



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



Trabalho de Conclusão de Curso

Explorando os modelos de difusão em sistemas BCI-SSVEP

Gabriel Machado Kaizer Pereira

João Monlevade, MG
2023

Gabriel Machado Kaizer Pereira

**Explorando os modelos de difusão em sistemas
BCI-SSVEP**

Trabalho de Conclusão de curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

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

Coorientador: Me. Guilherme Vettorazzi Vargas

Universidade Federal de Ouro Preto
João Monlevade
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

P436e Pereira, Gabriel Machado Kaizer.
Explorando os modelos de difusão em sistemas BCI-SSVEP.
[manuscrito] / Gabriel Machado Kaizer Pereira. - 2023.
36 f.

Orientadora: Profa. Dra. Sarah Negreiros de Carvalho Leite.
Coorientador: Me. Guilherme Vettorazzi Vargas.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia
Elétrica .

1. Interação homem-máquina. 2. Interface cérebro-computador. 3.
Potencial evocado (Eletrofisiologia). 4. Redes neurais (Computação). I.
Leite, Sarah Negreiros de Carvalho. II. Vargas, Guilherme Vettorazzi. III.
Universidade Federal de Ouro Preto. IV. Título.

CDU 004.5

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Gabriel Machado Kaizer Pereira

Explorando os modelos de difusão em sistemas BCI-SSVEP

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Aprovada em 30 de agosto de 2023.

Membros da banca

Profa. Dra. Sarah Negreiros de Carvalho Leite - Orientadora - Universidade Federal de Ouro Preto
Me. Guilherme Vettorazzi Vargas - Coorientador - Universidade Estadual de Campinas
Profa. Mylena Cruzinha da Silva- Universidade Federal de Ouro Preto
Prof. Dr. Harlei Miguel de Arruda Leite- Universidade Federal de Ouro Preto

Professora Sarah Negreiros de Carvalho Leite, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 31/09/2023.



Documento assinado eletronicamente por **Sarah Negreiros de Carvalho Leite**, PROFESSOR DE MAGISTERIO SUPERIOR, em 31/08/2023, às 22:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0583568** e o código CRC **73C1B038**.

Resumo

Este trabalho tem como principal objetivo a implementação de um sistema de Interface Cérebro-Computador (BCI) baseado em potenciais evocados visualmente em regime estacionário e a subsequente avaliação do impacto do desempenho do sistema quando são empregados sinais cerebrais gerados artificialmente por meio de um modelo de difusão. A pesquisa busca investigar como a inclusão de sinais sintéticos no processo de treinamento pode influenciar a capacidade de detecção de padrões pelo sistema BCI, abrindo novas possibilidades para aprimorar a eficiência e a precisão das interações entre o cérebro humano e as máquinas. Os resultados obtidos apontam um ganho de desempenho quando se incluem sinais sintéticos na etapa de treinamento da BCI. Este resultado é significativo e têm o potencial de contribuir para o avanço das interfaces cérebro-máquina e suas aplicações práticas.

Palavras-chave: BCI, Interface Cérebro-Computador, SSVEP, modelo de difusão, redes neurais artificiais.

Abstract

This study has as its main objective the implementation of a Brain-Computer Interface (BCI) system based on visually evoked potentials in a steady-state condition and the subsequent evaluation of the impact on the system's performance when artificially generated brain signals are employed through a diffusion model. The research aims to investigate how the inclusion of synthetic signals in the training process can influence the system's pattern detection capability, opening up new possibilities to enhance the efficiency and accuracy of interactions between the human brain and machines. The results obtained indicate a performance improvement when synthetic signals are included in the BCI training phase. This result is significant and has the potential to contribute to the advancement of brain-machine interfaces and their practical applications.

