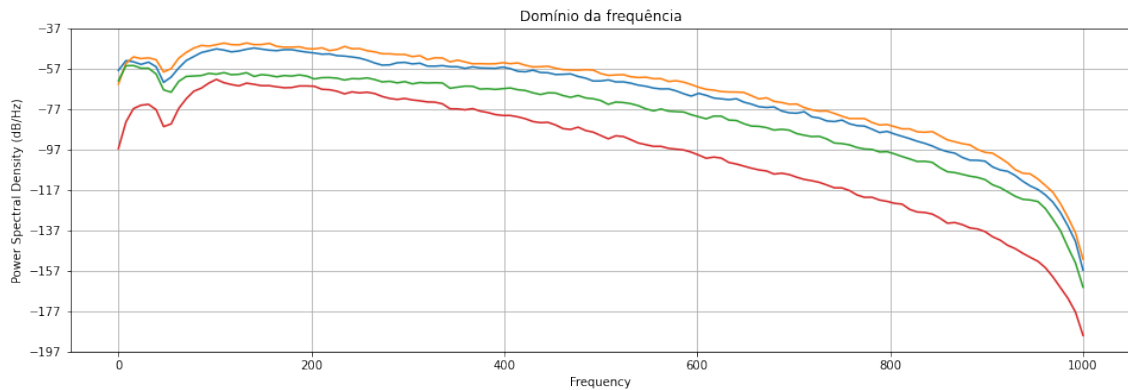


Fonte: Autoria Própria (2022).

3.4.1.3 *Dataset* 3

A terceira base de dados fornecida também já estava com seus dados filtrados. Segundo os autores, foi aplicado um filtro Notch de segunda ordem para remoção de ruídos da rede elétrica, em 50 Hz e um passa-faixas Butterworth de sexta-ordem com o intervalo entre 5 e 500 Hz. A figura 17 mostra a densidade espectral de potência para uma parte do *Dataset*:

Figura 17 – Espectro de potência para o terceiro *Dataset*



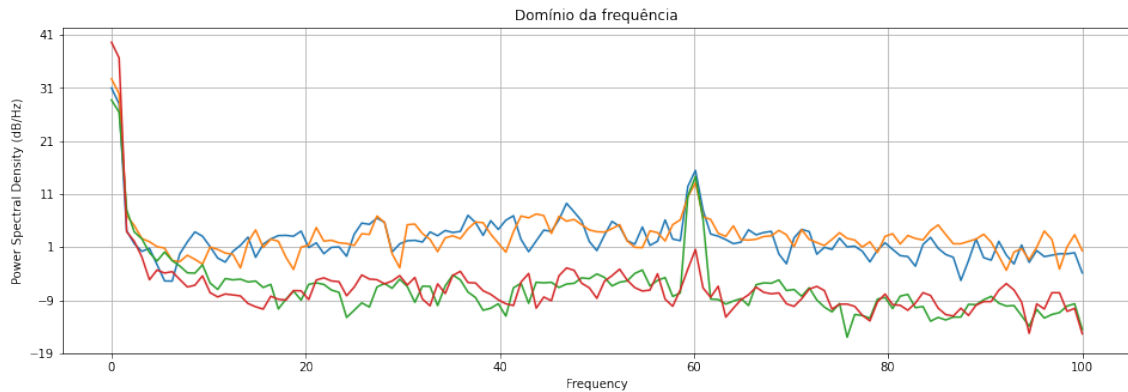
Fonte: Autoria Própria (2022).

Como o sinal já estava adequado para a etapa de janelamento, nenhum filtro foi aplicado.

3.4.1.4 *Dataset 4*

Os dados do quarto *Dataset* foram fornecidos brutos, ou seja, sem aplicação de nenhum filtro. A PSD dos sinais brutos pode ser vista à seguir, a na figura 18:

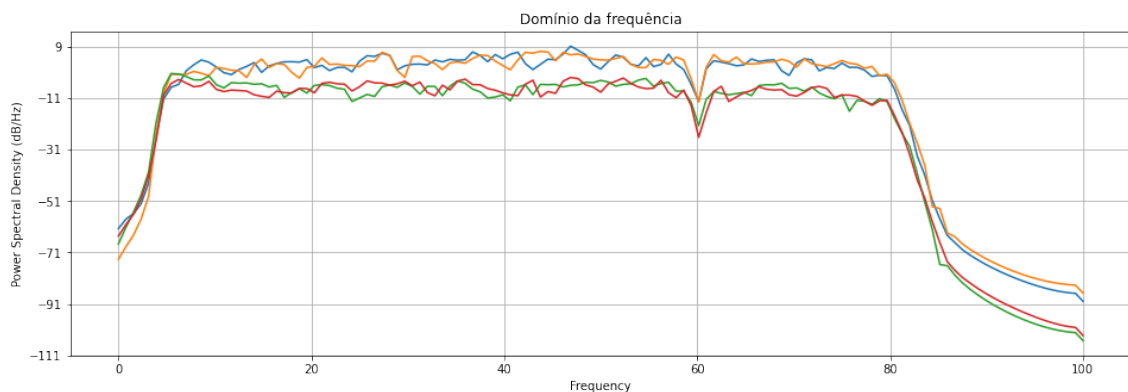
Figura 18 – Espectro de potência para o quarto *Dataset*



Fonte: Aatoria Própria (2022).

É possível notar a presença da forte contaminação pela rede elétrica em 60 Hz, além de um forte pico próximo à componente DC do sinal (0 Hz). Para isso filtragem dos sinais desse *Dataset*, foram aplicados um filtro notch Butterworth de oitava ordem em 60 Hz e um passa-faixas entre 5 e 80 Hz para obtenção da faixa de frequências desejada. O gráfico da PSD após a filtragem dos sinais é mostrado à seguir, na figura 19:

Figura 19 – Espectro de potência para o terceiro *Dataset* após filtragem



Fonte: Aatoria Própria (2022).

3.4.2 Janelamento dos sinais

Após a etapa de filtragem dos sinais, eles precisam ser janelados para que a etapa de extração de características acontecesse. O janelamento dos sinais foi feita no domínio do tempo,

tornando possível a extração de características no domínio do tempo, e, também, para os sinais no domínio da frequência, para servirem de entrada para a etapa de extração das características que se baseiam nos dados da frequência. O código à seguir mostra os métodos de Janelamento implementados, tanto para o domínio do tempo quanto para o domínio da frequência.

Listing 3.2 – Código de Janelamento do Sinal

```

1 from scipy.signal import stft
2
3 step = 1200
4 segment = 8192
5 print(' ', data.shape)
6
7 n_win = int((data.shape[-1] - segment) / step) + 1
8 print(data.shape[-1])
9 print(n_win)
10 ids = np.arange(n_win) * step
11
12
13 # Janelas do dado no dominio do tempo
14 chunks_time = np.array([data[:, :, :, k:(k + segment)] for k in ids]).
15 transpose(1, 2,3,0,4)
16
17 # Janelas do dado no domínio da frequência
18 f, t, chunks_freq = stft(data, fs=4000, nperseg=8192, noverlap=5700)
19 chunks_freq = np.swapaxes(chunks_freq, 3, 4)
20
21 print('Formato (shape) dos dados depois da divisão de janelas')
22 print(f'Domínio do tempo: {chunks_time.shape} - (classes, ensaios, canais,
23 janelas, linhas)')
24 print(f'Domínio da frequência: {chunks_freq.shape} - (classes, ensaios, canais,
25 janelas, linhas)')
```

Como é possível ver no código mostrado, o janelamento no tempo do vetor de dados filtrados foi feita através de um *array slicing*, ou seja, pega-se a última dimensão do vetor e a divide em vários outros subvetores, resultando em uma dimensão maior. Já o janelamento dos dados no domínio da frequência é realizada através de uma Transformada de Fourier de Tempo Curto (STFT, do inglês *Short-Time Fourier Transform*) pois além de fornecer os dados transformados

no domínio da frequência, já os fornece janelados. A STFT foi implementada através do método `stft` da biblioteca **scipy**.

Cada *Dataset* foi dividido de forma que, em uma classificação preliminar experimental com algumas características pré-definidas sua acurácia não variasse muito. Os *Datasets* foram divididos em um número aproximado de janelas (de 9 a 12), com sobreposição no domínio da frequência de 70% para os três primeiros *Datasets* e 90% para o último. Os parâmetros de janelamento temporal eram definidos de forma que o número de janelas no tempo fosse igual ao número de janelas fornecido pela `stft`. Para cada *Dataset*, os seguintes parâmetros foram adotados:

- *Dataset* 1: dividido em 10 janelas, onde cada janela possuía 8192 no domínio do tempo e 4097 no domínio da frequência (5700 pontos de sobreposição na `stft`);
- *Dataset* 2: dividido em 11 janelas, onde cada janela possuía 1024 pontos no domínio do tempo e 513 no domínio da frequência (716 pontos de sobreposição na `stft`);
- *Dataset* 3: dividido em 11 janelas, onde cada janela possuía 4096 pontos no domínio do tempo e 2049 pontos no domínio da frequência (2860 pontos de sobreposição na `stft`);
- *Dataset* 4: dividido em 12 janelas, onde cada janela possuía 512 pontos no domínio do tempo e 257 no domínio da frequência (257 pontos de sobreposição na `stft`).

Se o vetor de dados de antes de entrar na etapa de segmentação tinham o formato (Número de Classes x Número de Trials x Número de Canais x Dados Temporais), ele sairá dessa etapa com o seguinte formato: (Número de Classes x Número de Trials x Número de Canais x Janelas x Dados Temporais) para os dados do domínio do tempo; para a frequência, será gerado um segundo vetor com número de janelas idênticas cujo formato é (Número de Classes x Número de Trials x Número de Canais x Janelas x dados Na frequência).