Keywords: *BCI, Brain-Computer Interfaces, SSVEP, Diffusion model, Artificial Neural Networks.*

Lista de ilustrações

Figura 1 – Etapas de um sistema BCI com base no paradigma SSVEP.	2
Figura 2 – Representação no domínio do tempo e da frequência do sinal coletado em Oz - Estímulo visual em 10 Hz.	7
Figura 3 – SVM - Maximizando a margem.	10
Figura 4 – Neurônio biológico.	12
Figura 5 – Neurônio artificial.	12
Figura 6 – Funções de ativação.	14
Figura 7 – Rede <i>feedforward</i> de camada única.	15
Figura 8 – Rede <i>feedforward</i> de múltipla camada.	16
Figura 9 – Rede recorrente.	16
Figura 10 – Autoencoder variacional.	19
Figura 11 – Autoencoder variacional condicional.	19
Figura 12 – Etapa <i>forward</i> da difusão.	21
Figura 13 – Amostras latentes de cronogramas lineares (superior) e cosseno (inferior).	23
Figura 14 – Processo reverso da difusão.	23
Figura 15 – Fluxograma do processo.	25
Figura 16 – Acurácia dos cenários testados por indivíduo.	28
Figura 17 – Sinal real e sintético do indivíduo 1 na frequência de 8 Hz - Eletrodo Oz.	30
Figura 18 – Sinal real e sintético do indivíduo 1 na frequência de 10 Hz - Eletrodo Oz.	30
Figura 19 – Sinal real e sintético do indivíduo 1 na frequência de 12 Hz - Eletrodo Oz.	31
Figura 20 – Sinal real e sintético do indivíduo 1 na frequência de 15 Hz - Eletrodo Oz.	31

Sumário

1	INTRODUÇÃO	1
1.1	Objetivo	3
1.2	Estado da arte	3
2	METODOLOGIA	6
2.1	Base de dados	6
2.2	Pré-processamento	6
2.3	Extração de características	7
2.4	Classificador	8
2.4.1	Classificador linear	8
2.4.2	<i>Support vector machine</i>	9
2.5	Redes neurais artificiais	11
2.5.1	Autoencoders variacionais	18
2.5.2	Modelo de difusão	20
2.5.2.1	Processo de difusão	20
2.5.2.2	Processo direto	21
2.5.2.3	Variância	22
2.5.2.4	Processo reverso	22
2.5.2.5	Treinamento do modelo de difusão	24
3	RESULTADOS E DISCUSSÃO	25
3.1	Procedimento experimental	25
3.2	Análise de resultados	26
4	CONCLUSÃO	33
	REFERÊNCIAS	34

1 Introdução

As Interfaces Cérebro-Computador (BCI, do inglês Brain-Computer Interfaces) são sistemas que combinam hardware e software para permitir a interação entre seres humanos e o ambiente ao redor, utilizando sinais de controle gerados a partir da atividade eletroencefalográfica, sem a necessidade de envolver nervos periféricos ou músculos. Dessa forma, as BCI transmitem as intenções dos indivíduos para sistemas externos, como cadeiras de rodas ou braços robóticos, por meio de canais não musculares (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

Uma componente essencial da BCI é o eletroencefalograma (EEG). Dentre diversas definições, um EEG pode ser descrito como a medição dos potenciais que refletem a atividade elétrica cerebral do cérebro humano. Para o seu registro, existem dois métodos distintos: a abordagem não invasiva e a invasiva. No método não invasivo, não há necessidade de procedimentos cirúrgicos, visto que os sinais são coletados sob o couro cabeludo do paciente, por meio de toucas, por exemplo, porém a qualidade dos sinais cerebrais coletados tende a ser inferior à obtida com técnicas invasivas. Estas últimas exigem a implantação de eletrodos diretamente no cérebro do paciente, por meio de cirurgia, tornando-se assim mais arriscadas (LEBEDEV; NICOLELIS, 2009), mas possuem sinais de EEG com melhor sinal-ruído (SNR), considerando que o processo de coleta se dá em camadas mais profundas do cérebro.

Além do apelo assistencial associado ao seu emprego em BCIs, o EEG é amplamente utilizado para o diagnóstico de distúrbios neurológicos, e sua amplitude normalmente varia de cerca de 1 a 100 μV em um adulto em condições normais. Em termos de equipamentos/dispositivos para a coleta de sinais de EEG, são utilizados eletrodos compostos por materiais condutores, como ouro e prata, por exemplo, pois o bom funcionamento deles depende de sua localização e do contato adequado. Em registros não invasivos, por exemplo, um gel condutor pode ser aplicado nas pessoas para garantir um sinal com uma boa relação sinal-ruído, ou eletrodos presos em toucas podem ser empregados.

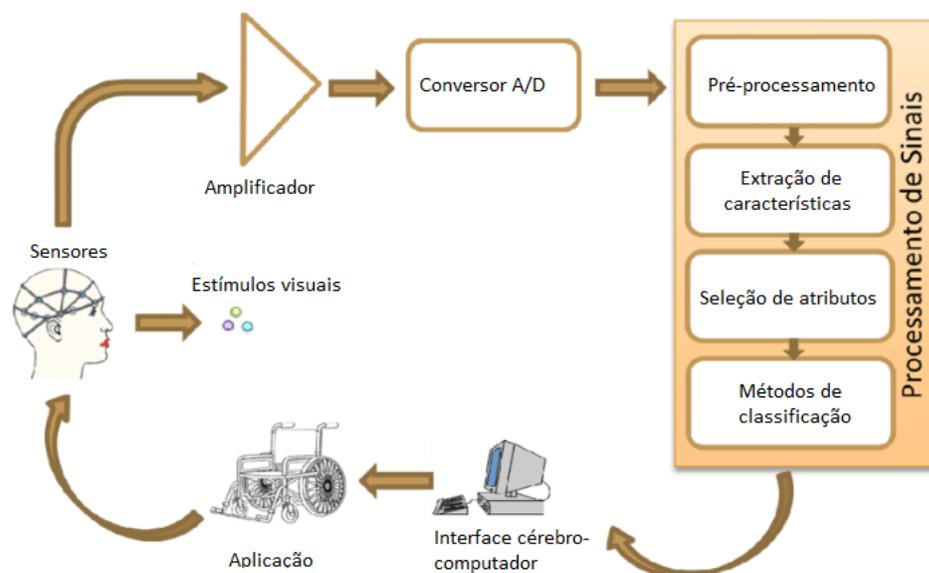
A padronização do posicionamento dos eletrodos foi definida no Primeiro Congresso Internacional de EEG, em 1947. Em 1958, Herbert Jasper sugeriu o método de posicionamento 10-20, utilizado até os dias de hoje (DEMARRE; MICHAELS; VALDEZ, 1983). Por meio desse sistema, os eletrodos são posicionados utilizando os marcos anatômicos da cabeça. Estes podem ser utilizados como referência tornando o posicionamento dos eletrodos o mais uniforme na região do escalpo. Neste sistema, a posição dos eletrodos pode ser referenciada através das seguintes letras: P para parietal, F para frontal, T para temporal, C para central, O para occipital e A para auricular (TYNER; KNOTT; MAYER, 1989). Dessa forma, eletrodos posicionados no hemisfério esquerdo e direito são numerados com números ímpares e pares, respectivamente. Já os colocados sobre a linha central são

nomeados com o índice 'z'. O sistema 10-20 se deu através das distâncias observadas entre os eletrodos, posicionados a cada 10 ou 20% dos marcos anatômicos (násio, ínio e orelhas).