3.5 Extração de Características

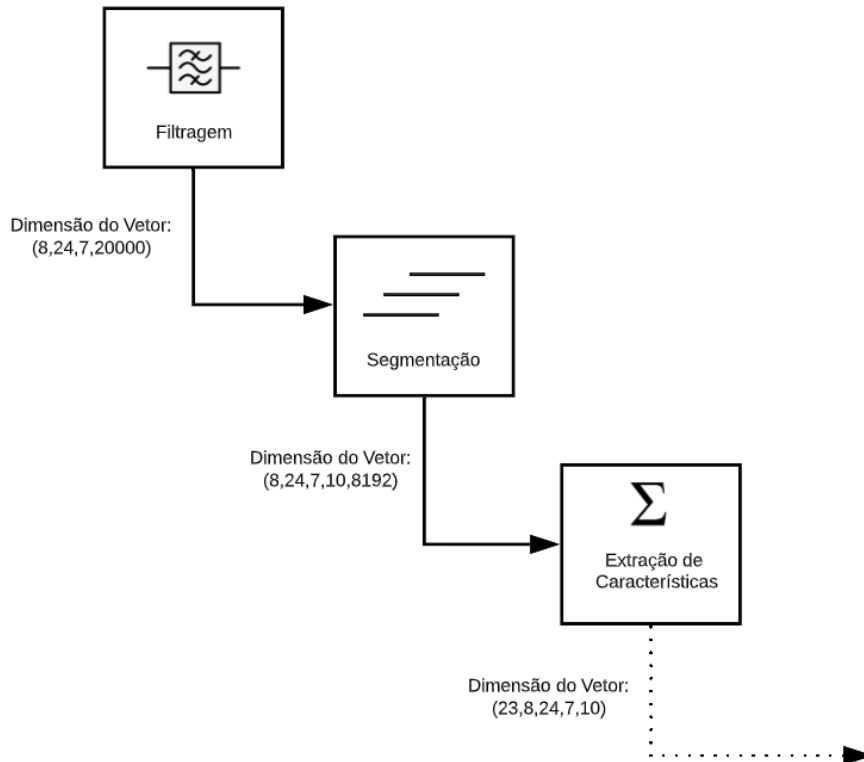
3.5.1 Implementação

Todos os métodos mostrados de extração de características expostos no capítulo 2 foram implementados em `python`, onde cada método recebe o conjunto de dados e o retorna transformado como uma característica. Os métodos implementados são apresentados no apêndice A.

Cada função é responsável por fornecer uma característica para cada janela de dados fornecida, ou seja, a transformação da característica é feita utilizando-se a última dimensão do vetor, com os dados do domínio da frequência e dados do domínio do tempo. Como são 23 as

características extraídas, teremos que pra cada movimento teremos 23 possibilidades diferentes de características. A figura 20 a seguir exemplifica essa transformação dos dados através do *Dataset 1*:

Figura 20 – Dimensões do vetor de dados após as sucessivas etapas



Fonte: Autoria Própria (2022).

3.5.2 Preparação para Classificação

Como o intuito do presente estudo é investigar os impactos de diversas características e suas combinações na classificação dos movimentos, faz sentido gerar um vetor de características contendo todas as possibilidades a serem investigadas. Para exemplificar:

- para 1 característica utilizada, pode-se escolher 1 das 27, 1 por vez, totalizando-se 27 opções (combinações) possíveis;
- para 2 características utilizadas, pode-se combinar 2 características de uma vez em um universo de 27 características.
- para 3 características utilizadas, pode-se combinar três características de uma vez em universo de 27 características, e assim por diante.

A ideia de combinação de características foi implementada por meio do método **combinations**, da biblioteca `itertools` e, pode ser generalizada pela combinação:

$$C_i^j = \frac{i!}{j!(i-j)!} \quad (31)$$

onde i é o número de características disponíveis e j o número de subconjuntos de características possíveis.

Cada combinação possível de vetor de características deve, então, ser atribuída ao classificador. Entretanto, para que a classificação seja realizada, cada vetor de características deve ter suas dimensões reordenadas de modo que apresentem o seguinte formato, mostrado na tabela 1:

Tabela 1 – Formato de dados para classificação

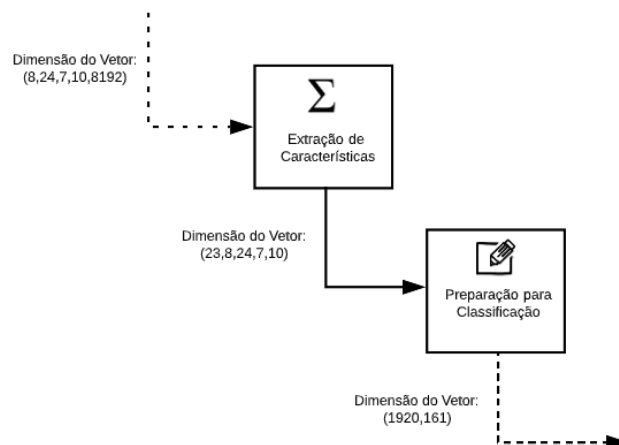
Amostra	C1CH1	C1CH2	C2CH1	C2CH2	CNC H_K	Classe (Movimento)
1	c1	c1	c1	c1	0	0
2	c2	c2	c2	c2	0	1
...
N	cN	cN	cN	cN	cN	Número de Classes

Fonte: Autoria Própria, 2022.

Cada coluna representa uma característica. É importante notar que cada canal de dados multiplica a característica no conjunto de dados, ou seja, o número de colunas do vetor de característica transformado é multiplicado pelo número de canais existentes.

Desta forma, métodos para redimensionamento/achatamento de vetores foram utilizados. A figura 21 abaixo representa o formato de dados após sair dessa etapa de achatamento das dimensões dos dados, para o *Dataset* 1:

Figura 21 – Novo *shape* dos dados.



Fonte: Autoria Própria (2022).

A primeira dimensão do vetor é obtida multiplicando-se as dimensões de movimentos (8), trials (24) e janelas (10). A segunda dimensão do vetor é obtida multiplicando-se as dimensões de características disponíveis (23) pelo número de canais do *Dataset* (7).

O vetor que indica cada classe referente à cada linha de matriz resultante também é adequado de forma que acompanhe o achatamento realizado no vetor de características.

3.6 Classificação dos Sinais

Antes de entrar nos classificadores propriamente ditos, a matriz bidimensional de características obtida é dividida em um conjunto de treino e em um conjunto de teste, onde o conjunto de treino é composto por 70% do conjunto de dados e o conjunto de teste por 30%. A divisão do vetor de dados foi feito utilizando-se o método `train_test_split` da biblioteca `scikit-learn` (python).

A classificação é feita com o auxílio da classe `Pipeline`, que permite a padronização dos dados através do método `StandardScaler` e, já em seguida, na mesma linha, aplicar o classificador. O classificador SVM foi adotado, através de sua versão de implementação conhecida como `SVC`, cujo parâmetro `gamma` foi deixado automático e o parâmetro `C` foi modificado de forma a achar um número de acurácia que mudasse insignificamente após a variação deste. A acurácia do modelo é obtida através da função `score`, também fornecida da biblioteca `scikit-learn` (python).

3.7 Visualização/Análise dos Resultados

Para complementar a análise dos resultados além da acurácia, a matriz de confusão foi implementada através do método `confusion_matrix` e, ainda, um relatório de classificação também foi implementado, contendo as métricas para análises de modelos de machine learning. A função `classification_report` foi utilizada para a obtenção do relatório.

Uma função também foi criada para visualizar uma característica implementada e suas relações com cada movimento do *Dataset*. Dessa forma, foi possível identificar quais características apresentavam melhor distribuição (e, conseqüentemente, seriam mais apropriadas para a classificação). A figura 22 abaixo, apresenta um exemplo de saída dessa função.:

Os *trials* escolhidos para plotar são escolhidos arbitrariamente, assim como os canais escolhidos.

4 RESULTADOS

O primeiro ponto avaliado foi como o classificador performou para cada *dataset*, tanto no tempo quanto no domínio da frequência para todas as combinações de características possíveis. Uma vez que o desempenho geral é conhecido, as melhores características em cada domínio serão analisadas com mais detalhes. É importante notar que, para um dos *datasets*, foram analisados os dados de quatro participantes, de forma a tornar a discussão mais enxuta e objetiva.

4.1 Dataset 1

4.1.1 Visão Geral das características no domínio do tempo

Todas as possibilidades entre as características do domínio foram testadas, para cada participante. Como são 15 as características no tempo, foram obtidas 32767 combinações de características diferentes. A figura 23 abaixo mostra alguns resultados de acurácia para uma pequena amostra de características.

Figura 23 – Parte das amostras de resultados de acurácia - Participante 1

```
In [17]: time_results_df
```

Out[17]:

	Características Consideradas	Acurácia do SVM
0	(var.)	0.934028
1	(rms.)	0.991319
2	(wl.)	0.984375
3	(ssi.)	0.934028
4	(iemg.)	0.993056
...
32762	(var, rms, wl, iemg, mav, mod1, mod2, tm3, tm4...	0.993056
32763	(var, rms, ssi, iemg, mav, mod1, mod2, tm3, tm...	0.993056
32764	(var, wl, ssi, iemg, mav, mod1, mod2, tm3, tm4...	0.993056
32765	(rms, wl, ssi, iemg, mav, mod1, mod2, tm3, tm4...	0.993056
32766	(var, rms, wl, ssi, iemg, mav, mod1, mod2, tm3...	0.993056

Fonte: Autoria Própria, 2022.

Como são milhares de características a serem analisadas por participante, analisar uma a uma seria uma tarefa onerosa. Por isso, alguns dados estatísticos foram retirados do conjunto em geral para, então, uma análise das melhores (e piores) combinações serem analisadas

minuciosamente. A tabela 2 fornece uma visão geral estatística dos resultados de acurácia obtidos para cada participante do *dataset* 1:

Tabela 2 – Resumo dos resultados para as características do domínio do tempo

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	32767	32767	32767	32767
Média	99,17%	99,87%	97,44%	97,58%
Desvio Padrão	0,49%	0,46%	0,96%	0,87%
Mínimo	63,72%	65,10%	40,45%	62,50%
25%	99,13%	99,83%	96,88%	96,88%
50%	99,31%	100,00%	97,57%	97,57%
75%	99,31%	100,00%	98,09%	98,09%
Máximo	100%	100%	99%	100%

Fonte: Autorial Própria, 2022.

Observando-se a tabela 2, é possível notar que uma acurácia muito boa foi obtida ao longo de todas as combinações possíveis, ou seja, o modelo foi capaz de generalizar bem com os dados do conjunto de treino e, logo, classificou bem os resultados do conjunto de teste. Além disso, é importante notar um aumento da acurácia para todos os participantes conforme o número de característica é maior (por meio do aumento da porção do percentil de 25% dos dados para o percentil de 75%).

4.1.2 Visão Geral das Características no domínio da frequência

Todas as possibilidades entre as características no domínio da frequência também foram testadas, para cada participante. Como são 7 as características a serem extraídas no domínio da frequência, foram obtidas 127 combinações diferentes de características. A figura 25 abaixo mostra alguns resultados de acurácia para uma pequena amostra dessas combinações, que, assim como no domínio da frequência, foram compiladas em um dataframe pandas:

Figura 24 – Pequena amostra de resultados de acurácia (frequência) - Participante 1

```
In [27]: freq_results_df
```

```
Out[27]:
```

	Características Consideradas	Acurácia do SVM
0	(median_frequency,)	0.805556
1	(mean_power,)	0.805556
2	(total_power,)	0.805556
3	(pkf,)	0.680556
4	(SM1,)	0.798611
...
122	(median_frequency, mean_power, total_power, SM...	0.807292
123	(median_frequency, mean_power, pkf, SM1, SM2, ...	0.876736
124	(median_frequency, total_power, pkf, SM1, SM2, ...	0.876736
125	(mean_power, total_power, pkf, SM1, SM2, SM3)	0.876736
126	(median_frequency, mean_power, total_power, pk...	0.875000

127 rows × 2 columns

Fonte: Autorial Própria, 2022.

Alguns dados importantes também foram retirados do conjunto de características geral para, então, suportar uma análise das melhores (e piores) combinações serem analisadas minuciosamente. A tabela 3 fornece uma visão geral estatística dos resultados de acurácia obtidos para cada participante do *dataset* 1:

Tabela 3 – Resumo dos resultados para as características do domínio da frequência

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	127	127	127	127
Média	83,67%	87,66%	85,55%	86,27%
Desvio Padrão	3,54%	3,19%	2,79%	2,69%
Mínimo	68,06%	63,72%	57,81%	68,58%
25%	80,56%	85,76%	84,90%	84,72%
50%	81,42%	86,28%	85,42%	85,42%
75%	87,15%	90,28%	86,98%	88,54%
Máximo	87,67%	90,80%	88,02%	89,58%

Fonte: Autoria Própria, 2022.

Apesar dos resultados de acurácia um pouco menores e desvios-padrão um pouco maiores ao se comparado as características no domínio do tempo, para esse *dataset*, os valores foram satisfatórios. Além disso, nota-se o comportamento da acurácia que aumenta conforme mais características são utilizadas no modelo de classificação.

4.1.3 Investigação das melhores características e suas relações com os movimentos

Para os 4 participantes, a pior característica obtida foi a TM5. Já as combinações de características que forneceram a melhor acurácia pra cada conjunto de cada participante é sumarizada na tabela 4:

Tabela 4 – Melhor combinação de características no domínio do tempo - *dataset* 1

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(WL, TM3, vfeature)	100%
Participante 2	(VAR, logdetector)	100%
Participante 3	(RMS, WL, vfeature)	99,31%
Participante 4	(WL, logdetector)	99,83%

Fonte: Autoria Própria, 2022.

Vale ressaltar que o código considera a considera o menor conjunto de características para retornar a acurácia. Se, por exemplo, um conjunto com 5 características possui acurácia de 100% e um segundo conjunto de características com 8 características também apresenta essa acurácia, o primeiro será considerado como a melhor opção. Isso se deve ao fato de que, em aplicações *online* da classificação, quanto menos características utilizadas menor será o vetor de características e, conseqüentemente, menor o tempo para o classificador retornar os resultados, evitando atrasos na atuação de um braço robótico, por exemplo.

De modo semelhante, as melhores combinações de características no domínio da frequência foram como mostrado na tabela 5:

Tabela 5 – Melhor combinação de características no domínio da frequência - *dataset 1*

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(medianFreq, PKF, SM1, SM3)	87,67%
Participante 2	(medianFreq, meanPower, totalPower, PKF, SM2, SM3)	90,80%
Participante 3	(medianFreq, meanPower, totalPower, PKF, SM1, SM2, SM3)	88,02%
Participante 4	(medianFreq, PFK, SM1, SM2, SM3)	89,58%

Fonte: Autoria Própria, 2022.

Mão Aberta: Das características no domínio do tempo que apareceram nos conjuntos de melhores características, WL e `logdetector` foram as características que mais apresentaram clara distinção entre os outros movimentos e, ainda, pouca dispersão entre os dados do mesmo movimento.

Descanso: Das melhores características mencionadas, mais uma vez, WL e `logdetector` apresentaram a melhor performance, para esse movimento.

Punho Fechado: Para esse movimento, dentre os gráficos analisados, a característica `logdetector` foi capaz de fornecer uma divisão clara entre os movimentos, para todos os participantes.

Extensão do Pulso: Para esse movimento, dentre os gráficos analisados, a característica `logdetector` foi capaz de fornecer uma divisão clara entre os movimentos. Entretanto, para dois dos participantes, a TM3, por exemplo, conseguiu comprimir melhor o movimento em uma região do que o `logdetector`.

Flexão de Pulso: A flexão de pulso se mostrou um movimento muito facilmente separável dos outros. Exceto TM3, TM4 e TM5, qualquer característica torna é apropriada para esse movimento.

Movimento de Pinça: Todas as características que apareceram no conjunto das melhores características se mostraram adequadas para o movimento, dentre os participantes analisados.

Pronação: A pronação, assim como a flexão de pulso, se mostrou um movimento muito facilmente separável dos outros. Exceto TM3, TM4 e TM5, qualquer característica é apropriada para esse movimento.

Supinação: Esse movimento apresenta uma distribuição um pouco mais complexa (variando entre os participantes). Entretanto, para todos, a característica `logdetector` foi a que mais conseguiu fornecer separabilidade para o movimento.

No apêndice B desse trabalho são apresentados os relatórios de classificação para as características, que corroboram quantitativamente para a análise exposta.

4.2 Dataset 2

4.2.1 Visão Geral das características no domínio do tempo

Para o *dataset 2*, para o domínio do tempo, a acurácia se distribui ao longo dos dados conforme apresentado na tabela 6:

Tabela 6 – Resumo dos resultados para as características no domínio do tempo - *dataset 2*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	32767	32767	32767	32767
Média	94,64%	94,97%	96,70%	99,05%
Desvio Padrão	2,28%	1,67%	1,99%	1,08%
Mínimo	56,57%	76,60%	49,33%	62,63%
25%	94,44%	94,11%	96,80%	98,99%
50%	95,45%	94,95%	97,31%	99,33%
75%	95,96%	96,46%	97,81%	99,49%
Máximo	97,7%	97,47%	98,65%	99,83%

Fonte: Autoria Própria, 2022.

Analisando-se os dados da tabela acima, é possível ver que a acurácia melhora, conforme mais características são utilizadas, assim como no *dataset 1*. O desvio padrão dos dados também é um pouco maior se comparado ao primeiro.

4.2.2 Visão Geral das Características no domínio da frequência

Para o *dataset 2*, no domínio da frequência, a acurácia se distribui ao longo do *dataset* conforme apresentado na tabela 7:

Tabela 7 – Resumo dos resultados para as características no domínio da frequência - *dataset 2*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	127	127	127	127
Média	80,50%	73,44%	85,74%	86,63%
Desvio Padrão	5,50%	5,71%	5,09%	6,64%
Mínimo	37,21%	26,09%	54,38%	62,12%
25%	79,55%	71,55%	85,10%	84,34%
50%	81,48%	75,08%	86,36%	89,73%
75%	84,18%	76,60%	88,97%	91,58%
Máximo	86,36%	78,79%	91,08%	92,09%

Fonte: Autoria Própria, 2022.

Na tabela é possível ver que o mesmo comportamento visto anteriormente se aplica para esse *dataset*, apesar um pouco da acurácia menor em relação às características do domínio do tempo.

4.2.3 Investigação das melhores características e suas relações com os movimentos

Para os 4 participantes, a pior característica obtida também foi a TM5. Já as combinações de características que forneceram a melhor acurácia pra cada conjunto de cada participante é sumarizada na tabela 8 a seguir:

Tabela 8 – Melhor combinação de características no domínio do tempo - *dataset 2*

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(WL, IEMG, MAV, MODT2, TM3, TM4, logdetector, DASDV)	97,47%
Participante 2	(VAR, RMS, WL, SSI, modt2, TM3, vfeature, AAC)	97,47%
Participante 3	(RMS, WL, IEMG, TM3, TM4, logdetector, AAC, DASDV)	98,65%
Participante 4	(RMS, WL, IEMG, MAV, AAC)	99,83%

Fonte: Autoria Própria, 2022.

De modo semelhante, as melhores combinações de características no domínio da frequência são mostradas na tabela 9:

Tabela 9 – Melhor combinação de características no domínio da frequência - *dataset 2*

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(SM1, SM2)	86,36%
Participante 2	(medianFreq, meanPower, totalPower, SM1, SM3)	78,79%
Participante 3	(medianFreq, SM1, SM2, SM3)	91,08%
Participante 4	(medianFreq, meanPower, SM1, SM2)	92,09%

Fonte: Autoria Própria, 2022.

Com base nas melhores características encontradas, segue-se a análise para cada movimento, de forma análoga feita no primeiro *dataset*:

Cilíndrico: dentre os dados dos quatro participantes, a característica `logdetector` foi a única capaz de separar os dados desse movimento dos outros. Entretanto, as outras características que compõem os grupos das melhores combinações (exceto TM3 e TM4 e TM5) se mostraram adequadas para esse tipo de movimento.

Gancho: Excluindo as características TM3, TM4 e TM5, todas as características presentes nas melhores combinações conseguiram separar esse movimento dos outros com facilidade, inclusive para os dados do participante 2, cuja dispersão dos dados era maior.

Esférico: Assim como no movimento cilíndrico, a característica `logdetector` foi a única capaz de separar os dados desse movimento dos outros nos quatro participantes. Para o participante 1, para todas as outras características esse movimento se confundia com o cilíndrico. Entretanto, o `logdetector` foi capaz de separar esses movimentos corretamente. Para as outras características que compõem os grupos das melhores combinações (exceto TM3 e TM4 e TM5), todas se mostraram adequadas para esse tipo de movimento para os dados dos outros três participantes.

Pinça: Para esse movimento, apesar do `logdetector` (que teve dificuldade em separar o movimento de pinça de segurar o cartão no participante 3) fornecer boa separação do movimento, outras características se sobressaíram, como VAR e RMS.

Segurar Caneta: A característica `logdetector` foi a única característica que conseguiu separar esse movimento com facilidade dos outros, para os dados do participante 1. Já para o participante 2, por exemplo, `logdetector` não foi uma boa escolha, visto que os dados ficaram muito dispersos e se sobreporam com outros movimentos. Para os dados dos participante 3 e 4 a característica conseguiu separar bem os dados, reforçando que a característica continua sendo uma ótima opção.