Considerando que o sinal de EEG pode surgir como uma resposta a um estímulo externo, diversas estratégias são exploradas na elaboração de uma BCI. Entre as abordagens utilizadas estão a Imagética Motora, Imaginação de Tarefas, P300 e SSVEP (*Steady State Visually Evoked Potential*). O potencial SSVEP surge como uma resposta contínua do córtex visual a estímulos visuais, que piscam em uma frequência constante. Geralmente, essa resposta se manifesta como uma onda oscilatória quase senoidal, apresentando a mesma frequência fundamental do estímulo e alguns harmônicos (XIE et al., 2016). Com isso, de acordo com um número de frequências de estímulo pré-definido, é possível empregar técnicas de processamento de sinais e aprendizado de máquina para a identificação desses estímulos, de forma a ser possível mapear cada sinal a um comando a ser executado, por exemplo, por uma cadeira de rodas automática, de forma que o paciente seja capaz de controlá-la, idealmente, direcionando seu olhar para o estímulo desejado.

Uma interface que emprega sinais coletados via SSVEP é denominada BCI-SSVEP. Em termos de seu estabelecimento, ela consiste de etapas importantes, conforme ilustrado na Figura 1, sendo estas: estágio de coleta que registra os sinais cerebrais, estágio de pré-processamento que filtra e extrai ruído do sinal gravado, estágio de extração que extrai informações importantes dos dados neurais, estágio de classificação que determina a saída de uma BCI através dos dados neurais processados e, por fim, estágio de feedback que apresenta a saída da BCI para o usuário (XU et al., 2023).

Figura 1 – Etapas de um sistema BCI com base no paradigma SSVEP.



Fonte: Adaptado de (COSTA, 2015).

Nesse contexto, BCIs baseadas em SSVEP, de acordo com (CECOTTI, 2010), são mais acessíveis do que outros sistemas BCI e apresentam algumas vantagens, como uma

alta taxa de transferência de informações (ITR do inglês, *high information transfer rate*). No entanto, estão vinculadas aos músculos extraoculares e nervos específicos para gerar potenciais visuais evocados, o que as torna dependentes desses fatores. Além disso, essas BCIs são frequentemente criticadas pelos estímulos visuais desconfortáveis que podem causar.

Devido ao processo de coleta de sinais se mostrar cansativo para alguns pacientes, ele se dá por breves sessões experimentais, e, por isso, a quantidade de amostras coletadas, especialmente se considerarmos a alta necessidade de ingestão de dados por modelos estado-da-arte em inteligência artificial, pode ser considerada insuficiente. Além disso, na grande maioria dos casos, existe uma relação direta entre a quantidade de dados e o desempenho de técnica de classificação, desde que estes sinais apresentem uma qualidade desejada. Com isso, muito tem se estudado a respeito de técnicas de data augmentation, de forma a expandir o número de amostras disponíveis para treinamento dos sistemas BCI-SVVEP.

Recentemente, modelos baseados em aprendizado profundo têm atraído atenção de pesquisadores devido à sua capacidade de gerar dados, em suas várias formas (imagens, áudio, texto, etc.), com resultados promissores. Neste contexto, podemos citar os modelos de difusão que, conforme explicado por (CROITORU et al., 2023), representam um tipo de modelo generativo profundo, consistindo em duas fases: a fase de difusão direta, na qual os dados de entrada são gradualmente perturbados pela adição de ruído Gaussiano; e a fase de difusão reversa, na qual um modelo generativo é treinado para recuperar os dados originais a partir dos dados perturbados, aprendendo a reverter o processo de difusão passo a passo.

Neste trabalho, implementamos o modelo de difusão para gerar dados sintéticos de BCI-SSVEP, além de abordar as etapas de processamento digital de sinais cerebrais necessárias para desenvolver uma interface cérebro-computador utilizando a técnica SSVEP.

1.1 Objetivo

Este trabalho tem como objetivo implementar uma sistema BCI-baseado em SSVEP e avaliar o impacto no desempenho do sistema quando se empregam de sinais cerebrais gerados artificialmente por modelo de difusão na sua etapa de treinamento.