Segurar Cartão: Para esse movimento, IEMG, MAV e `logdetector` apresentaram a melhor performance. Enquanto o `logdetector` não separou bem os dados para esse participante, de forma complementar, as duas tiveram um bom desempenho para separar esse movimento dos demais.

No apêndice B desse trabalho são apresentados os relatórios de classificação para as características, que corroboram quantitativamente para a análise exposta.

4.3 Dataset 3

4.3.1 Visão Geral das características no domínio do tempo

Para o *dataset 3*, para o domínio do tempo, a acurácia se distribui ao longo do *dataset* conforme apresentado na tabela 10:

Tabela 10 – Resumo dos resultados para as características no domínio do tempo - *dataset 3*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	32767	32767	32767	32767
Média	97,37%	98,28%	95,65%	97,36%
Desvio Padrão	1,39%	0,99%	1,27%	0,81%
Mínimo	27,27%	31,52%	44,85%	49,70%
25%	96,97%	98,18%	95,15%	96,97%
50%	97,58%	98,18%	95,76%	97,58%
75%	98,18%	98,79%	96,36%	97,58%
Máximo	99,39%	100,00%	97,58%	98,79%

Fonte: Autoria Própria, 2022.

Analisando-se os dados da tabela acima, é possível ver que a acurácia melhorara conforme mais características são utilizadas, assim como nos *datasets 1* e *2*. Para esse *dataset*, o desvio padrão médio dos dados é um pouco maior comparado ao *dataset 1* e menor comparado *dataset 2*.

4.3.2 Visão Geral das Características no domínio da frequência

Para o *dataset 3*, no domínio da frequência a acurácia se distribui ao longo do *dataset* conforme apresentado na tabela 11:

Tabela 11 – Resumo dos resultados para as características no domínio da frequência - *dataset 3*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	127	127	127	127
Média	70,63%	74,32%	66,09%	72,26%
Desvio Padrão	6,40%	3,00%	4,58%	5,33%
Mínimo	32,12%	47,88%	30,91%	38,79%
25%	67,88%	72,73%	65,45%	69,09%
50%	71,52%	74,55%	67,88%	72,73%
75%	75,76%	76,06%	68,48%	76,36%
Máximo	79,39%	78,18%	69,70%	78,79%

Fonte: Autoria Própria, 2022.

Na tabela 11 é possível ver que o mesmo comportamento visto anteriormente se aplica para esse *dataset*, apesar da acurácia um menor em relação às características do domínio do tempo.

4.3.3 Investigação das melhores características e suas relações com os movimentos

Para os 4 participantes, a pior característica obtida também foi a TM5. Já as combinações de características que forneceram a melhor acurácia por participante é sumarizada na tabela 12 a seguir:

Tabela 12 – Melhor combinação de características no domínio do tempo - *dataset 3*

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(RMS, WL, vfeature)	99,39%
Participante 2	(RMS, vfeature, logdetector)	100%
Participante 3	(vfeature, logdetector)	97,58%
Participante 4	(WL, vfeature, logdetector)	98,79%

Fonte: Autoria Própria, 2022.

De modo semelhante, as melhores combinações de características no domínio da frequência foram as mostradas na tabela 13:

Tabela 13 – Melhor combinação de características no domínio da frequência - *dataset 3*

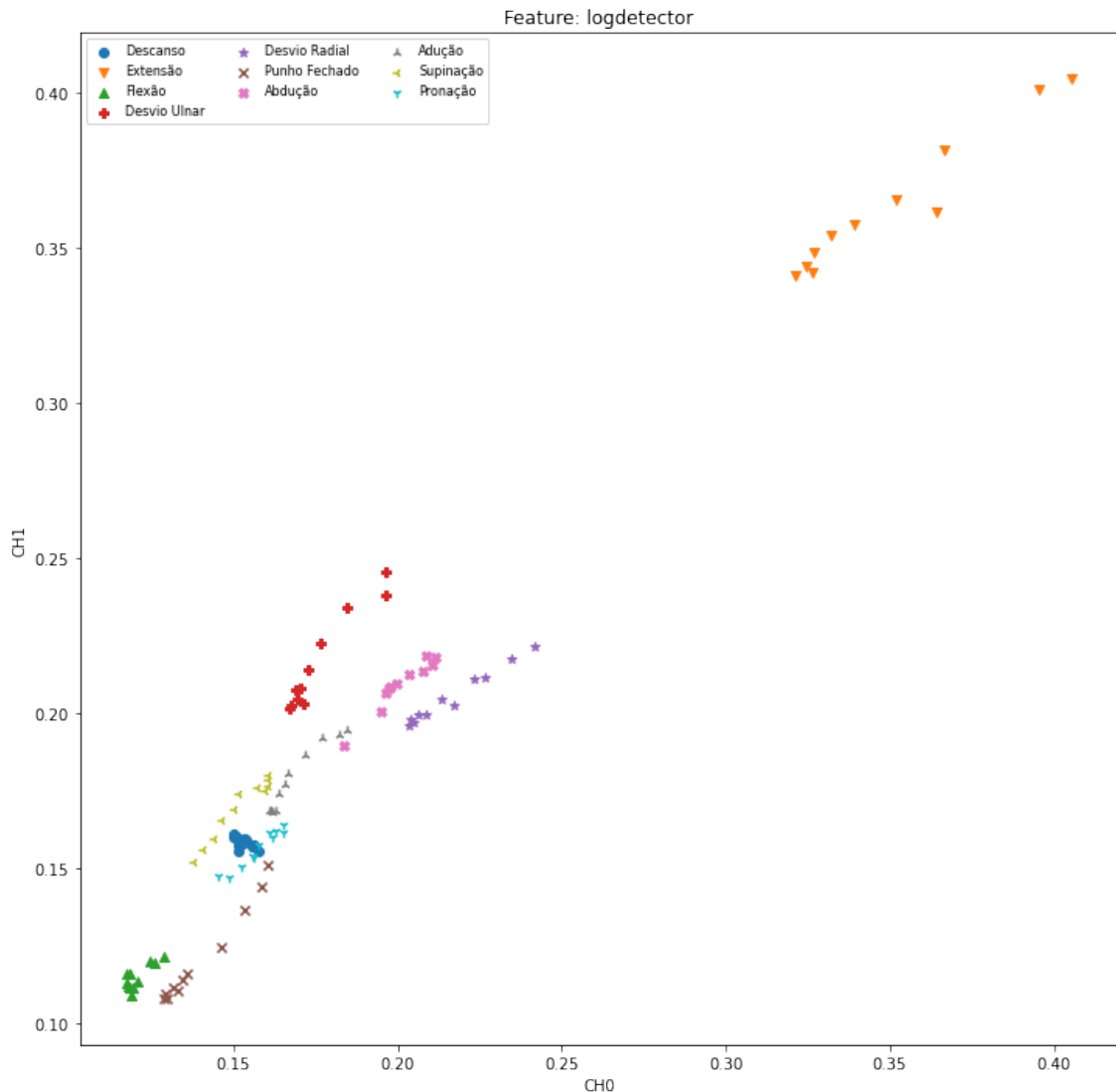
Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(medianFreq, SM3)	79,39%
Participante 2	(medianFreq, meanPower, PKF, SM1, SM2, SM3)	78,18%
Participante 3	(medianFreq, PFK, SM3)	69,70%
Participante 4	(SM1, SM2, SM3)	78,79%

Fonte: Autoria Própria, 2022.

Para esse *dataset*, foram apenas dois movimentos que apresentaram uma divisão clara: extensão e desvio ulnar. Conseqüentemente, tornou a análise visual por meio da função criada um pouco menos intuitiva.

Entretanto, ao olhar nos gráficos fornecidos, as características que fazem parte das melhores combinações para o *dataset* (RMS, WL, Vfeature e logdetector) conseguiram dividir, mesmo assim, alguns movimentos.

Figura 25 – Performance da característica `logdetector` para os movimentos do participante 2.



Fonte: Autoria Própria, 2022.

Na figura 25 acima, é possível notar que o `logdetector` conseguiu separar bem os movimentos de flexão, punho fechado, abdução, desvio radial, além do desvio ulnar e extensão, já mencionados; apenas para os movimentos de pronação, supinação e descanso foram sobrepostos.

O relatório de classificação para essa característica de forma isolada reforça o que a figura anterior mostrou:

É importante levar em consideração também, que ao contrário dos *datasets* 2 e 3, por exemplo, onde foram classificados apenas 6 movimentos, para esse *dataset*, a análise se estendeu para 10.

No apêndice B desse trabalho são apresentados os relatórios de classificação para as características, que corroboram quantitativamente para a análise exposta.

Figura 26 – Relatório de classificação para Logdetector para os movimentos do participante 2.

	precision	recall	f1-score	support
Descanso	0.89	1.00	0.94	16
Extensão	1.00	1.00	1.00	24
Flexão	1.00	1.00	1.00	13
Desvio Ulnar	1.00	1.00	1.00	16
Desvio Radial	1.00	1.00	1.00	11
Punho Fechado	1.00	1.00	1.00	16
Abdução	0.95	1.00	0.97	19
Adução	0.90	0.75	0.82	12
Supinação	0.88	0.94	0.91	16
Pronação	0.90	0.82	0.86	22
accuracy			0.95	165
macro avg	0.95	0.95	0.95	165
weighted avg	0.95	0.95	0.95	165

Fonte: Autoria Própria (2022).

4.4 Dataset 4

4.4.1 Visão Geral das características no domínio do tempo

Para o *dataset 4*, para o domínio do tempo, a acurácia se distribui ao longo do *dataset* conforme apresentado na tabela 14:

Tabela 14 – Resumo dos resultados para as características no domínio do tempo - *dataset 4*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	32767	32767	32767	32767
Média	90,34%	89,43%	89,12%	90,16%
Desvio Padrão	2,43%	3,00%	2,30%	1,88%
Mínimo	41,38%	32,18%	40,23%	34,48%
25%	89,66%	88,51%	87,36%	89,66%
50%	90,80%	90,80%	89,66%	90,80%
75%	91,95%	90,80%	90,80%	90,80%
Máximo	96,55%	95,40%	94,25%	94,25%

Fonte: Autoria Própria, 2022.

Analisando-se os dados da tabela acima, é possível ver que a tendência da acurácia melhorar conforme mais características são utilizadas também ocorre, assim como nos outros três *datasets*. Para esse *dataset*, o desvio padrão médio dos dados é o maior dos *datasets*, embora os valores não sejam discrepantes.

4.4.2 Visão Geral das Características no domínio da frequência

Para o *dataset 4*, no domínio da frequência a acurácia se distribui ao longo do *dataset* conforme apresentado na tabela 15:

Na tabela é possível ver que o mesmo comportamento visto anteriormente se aplica para esse *dataset*, em termos de distribuição de acurácia ao longo das combinações. Entre-

Tabela 15 – Resumo dos resultados para as características no domínio da frequência - *dataset 2*

Métricas	Participante 1	Participante 2	Participante 3	Participante 4
Contagem	127	127	127	127
Média	44,92%	44,63%	47,04%	46,33%
Desvio Padrão	4,57%	7,23%	5,07%	6,25%
Mínimo	19,54%	28,74%	19,54%	29,89%
25%	41,38%	39,08%	44,83%	43,68%
50%	45,98%	42,53%	47,13%	48,28%
75%	48,28%	52,87%	50,57%	50,57%
Máximo	52,87%	57,47%	57,47%	57,47%

Fonte: Autoria Própria, 2022.

tanto, os valores de acurácia apresentados pelo classificador não foram muito satisfatórias. Provavelmente, como o *dataset* possui uma taxa de amostragem pequena e, conseqüentemente, as amostras do conjunto de treino são menores, o classificador não está sendo capaz de generalizar bem. Dado isso, uma possibilidade seria mudar o janelamento dos sinais para que a sobreposição dos sinais seja maior e mais pontos sejam transferidos para o classificador.

4.4.3 Investigação das melhores características e suas relações com os movimentos

Para os 4 participantes, a pior característica obtida também foi a TM5. Já as combinações de características que forneceram a melhor acurácia pra cada conjunto de cada participante é resumida na tabela 16:

Tabela 16 – Melhor combinação de características no domínio do tempo - *dataset 4*

Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(VAR, RMS, WL, TM5, DASDV)	96,55%
Participante 2	(WL, IEMG, logdetector)	95,40%
Participante 3	(RMS, TM4, logdetector)	94,25%
Participante 4	(TM5, vfeature, logdetector)	94,25%

Fonte: Autoria Própria, 2022.

De modo semelhante, as melhores combinações de características no domínio da frequência são mostradas na tabela 17:

Tabela 17 – Melhor combinação de características no domínio da frequência - *dataset 4*

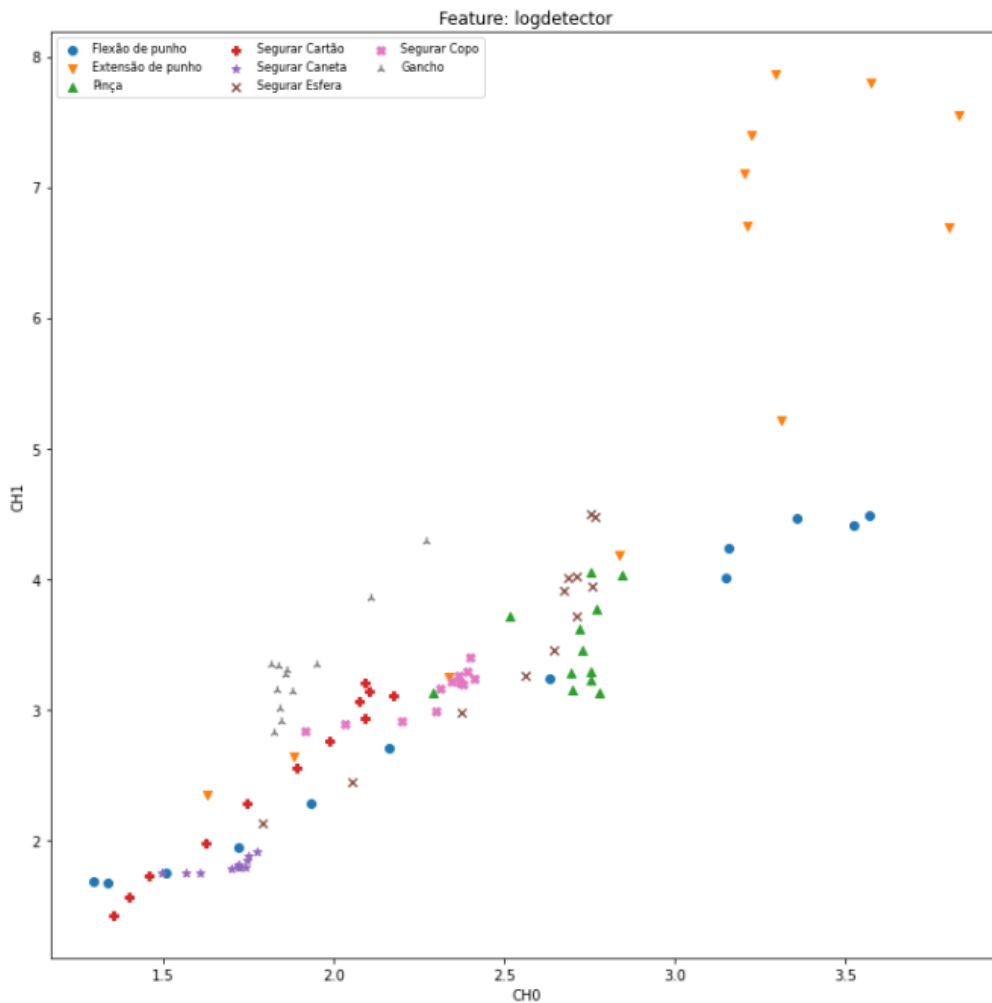
Participante	Melhor Conjunto de Características	Acurácia
Participante 1	(SM3)	52,87%
Participante 2	(SM3)	57,47%
Participante 3	(SM1, SM2)	57,47%
Participante 4	(medianFreq, SM2, SM3)	57,47%

Fonte: Autoria Própria, 2022.

Para esse *dataset*, a análise de cada movimento também fica menos intuitiva por meio do gráfico de características, como mostra a figura 27, que mostra a distribuição dos dados selecionando-se o terceiro participante e `logdetector` como característica.

Entretanto, por meio do relatório de classificação das melhores características selecionadas, é possível ver com mais clareza a combinação de cada uma para os todos os movimentos.

Figura 27 – Performance da característica `logdetector` para os movimentos do participante 4.



Fonte: Autoria Própria (2022)..

Para o `logdetector`, característica que aparece na maioria das combinações das melhores características, o relatório de classificação é exibido à seguir na figura 28:

A figura 27 mostrou que o movimento "flexão de punho" está extremamente disperso em toda área do gráfico, se misturando com os outros movimentos. Isso explica, por exemplo, a baixa precisão mostrada no relatório de classificação da figura 28.

No apêndice B desse trabalho são apresentados os relatórios de classificação para as características, que corroboram quantitativamente para a análise exposta. Eles mostram, também, que para as características presentes nas melhores combinações para todos os participantes são de fato as adequadas para esses movimentos.

Figura 28 – Performance da característica logdetector para os movimentos do participante 3.

	precision	recall	f1-score	support
Flexão de punho	0.53	0.80	0.64	10
Extensão de punho	0.91	1.00	0.95	10
Pinça	1.00	0.62	0.77	8
Segurar Cartão	0.71	0.50	0.59	10
Segurar Caneta	0.86	1.00	0.92	12
Segurar Esfera	0.82	0.75	0.78	12
Segurar Copo	0.92	1.00	0.96	12
Gancho	1.00	0.85	0.92	13
accuracy			0.83	87
macro avg	0.84	0.82	0.82	87
weighted avg	0.85	0.83	0.83	87

Fonte: Autoria Própria (2022).

5 CONCLUSÃO

O trabalho exposto apresentou uma investigação de diversas técnicas de extração de características e suas relações com os movimentos de mãos executados no dia a dia. Após uma extensa análise da literatura, foram selecionados 4 *datasets*, que tiveram seus dados inseridos no ambiente `jupyter notebook` para serem filtrados, segmentados, passarem pela extração de características para finalmente, então, serem combinados e classificados. A partir dos dados de acurácia fornecidos, tiveram seus resultados analisados iterativamente, de forma qualitativa e quantitativa. Variadas foram as técnicas de extração de características implementadas e testadas, tanto no domínio do tempo quanto no domínio da frequência.

De forma geral, as características no domínio do tempo tiveram um desempenho bem superior às características implementadas no domínio da frequência. Apesar de ser um ponto que poderá ser melhor estudado, as características no domínio da frequência tiveram um desempenho um pouco melhor apenas nos *datasets* cujas frequências de amostragem eram maiores (2000 Hz e 4000 Hz). Isso, talvez, se deve ao fato de que quanto maior à frequência de amostragem de um conjunto de dados, maior será o número de amostras disponíveis. Logo, é razoável pensar que as características no domínio da frequência precisam de mais pontos para tornar os movimentos mais definidos e, conseqüentemente, fornecer mais dados para os conjuntos de treino e teste, a serem classificados. Outro ponto importante, ainda, se deve ao fato de que a extração de características na frequência é mais custosa computacionalmente, pelo fato de envolver transformações. Sob a ótica de uma aplicação online, ou seja, onde os dados deveriam ser recebidos e classificados em tempo real, as características nesse domínio não teriam seu uso tão atrativo.

Para as características no domínio do tempo, grupo cujo foi desempenho melhor, a disparidade entre quais seriam características boas e quais seriam ruins ficou bem óbvia. Para todos os *datasets* e participantes analisados, as características TM3, TM4 e TM5 apresentaram um desempenho ruim, enquanto que VAR, RMS, WL e *logdetector*. Esta última, especialmente, nas análises intuitivas através do gráfico das características, conseguiu separar os dados dos movimentos em quase todos os conjuntos de dados. Os resultados foram reforçados, ainda, pelos relatórios de classificação.