1.2 Estado da arte

No contexto da BCI há algumas deficiências na obtenção de classificadores de alta performance. Isso se deve à complexidade cerebral variável entre os indivíduos e a escassez de dados disponíveis para treinamento do sistema. No trabalho de (AZNAN et al., 2019), os autores abordaram esse problema gerando dados sintéticos para aprimorar o sistema de

classificação SSVEP, aumentando o volume de dados para realizar o treinamento do modelo. Devido aos desafios envolvidos no registro de sinais cerebrais, eles utilizaram modelos generativos, tais como: Rede Generativa Adversarial (GAN), Rede Generativa Adversarial de Wasserstein (WGAN) e Codificador Auto-Variacional (VAE), os quais foram treinados com uma quantidade limitada de sinais cerebrais provenientes de diferentes indivíduos, para criar vetores sintéticos complementares. Esses vetores foram posteriormente empregados no treinamento de um classificador SSVEP, resultando em várias melhorias, incluindo um aumento de 35% na capacidade de generalização entre diferentes indivíduos.

Em (AZNAN et al., 2021), com o objetivo de aprimorar a precisão e eficiência dos sistemas de interface cérebro-computador em paradigma SSVEP, os autores propõem um novo método de geração de dados EEG sintéticos altamente realistas que seja independente dos indivíduos, sessões ou condições ambientais. Essa abordagem, denominada como Subject Invariant SSVEP Generative Adversarial Network (SIS-GAN), utiliza uma única rede para gerar dados de EEG sintéticos de várias classes de SSVEP. Além disso, ao utilizar uma rede de classificação, pré-treinada, com pesos fixos pode-se garantir que o modelo generativo seja insensível às características de cada indivíduo. Como consequência, foram obtidos dados sintéticos invariantes no que tange aos indivíduos possibilitando a aplicação do modelo a novos sujeitos que não foram observados anteriormente. Nessa avaliação experimental, ao realizar o treinamento com apenas dados sintéticos, os autores foram capazes de melhorar a generalização para sujeitos não vistos previamente ao realizar a classificação sem calibração em até 16%.

No estudo realizado por (KWON; IM, 2022), os autores propõem um novo método de tradução de sinais multidomínio baseado no StarGAN v2. O objetivo foi gerar sinais de potencial evocado visual de estado estacionário a partir de eletroencefalogramas em repouso. O modelo proposto, baseado no StarGAN, foi treinado com dados de EEG adquiridos de três participantes e obteve sucesso na geração de sinais SSVEP artificiais para 15 participantes de teste. Esses sinais foram gerados a partir de dados de EEG em repouso coletados durante um curto período de 16 segundos de cada participante de teste. Para avaliar o desempenho das BCIs baseadas em SSVEP, os autores investigaram o uso dos sinais SSVEP artificialmente gerados juntamente com o Combined-ECCA, uma versão estendida da análise de correlação canônica combinada (Combined-CCA), proposta neste estudo. Isso foi feito com o objetivo de verificar a eficácia dessas BCIs na tarefa proposta. Por fim, pode-se perceber que o desempenho do sistema em relação à precisão de classificação e ITR de BCIs baseadas em SSVEP foram significativamente melhorados com o uso das técnicas propostas.

No trabalho de (PAN; CHEN; ZHANG, 2023), os autores fazem uma revisão dos métodos de classificação baseados em aprendizado profundo (do inglês, *Deep Learning*) para SSVEP. Nesse contexto, considerando que o desempenho desses sistemas estão muito atrelados ao método de classificação, esse estudo tem como objetivo realizar uma revisão

do progresso dos métodos de classificação que são baseados em *deep learning*. Além disso, a técnica de aprendizado profundo possibilita uma alternativa para a classificação de dados em BCIs baseadas em SSVEP e tem recebido enorme visibilidade nos últimos anos. Esse estudo também aborda os modelos das Redes Generativas Adversárias para a criação de dados sintéticos.