Um ponto importante é que esse o presente trabalho se limitou à investigar os efeitos das diversas técnicas de extração de características na classificação dos sinais. A filtragem e a segmentação de sinais, por exemplo, são duas etapas que podem ser exploradas com mais afinco em trabalhos futuros, combinando os parâmetros ótimos encontrados para essas duas etapas com os parâmetros encontrados. Para a filtragem dos sinais, outros tipos de implementação de filtros poderiam ser testados, como filtros FIR e outras aproximações além da *Butterworth*. Para a segmentação dos sinais, é possível testar diferentes números de janelas para um *dataset*, além de diversas porcentagens de sobreposição entre as janelas. Também pode ser assertivo testar outras técnicas de janelamento que reduzam a inserção de artefatos

no domínio da frequência. O presente estudo também não contemplou a hiper-parametrização do classificador, cujos parâmetros poderiam fornecer resultados ótimos. Outra sugestão para trabalhos futuros está relacionada ao uso de outras técnicas para apresentação dos resultados, como diagramas de caixa (conhecidos como *boxplots*) para apresentar os resultados de maneira mais objetiva e enxuta. Os gráficos também permitiriam uma análise de um conjunto de dados (maior de participantes).

REFERÊNCIAS

- BASMAJIAN, J.; LUCA, C. D. **Muscles Alive: Their Function Revealed by Electromyography**. 2. ed. [S.l.]: Williams & Wilkins, 1985. v. 1. 81 p.
- DUDA, R.; HART, P.; STORK, D. **Pattern Classification**. 2. ed. New York: Wiley, 2000. v. 1.
- FARINA, D.; HOLOBAR, A. Characterization of Human Motor Units From Surface EMG Decomposition. **Proceedings of the IEEE**, v. 104, n. 2, p. 353–373, 2 2016. ISSN 0018-9219.
- GESELOWITZ, D. B. Magnetocardiography: An Overview. **IEEE Transactions on Biomedical Engineering**, BME-26, n. 9, p. 497–504, 9 1979. ISSN 0018-9294.
- HOSSAIN, M. M.; MIAH, M. S. Evaluation of different SVM kernels for predicting customer churn. *In: 2015 18th International Conference on Computer and Information Technology (ICCI)*. [S.l.]: IEEE, 2015. p. 1–4. ISBN 978-1-4673-9930-2.
- ISLAM, M. A. *et al.* Mechanomyography Sensor Development, Related Signal Processing, and Applications: A Systematic Review. **IEEE Sensors Journal**, v. 13, n. 7, p. 2499–2516, 7 2013. ISSN 1530-437X.
- JEPPESEN, J. H. *et al.* A cloud detection algorithm for satellite imagery based on deep learning. **Remote Sensing of Environment**, v. 229, p. 247–259, 8 2019. ISSN 00344257.
- KAVYA, R. *et al.* Photoplethysmography — a Modern Approach and Applications. *In: 2020 International Conference for Emerging Technology (INCET)*. [S.l.]: IEEE, 2020. p. 1–4. ISBN 978-1-7281-6221-8.
- KHUSHABA, R. N. *et al.* Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features. **Neural Networks**, v. 55, p. 42–58, 7 2014. ISSN 08936080.
- KO, L.-W. *et al.* Development of SSVEP-based BCI using common frequency pattern to enhance system performance. *In: 2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI)*. [S.l.]: IEEE, 2014. p. 30–35. ISBN 978-1-4799-4543-6.
- LOLURE, A.; THOOL, V. R. Wavelet transform based EMG feature extraction and evaluation using scatter graphs. *In: 2015 International Conference on Industrial Instrumentation and Control (IIC)*. [S.l.]: IEEE, 2015. p. 1273–1277. ISBN 978-1-4799-7165-7.
- LUKACS, P.; PIETRIKOVA, E. Wrist Rehabilitation in Carpal Tunnel Syndrome by Gaming using EMG Controller. *In: 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. [S.l.]: IEEE, 2020. p. 412–417. ISBN 978-1-6654-2226-0.
- NAZMI, N. *et al.* A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions. **Sensors**, v. 16, n. 8, p. 1304, 8 2016. ISSN 1424-8220.
- NEGI, S.; KUMAR, Y.; MISHRA, V. M. Feature extraction and classification for EMG signals using linear discriminant analysis. *In: 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Fall)*. [S.l.]: IEEE, 2016. p. 1–6. ISBN 978-1-5090-3480-2.
- OROSCO, E. C.; LOPEZ, N. M.; SCIASCIO, F. di. Bispectrum-based features classification for myoelectric control. **Biomedical Signal Processing and Control**, v. 8, n. 2, p. 153–168, 3 2013. ISSN 17468094.

- OSKOEI, M. A.; HU, H. GA-based Feature Subset Selection for Myoelectric Classification. *In: 2006 IEEE International Conference on Robotics and Biomimetics*. [S.l.]: IEEE, 2006. p. 1465–1470. ISBN 1-4244-0570-X.
- OZDEMIR, M. A. *et al.* Dataset for multi-channel surface electromyography (sEMG) signals of hand gestures. **Data in Brief**, v. 41, p. 107921, 4 2022. ISSN 23523409.
- PANCHOLI, S.; AGARWAL, R. Development of low cost EMG data acquisition system for arm activities recognition. *In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.]: IEEE, 2016. p. 2465–2469. ISBN 978-1-5090-2029-4.
- PATEL, A. *et al.* EMG-based Human Machine Interface Control. *In: 2019 12th International Conference on Human System Interaction (HSI)*. [S.l.]: IEEE, 2019. p. 127–131. ISBN 978-1-7281-3980-7.
- PHINYOMARK, A.; LIMSAKUL, C.; PHUKPATTARANONT, P. Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification. **Measurement Science Review**, v. 11, n. 2, 1 2011. ISSN 1335-8871.
- PHINYOMARK, A.; LIMSAKUL, C.; PHUKPATTARANONT, P. Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification. **Measurement Science Review**, v. 11, n. 2, 1 2011. ISSN 1335-8871.
- RAMADAN, R. A. *et al.* Basics of Brain Computer Interface. *In: .* [S.l.: s.n.], 2015. p. 31–50.
- SAMUEL, O. W. *et al.* Activity Recognition Based on Pattern Recognition of Myoelectric Signals for Rehabilitation. *In: .* [S.l.: s.n.], 2017. p. 427–442.
- SAPSANIS, C.; GEORGOULAS, G.; TZES, A. EMG based classification of basic hand movements based on time-frequency features. *In: 21st Mediterranean Conference on Control and Automation*. [S.l.]: IEEE, 2013. p. 716–722. ISBN 978-1-4799-0997-1.
- SENGAR, N.; DUTTA, M. K.; BURGET, R. Detection of neuro muscular disease using EMG signals in wavelet domain. *In: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*. [S.l.]: IEEE, 2017. p. 624–627. ISBN 978-1-5386-3004-4.
- SIJIANG, D.; VUSKOVIC, M. Temporal vs. spectral approach to feature extraction from prehensile EMG signals. *In: Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004*. [S.l.]: IEEE, 2004. p. 344–350. ISBN 0-7803-8819-4.
- SMITH, L. H. *et al.* Determining the Optimal Window Length for Pattern Recognition-Based Myoelectric Control: Balancing the Competing Effects of Classification Error and Controller Delay. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 19, n. 2, p. 186–192, 4 2011. ISSN 1534-4320.
- SPIEWAK, C. A Comprehensive Study on EMG Feature Extraction and Classifiers. **Open Access Journal of Biomedical Engineering and Biosciences**, v. 1, n. 1, 2 2018. ISSN 26374579.
- SUDARSAN, S.; SEKARAN, E. C. Design and Development of EMG Controlled Prosthetics Limb. **Procedia Engineering**, v. 38, p. 3547–3551, 2012. ISSN 18777058.

SUPRATAK, A. *et al.* Survey on Feature Extraction and Applications of Biosignals. *In*: HOLZINGER, A. (Ed.). [S.l.: s.n.], 2016. p. 161–182.

ULLAH, S.; IQBAL, K. A Preliminary Review on EMG Signals for Assessment of Diabetic Peripheral Neuropathy Disorder. *In*: **2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)**. [S.l.]: IEEE, 2020. p. 42–46. ISBN 978-1-7281-6788-6.

WEIDERPASS, H. A. *et al.* Time-frequency analysis methods for detecting effects of diabetic neuropathy. **International Journal for Numerical Methods in Biomedical Engineering**, v. 29, n. 9, p. 1000–1010, 9 2013. ISSN 20407939.

ZARDOSHTI-KERMANI, M. *et al.* EMG feature evaluation for movement control of upper extremity prostheses. **IEEE Transactions on Rehabilitation Engineering**, v. 3, n. 4, p. 324–333, 1995. ISSN 10636528.

ZECCA, M. *et al.* Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal. **Critical Reviews? in Biomedical Engineering**, v. 30, n. 4-6, p. 459–485, 2002. ISSN 0278-940X.

APÊNDICE A – Código Utilizado para Processamento dos Dados

Código Utilizado para Processamento dos Dados

Listing A.1 – Métodos de filtragem e plotagem das PSDs

```
1 # definições de filtros
2
3 def butter_bandpass(data, lowcut, highcut, fs=2000, order=4):
4     nyq = fs * 0.5
5     low = lowcut / nyq
6     high = highcut / nyq
7     b, a = signal.butter(order, [low, high], btype='bandpass')
8     return signal.filtfilt(b, a, data)
9
10
11 def butter_lowpass(data, lowcut, fs=2000, order=4):
12     nyq = fs * 0.5
13     low = lowcut / nyq
14     b, a = signal.butter(order, low, btype='lowpass')
15     return signal.filtfilt(b, a, data)
16
17
18 def butter_highpass(data, highcut, fs=2000, order=4):
19     nyq = fs * 0.5
20     high = highcut / nyq
21     b, a = signal.butter(order, high, btype='highpass')
22     return signal.filtfilt(b, a, data)
23
24
25 def butter_notch(data, cutoff, var=1, fs=2000, order=4):
26     nyq = fs * 0.5
27     low = (cutoff - var) / nyq
28     high = (cutoff + var) / nyq
29     b, a = signal.iirfilter(order, [low, high], btype='bandstop', ftype="butter")
30     return signal.filtfilt(b, a, data)
31
```

```

32     def print_graphs(data):
33     for i in range(data.shape[2]):
34         plt.plot(data[3,0,i,:])
35     plt.title('Domínio do tempo')
36     plt.show()
37
38     for i in range(data.shape[2]):
39         plt.psd(data[3,0,i,:], Fs=2000)
40     plt.title('Domínio da frequência')
41     plt.show()
42
43 from matplotlib import rcParams
44 import matplotlib.pyplot as plt
45
46 rcParams['figure.figsize'] = [16., 5.]
47 print_graphs(X1)

```

Listing A.2 – Código de Segmentação do Sinal para tempo e frequência

```

1 from scipy.signal import stft
2
3 step = 1200
4 segment = 8192
5 print('', data.shape)
6
7 n_win = int((data.shape[-1] - segment) / step) + 1
8 print(data.shape[-1])
9 print(n_win)
10 ids = np.arange(n_win) * step
11
12
13 # Janelas do dado no domínio do tempo
14 chunks_time = np.array([data[:, :, :, k:(k + segment)] for k in ids]).
15 transpose(1, 2,3,0,4)
16
17 # Janelas do dado no domínio da frequência

```

```

18 f, t, chunks_freq = stft(data, fs=4000, nperseg=8192, noverlap=5700)
19 chunks_freq = np.swapaxes(chunks_freq, 3, 4)
20
21 print('Formato (shape) dos dados depois da divisão de janelas')
22 print(f'Domínio do tempo: {chunks_time.shape} – (classes, ensaios, canais,
23 janelas, linhas)')
24 print(f'Domínio da frequência: {chunks_freq.shape} – (classes, ensaios, canais,
25 janelas, linhas)')

```

Listing A.3 – Funções de Extração de Características

```

1 from math import prod
2
3 # funções auxiliares
4 def PSD(w):
5     ''' definição da função PSD para o sinal no domínio da frequência '''
6     return np.abs(w)**2
7
8
9 # funções de extração de características
10
11
12 def iemg(x):
13     return np.sum(np.abs(x), axis = -1) #-1 soma o vetor referente do ultimo axis,
14     que é o vetor com os dados
15
16 def mav(x):
17     return (np.sum(np.abs(x), axis = -1)) / (x.shape[-1])
18
19
20 def mod_mav_type1(x):
21
22     N = (x.shape[-1])
23     window = np.array([1 if (int(0.25*N) <= i <= int(0.75*N))
24     else 0.5 for i in range((x.shape[-1]))])
25

```

```

26     x_win = x*window
27
28     return (np.sum(np.abs(x_win), axis = -1)) / (x.shape[-1])
29
30
31 def mod_mav_type2(x): #Modified Mean Absolute Value Type 2
32
33     N = (x.shape[-1])
34     window = np.array([1 if ((0.25*N) <= i <= (0.75*N)) else
35 4*i/N if i<0.25*N else 4*(i-N)/N for i in range((x.shape[-1]))])
36
37     x_win = x*window
38
39     return (np.sum(np.abs(x_win), axis = -1)) / (x.shape[-1])
40
41 def tm3(x):
42
43     return np.abs(np.sum(x ** 3, axis=-1) / ((x.shape[-1])))
44
45 def tm4(x):
46
47     return np.abs(np.sum(x ** 4, axis=-1) / ((x.shape[-1])))
48
49
50 def tm5(x):
51
52     return np.abs(np.sum(x ** 5, axis=-1) / ((x.shape[-1])))
53
54 def vfeature(x):
55
56     v = 4 # p/ v = 2, v_feature = RMS
57
58     return np.power(np.sum(x ** v, axis=-1) / ((x.shape[-1])),1/v)
59

```

```

60
61 def logdetector(x):
62
63     return np.exp((np.sum(np.log10(np.abs(x)), axis = -1))/ x.shape[-1])
64
65 def aac(x):
66
67     return wl(x)/x.shape[-1]
68
69 def dasdv(x):
70
71     return np.sqrt(sum(np.power(x[:, :, :, :, i+1] - x[:, :, :, :, i], 2)
72     for i in range(x.shape[-1] - 1))/(x.shape[-1]-1))
73
74
75 def var(x):
76     return np.sum(x ** 2, axis=-1) / ((x.shape[-1]) - 1)
77
78 def rms(x):
79     return np.sqrt(np.sum(x ** 2, axis=-1) / ((x.shape[-1])))
80
81 def wl(x):
82     return sum([np.abs(x[:, :, :, :, i+1] - x[:, :, :, :, i]) for i in
83     range(x.shape[-1] - 1)])
84
85 def ssi(x):
86     return np.sum(x ** 2, axis=-1)
87
88 #def zc(x):
89 #
90 #     th = 0.1
91 #     cruce = 0
92 #     for i in range(x.shape[-1] - 1):
93 #         can = x[:, :, :, :, i]*x[:, :, :, :, i+1]

```

```

94 #         can2 = abs(x[:, :, :, :, i] - x[:, :, :, :, i+1])
95 #         if (can < 0) and (can2 > th):
96 #             cruce = cruce + 1
97 #     return cruce
98
99 def median_frequency(w):
100     return np.sum(PSD(w), axis=-1) / 2
101
102 def mean_power(w):
103     return (np.sum(PSD(w), axis=-1) / ((w.shape[-1])))
104
105 def pkf(f, t, w):
106
107     return f[PSD(w).argmax(axis = -1)]
108
109 def total_power(w): #também chamado de SM0 (Zero Spectral Moment)
110
111     return np.sum(PSD(w), axis = -1)
112
113 def SM1(f, t, w):
114
115     return np.sum(PSD(w)*f, axis = -1)
116
117 def SM2(f, t, w):
118
119     return np.sum(PSD(w)*f**2, axis = -1)
120
121 def SM3(f, t, w):
122
123     return np.sum(PSD(w)*f**3, axis = -1)

```

Listing A.4 – Combinações de Características e Classificação

```

1
2 features = list()
3 features.append(var(chunks_time))

```

```
4 features.append(rms(chunks_time))
5 features.append(wl(chunks_time))
6 features.append(ssi(chunks_time))
7 features.append(iemg(chunks_time))
8 features.append(zc(chunks_time))
9 features.append(mav(chunks_time))
10 features.append(mod_mav_type1(chunks_time))
11 features.append(mod_mav_type2(chunks_time))
12 features.append(tm3(chunks_time))
13 features.append(tm4(chunks_time))
14 features.append(tm5(chunks_time))
15 features.append(vfeature(chunks_time))
16 features.append(logdetector(chunks_time))
17 features.append(aac(chunks_time))
18 features.append(dasdv(chunks_time))
19 features.append(median_frequency(chunks_freq))
20 features.append(mean_power(chunks_freq))
21 features.append(total_power(chunks_freq))
22 features.append(pkf(f, t, chunks_freq))
23 features.append(SM1(f, t, chunks_freq))
24 features.append(SM2(f, t, chunks_freq))
25 features.append(SM3(f, t, chunks_freq))
26
27 features_names = list()
28 features_names = ['var', 'rms', 'ssi', 'iemg', 'mav', 'mod1', 'mod2', 'tm3', 'tm4',
29                  'tm5', 'vfeature', 'logdetector', 'aac', 'dasdv', 'sm2', 'sm3']
30
31
32 features = np.array(features)
33 plot_feat = features
34
35 len_features = features.shape
36 print(len_features)
37
```

```
38
39 from itertools import combinations
40
41 feature_combinations = list()
42 labels_combinations = list()
43
44
45 for n_combinations in range(1, features.shape[0]+1):
46     feature_combinations.append([combination for combination
47     in combinations(features, n_combinations)])
48     labels_combinations.append([combination for combination
49     in combinations(features_names, n_combinations)])
50
51
52 print(type(feature_combinations))
53 print(len(feature_combinations))
54 print(len(feature_combinations[1]))
55
56 accuracy_list = list()
57 labels_list = list()
58
59 for i in range(len(feature_combinations)):
60     for j in range(len(feature_combinations[i])):
61         print("Conjunto de Características: ", labels_combinations[i][j])
62
63         features = feature_combinations[i][j]
64
65         features_final = list()
66         for feature in (features):
67             feature = np.array(feature)
68             feature = feature.transpose(0, 1, 3, 2)
69             feature = feature.reshape(feature.shape[0] *
70             feature.shape[1] * feature.shape[2], 7)
71
```

```

72         features_final.append(feature)
73
74     final_data = np.concatenate(features_final, axis = 1)
75
76     movs = list(range(1, 9))
77
78     y = np.repeat(movs, len_features[2]*len_features[-1])
79
80
81     X_train, X_test, y_train, y_test = train_test_split(final_data,
82     y, test_size=0.3, shuffle=True, random_state = 42)
83
84
85     pipe1 = Pipeline([( 'scaler', StandardScaler()),
86     ( 'svc', SVC(gamma = 'auto', C = 10))])
87     pipe1.fit(X_train, y_train)
88     print("Acurácia para SVM: ", pipe1.score(X_test, y_test))
89     accuracy_list.append(pipe1.score(X_test, y_test))
90     labels_list.append(labels_combinations[i][j])
91
92
93 time_results_df = pd.DataFrame(
94     { 'Características Consideradas': labels_list,
95     'Acurácia do SVM': accuracy_list,
96     })
97
98 y_pred = pipe1.predict(X_test)
99 confusion = confusion_matrix(y_test, y_pred)
100 print('Matriz de Confusão p/ último Conjunto:\n')
101 print(confusion)
102
103 print('\nRelatório de Classificação p/ último conjunto:\n')
104 print(classification_report(y_test, y_pred, target_names = classes))

```

Listing A.5 – Melhores e Piores Características

```

1 #pior característica em termos de acurácia para todo o conjunto
2 idx = time_results_df[ 'Acurácia do SVM' ].idxmin()
3 print("Conjunto de Características de pior
4 desempenho para o conjunto de dados: ")
5 print(time_results_df[ 'Características Consideradas' ][ idx ])
6 print("Acurácia: ")
7 print(time_results_df[ 'Acurácia do SVM' ][ idx ])
8
9
10 #melhor característica em termos de acurácia para todo o conjunto
11 idx = time_results_df[ 'Acurácia do SVM' ].idxmax()
12 print("Conjunto de Características de melhor
13 desempenho para o conjunto de dados: ")
14 print(time_results_df[ 'Características Consideradas' ][ idx ])
15 print(time_results_df[ 'Acurácia do SVM' ][ idx ])

```

Listing A.6 – Função para plotar as características

```

1
2 def plot_features(features, features_names, classes_names, ch_x, ch_y):
3     plt.rcParams["figure.figsize"] = (12,12)
4
5
6     movs = np.arange(len(classes_names))
7     feature = np.arange(len(features_names))
8     markers = list(["o", "v", "^", "P", "*", "x", "X", "2", "3", "1"])
9
10    for feat in feature:
11        for mov, marker in zip(movs, markers[0:len(movs)]):
12
13            #primeiro valor: feature, classes, trials, canal, linhas
14            plt.scatter(features[feat, mov, 0, ch_x, :],
15                        features[feat, mov, 0, ch_y, :], marker = marker)
16
17            plt.legend((classes_names),
18                       scatterpoints=1,

```

```
19         loc='best',
20         ncol=3,
21         fontsize=8)
22
23     plt.title('Feature: {}'.format(features_names[feat]))
24     plt.xlabel('CH{}'.format(ch_x))
25     plt.ylabel('CH{}'.format(ch_y))
26     plt.show()
```

APÊNDICE B – Relatórios de Classificação das melhores combinações

As figuras a seguir mostram os relatórios de classificação para as melhores características encontradas, ou seja, as combinações de características no domínio do tempo para cada participante para os quatro *datasets*. Em cada figura são mostrados 4 relatórios, representando os quatro participantes, na seguinte ordem: canto superior esquerdo, primeiro participante, canto superior direito segundo participante, canto inferior esquerdo terceiro participante e, por fim, canto inferior direito quarto participante.