No trabalho de (HARTMANN; SCHIRRMESTER; BALL, 2018), que também explora o uso das redes GAN, destaca-se que inicialmente essa metodologia era aplicada principalmente em aplicações generativas envolvendo imagens. No entanto, recentemente, começaram a ser utilizadas em dados de séries temporais, como é o caso dos sinais EEG. Nesse estudo, foi empregado um framework GAN para a geração de sinais EEG artificiais. Uma escolha feita nesse trabalho foi utilizar redes neurais convolucionais regulares, uma vez que a maioria dos estudos GAN utiliza a arquitetura DCGAN baseada em CNNs. Com o objetivo de obter sinais mais realistas, foi proposta uma melhoria no treinamento do Wasserstein GAN, a fim de tornar o processo mais estável. Além disso, diferentes métricas de avaliação foram comparadas e houve discussões sobre a escolha da arquitetura da rede para alcançar resultados satisfatórios.

2 Metodologia

Neste Capítulo são apresentados os detalhes de cada uma das etapas clássicas de uma BCI-SSVEP que foram implementadas: pré-processamento, extração de características, seleção e classificação. Também é apresentada a discussão teórica a respeito das redes neurais artificiais e dos modelos de difusão, que foram empregados para aprimorar o sistema BCI.

2.1 Base de dados

Neste trabalho utilizou-se um conjunto de dados disponível em (LIU et al., 2020). A base de dados conta com sinal cerebral de 35 voluntários obtido por meio de EEG com 64 eletrodos. Para cada indivíduo foram realizadas 6 coletas de 6 s, sendo os primeiros 500 ms de pré-estímulo e os últimos 500 ms de pós estímulo. A taxa de amostragem utilizada foi de 250 Hz e ao todo havia 40 estímulos visuais diferentes. Os estímulos visuais cintilavam em frequências fixas no intervalo de 8 Hz a 15.8 Hz, com incrementos de 0.2 Hz (por exemplo, 8 Hz, 8.2 Hz, 8.4 Hz, ..., 15.8 Hz).

2.2 Pré-processamento

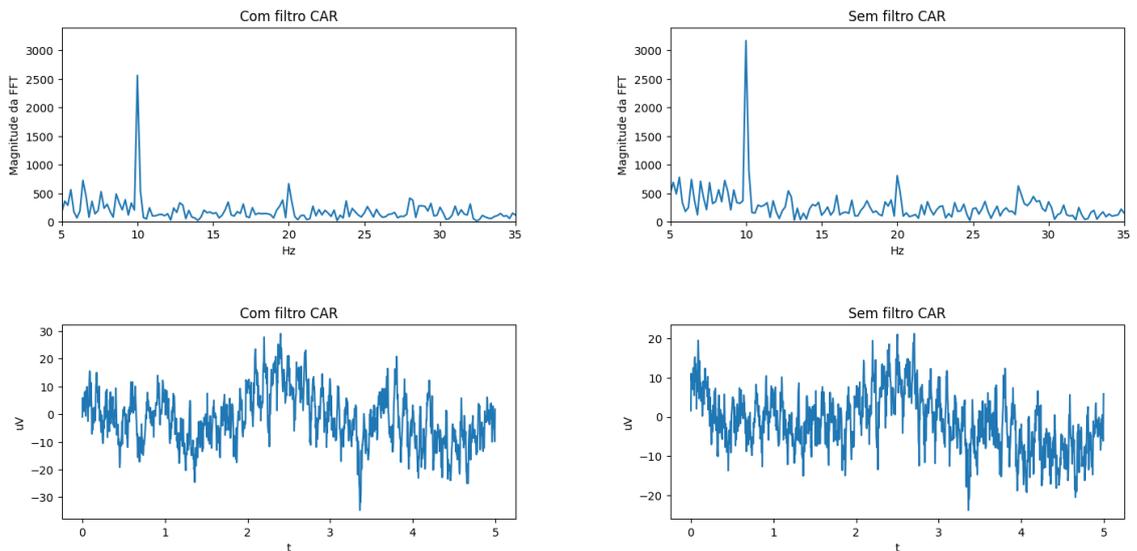
Durante a coleta de sinais de EEG, diversos artefatos podem comprometer a qualidade do sinal, como artefatos do equipamento, fontes externas e atividade elétrica do corpo. Para melhorar a detecção de padrões SSVEP e suprimir ruídos, são utilizadas técnicas de pré-processamento, como a filtragem espacial CAR (do inglês, *Common Average Reference*), que tem como objetivo reduzir os componentes que estão presente em vários eletrodos. Essas técnicas visam ampliar a relação sinal-ruído, tornando a implementação de interfaces cérebro-computador mais eficaz (SIULY; LI; ZHANG, 2016).

A técnica de filtragem CAR consiste em subtrair, amostra por amostra, o valor médio do potencial calculado para todos os eletrodos do potencial medido de cada eletrodo. A equação deste filtro é representada por:

$$V_i^{CAR} = V_i^{eletrodo} - \frac{1}{n} \sum_{j=1}^n V_j^{eletrodo} \quad (2.1)$$

sendo n o número de eletrodos utilizados (neste caso 64), $V_i^{eletrodo}$ o potencial medido no eletrodo de número i e V_i^{CAR} o potencial do eletrodo i após a aplicação da filtragem espacial CAR.

Figura 2 – Representação no domínio do tempo e da frequência do sinal coletado em Oz - Estímulo visual em 10 Hz.



Fonte: Do autor.

Na Figura 2, percebe-se o efeito do filtro CAR no domínio da frequência (gráficos superiores) e no domínio do tempo (gráficos inferiores). Nota-se que o pico de 10 Hz fica melhor evidenciado, no espectro filtrado, uma vez que as demais frequências foram atenuadas.

Após o processo da filtragem, antes de realizar a extração de características, foi realizado o janelamento de 1 s no sinal. Sendo assim, pode-se abstrair que em tempo real a estimulação ocorreria durante o período de 1 s. Como cada coleta tem 5 s de sinal com potencial evocado, foram geradas 5 janelas por trial, resultando num total de 30 janelas de sinal EEG para cada estímulo visual.

2.3 Extração de características

Com os sinais já previamente pré-processados, o próximo passo corresponde à extração de características. No contexto das BCIs, a extração de características está relacionada ao paradigma utilizado (DORNHEGE et al., 2007). Neste trabalho, considerando a técnica SSVEP, utiliza-se técnicas cujo objetivo é a extração de dados espectrais, como o cálculo da densidade espectral de potência próximo às frequências evocadas (LEITE, 2016). Dessa forma, esta etapa tem como intuito realizar o mapeamento do sinal de entrada possibilitando sua representação em um novo espaço e facilitando, conseqüentemente, a etapa de classificação.

O método escolhido foi a Transformada Rápida de Fourier (FFT), que permite

representar uma função como uma combinação de componentes periódicos e, ao mesmo tempo, possibilita recuperar o sinal original a partir dessas componentes (HECKBERT, 1995). A FFT realiza o mapeamento do sinal $x(n)$ do domínio do tempo para o domínio da frequência, conforme mostra a equação:

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n] \quad (2.2)$$

sendo que N representa o comprimento do sinal $x(n)$ e $k = 0, \dots, N-1$. Essa transformação pode ser realizada por meio da seguinte equação:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi j \frac{kn}{N}} y[k] \quad (2.3)$$

A Transformada de Fourier $y[k]$ fornece valores complexos relacionados à magnitude e fase de cada ponto no tempo e frequência. Dessa forma, a magnitude da FFT para determinada frequência auxilia na indicação da frequência na composição do sinal $x[n]$ (OPPENHEM, 1999). Neste trabalho, a FFT foi calculada para quatro frequências (8, 10, 12 e 15 Hz).