Figura 29 – Relatórios de Classificação - Dataset 1

	precision	recall	f1-score	support
Mão Aberta	1.00	1.00	1.00	61
Descanso	1.00	1.00	1.00	77
Punho Fechado	1.00	1.00	1.00	77
Movimento de Pinça	1.00	1.00	1.00	65
Extensão do Pulso	1.00	1.00	1.00	66
Flexão do Pulso	1.00	1.00	1.00	83
Pronação	1.00	1.00	1.00	79
Supinação	1.00	1.00	1.00	68
accuracy			1.00	576
macro avg	1.00	1.00	1.00	576
weighted avg	1.00	1.00	1.00	576

	precision	recall	f1-score	support
Mão Aberta	1.00	1.00	1.00	61
Descanso	1.00	1.00	1.00	77
Punho Fechado	1.00	1.00	1.00	77
Movimento de Pinça	1.00	1.00	1.00	65
Extensão do Pulso	1.00	1.00	1.00	66
Flexão do Pulso	1.00	1.00	1.00	83
Pronação	1.00	1.00	1.00	79
Supinação	1.00	1.00	1.00	68
accuracy			1.00	576
macro avg	1.00	1.00	1.00	576
weighted avg	1.00	1.00	1.00	576

	precision	recall	f1-score	support
Mão Aberta	0.94	1.00	0.97	61
Descanso	1.00	1.00	1.00	77
Punho Fechado	1.00	1.00	1.00	77
Movimento de Pinça	1.00	1.00	1.00	65
Extensão do Pulso	1.00	1.00	1.00	66
Flexão do Pulso	1.00	1.00	1.00	83
Pronação	1.00	1.00	1.00	79
Supinação	1.00	0.94	0.97	68
accuracy			0.99	576
macro avg	0.99	0.99	0.99	576
weighted avg	0.99	0.99	0.99	576

	precision	recall	f1-score	support
Mão Aberta	1.00	1.00	1.00	61
Descanso	0.99	1.00	0.99	77
Punho Fechado	1.00	1.00	1.00	77
Movimento de Pinça	1.00	1.00	1.00	65
Extensão do Pulso	1.00	1.00	1.00	66
Flexão do Pulso	1.00	0.99	0.99	83
Pronação	1.00	1.00	1.00	79
Supinação	1.00	1.00	1.00	68
accuracy			1.00	576
macro avg	1.00	1.00	1.00	576
weighted avg	1.00	1.00	1.00	576

Fonte: Autoria Própria, 2022.

Figura 30 – Relatórios de Classificação - Dataset 2

	precision	recall	f1-score	support
Cilíndrico	1.00	1.00	1.00	96
Gancho	1.00	0.99	1.00	113
Pinça	1.00	1.00	1.00	95
Segurar Caneta	1.00	1.00	1.00	90
Esférico	0.99	1.00	1.00	100
Segurar Cartão	1.00	1.00	1.00	100
accuracy			1.00	594
macro avg	1.00	1.00	1.00	594
weighted avg	1.00	1.00	1.00	594

	precision	recall	f1-score	support
Cilíndrico	0.99	1.00	0.99	96
Gancho	0.99	1.00	1.00	113
Pinça	0.94	0.87	0.91	95
Segurar Caneta	0.93	0.96	0.95	90
Esférico	1.00	0.99	0.99	100
Segurar Cartão	0.94	0.98	0.96	100
accuracy			0.97	594
macro avg	0.97	0.97	0.97	594
weighted avg	0.97	0.97	0.97	594

	precision	recall	f1-score	support
Cilíndrico	0.98	1.00	0.99	96
Gancho	0.97	0.98	0.98	113
Pinça	0.98	1.00	0.99	95
Segurar Caneta	0.96	0.96	0.96	90
Esférico	0.98	0.95	0.96	100
Segurar Cartão	0.98	0.96	0.97	100
accuracy			0.97	594
macro avg	0.97	0.97	0.97	594
weighted avg	0.97	0.97	0.97	594

	precision	recall	f1-score	support
Cilíndrico	1.00	1.00	1.00	96
Gancho	1.00	1.00	1.00	113
Pinça	0.99	0.97	0.98	95
Segurar Caneta	0.97	0.96	0.96	90
Esférico	1.00	1.00	1.00	100
Segurar Cartão	0.96	0.99	0.98	100
accuracy			0.99	594
macro avg	0.99	0.99	0.99	594
weighted avg	0.99	0.99	0.99	594

Fonte: Autoria Própria, 2022.

Figura 31 – Relatórios de Classificação - Dataset 3

	precision	recall	f1-score	support		precision	recall	f1-score	support
Descanso	1.00	1.00	1.00	16	Descanso	1.00	1.00	1.00	16
Extensão	1.00	1.00	1.00	24	Extensão	1.00	1.00	1.00	24
Flexão	1.00	1.00	1.00	13	Flexão	1.00	1.00	1.00	13
Desvio Ulnar	1.00	1.00	1.00	16	Ulnar	1.00	1.00	1.00	16
Desvio Radial	1.00	1.00	1.00	11	Radial	1.00	1.00	1.00	11
Punho Fechado	1.00	1.00	1.00	16	Grip	1.00	1.00	1.00	16
Abdução	1.00	1.00	1.00	19	Abduction	1.00	1.00	1.00	19
Adução	0.92	1.00	0.96	12	Adduction	1.00	1.00	1.00	12
Supinação	1.00	0.94	0.97	16	Supination	1.00	1.00	1.00	16
Pronação	0.95	0.95	0.95	22	Pronation	1.00	1.00	1.00	22
accuracy			0.99	165	accuracy			1.00	165
macro avg	0.99	0.99	0.99	165	macro avg	1.00	1.00	1.00	165
weighted avg	0.99	0.99	0.99	165	weighted avg	1.00	1.00	1.00	165

	precision	recall	f1-score	support		precision	recall	f1-score	support
Descanso	0.89	1.00	0.94	16	Descanso	0.94	1.00	0.97	16
Extensão	1.00	1.00	1.00	24	Extensão	1.00	1.00	1.00	24
Flexão	1.00	1.00	1.00	13	Flexão	1.00	1.00	1.00	13
Desvio Ulnar	0.94	0.94	0.94	16	Desvio Ulnar	1.00	1.00	1.00	16
Desvio Radial	1.00	1.00	1.00	11	Desvio Radial	1.00	1.00	1.00	11
Punho Fechado	1.00	1.00	1.00	16	Punho Fechado	1.00	1.00	1.00	16
Abdução	0.95	0.95	0.95	19	Abdução	1.00	1.00	1.00	19
Adução	1.00	1.00	1.00	12	Adução	1.00	1.00	1.00	12
Supinação	1.00	1.00	1.00	16	Supinação	1.00	1.00	1.00	16
Pronação	1.00	0.91	0.95	22	Pronação	1.00	0.95	0.98	22
accuracy			0.98	165	accuracy			0.99	165
macro avg	0.98	0.98	0.98	165	macro avg	0.99	1.00	0.99	165
weighted avg	0.98	0.98	0.98	165	weighted avg	0.99	0.99	0.99	165

Fonte: Autoria Própria, 2022.

Figura 32 – Relatórios de Classificação - Dataset 4

	precision	recall	f1-score	support		precision	recall	f1-score	support
Flexão de punho	0.83	1.00	0.91	10	Flexão de punho	1.00	1.00	1.00	10
Extensão de punho	1.00	1.00	1.00	10	Extensão de punho	1.00	1.00	1.00	10
Pinça	0.86	0.75	0.80	8	Pinça	1.00	1.00	1.00	8
Segurar Cartão	1.00	1.00	1.00	10	Segurar Cartão	1.00	0.70	0.82	10
Segurar Caneta	1.00	1.00	1.00	12	Segurar Caneta	0.80	1.00	0.89	12
Segurar Esfera	0.92	0.92	0.92	12	Segurar Esfera	1.00	1.00	1.00	12
Segurar Copo	1.00	1.00	1.00	12	Segurar Copo	1.00	1.00	1.00	12
Gancho	1.00	0.92	0.96	13	Gancho	1.00	1.00	1.00	13
accuracy			0.95	87	accuracy			0.97	87
macro avg	0.95	0.95	0.95	87	macro avg	0.97	0.96	0.96	87
weighted avg	0.96	0.95	0.95	87	weighted avg	0.97	0.97	0.96	87

	precision	recall	f1-score	support		precision	recall	f1-score	support
Flexão de punho	0.83	1.00	0.91	10	Flexão de punho	1.00	1.00	1.00	10
Extensão de punho	1.00	1.00	1.00	10	Extensão de punho	1.00	1.00	1.00	10
Pinça	1.00	0.75	0.86	8	Pinça	1.00	0.62	0.77	8
Segurar Cartão	1.00	1.00	1.00	10	Segurar Cartão	0.67	1.00	0.80	10
Segurar Caneta	0.86	1.00	0.92	12	Segurar Caneta	1.00	1.00	1.00	12
Segurar Esfera	1.00	0.92	0.96	12	Segurar Esfera	1.00	1.00	1.00	12
Segurar Copo	0.92	1.00	0.96	12	Segurar Copo	1.00	1.00	1.00	12
Gancho	1.00	0.85	0.92	13	Gancho	1.00	0.85	0.92	13
accuracy			0.94	87	accuracy			0.94	87
macro avg	0.95	0.94	0.94	87	macro avg	0.96	0.93	0.94	87
weighted avg	0.95	0.94	0.94	87	weighted avg	0.96	0.94	0.94	87

Fonte: Autoria Própria, 2022.