2.4 Classificador

No contexto da BCI-SSVEP, o classificador tem como objetivo realizar a distinção das frequências evocadas conforme as características previamente obtidas na etapa anterior. Isso equivale a associar corretamente cada classe de frequência do estímulo visual escolhida pelo indivíduo.

2.4.1 Classificador linear

Segundo (KOUTROUMBAS; THEODORIDIS, 2008), a maior vantagem dos classificadores lineares é sua simplicidade e sua vantagem computacional. Neste trabalho, dentre os classificadores lineares existentes, utilizou-se o Método dos Mínimos Quadrados.

Inicialmente, pode-se descrever os classificadores lineares da seguinte forma:

$$Y = \omega^T X \quad (2.4)$$

em que Y representa uma saída, ω é o vetor de parâmetros do classificador e X equivale a matriz que contém as características que foram extraídas do sinal.

O intuito desse método é minimizar o erro quadrático entre os valores desejados e o real valor de saída do classificador. A função custo pode ser definida como:

$$J(w) = \sum_{i=1}^N (y_i - x_i^T w)^2 = \sum_{i=1}^N e_i^2 \quad (2.5)$$

A equação 2.5 pode ser simplificada e assim:

$$\sum_{i=1}^N x_i(y_i - x_i^T w) = 0 \rightarrow \left(\sum_{i=1}^N x_i x_i^T\right) w = \sum_{i=1}^N (x_i y_i) \quad (2.6)$$

A equação 2.6 pode ser reescrita como:

$$(X^T X)w = X^T y \rightarrow (X^T X)^{-1} X^T y \quad (2.7)$$

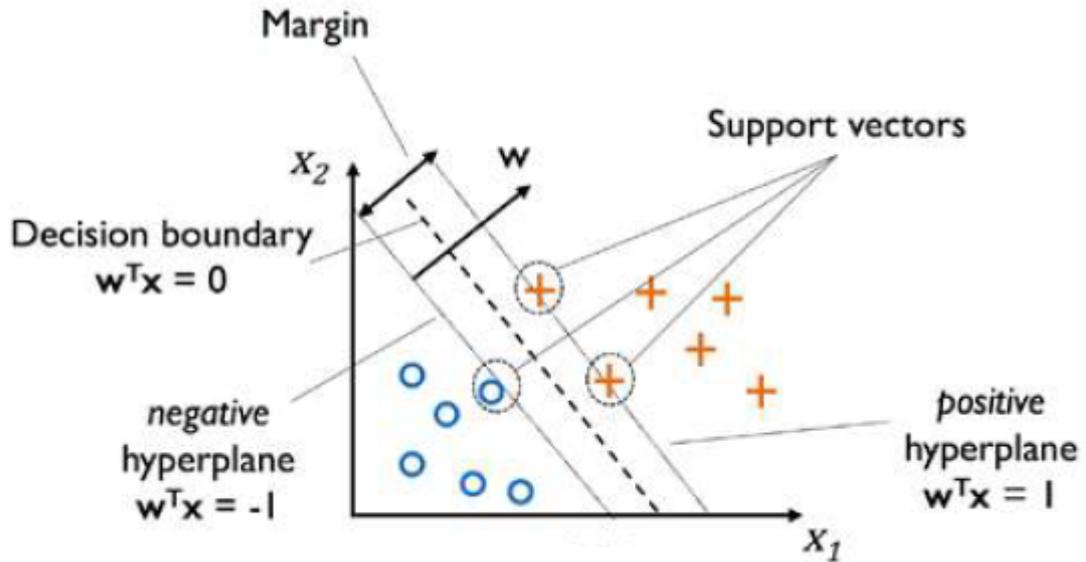
sendo a operação $(X^T X)^{-1}$ denominada como a pseudo-inversa da matriz X e o produto $(X^T X)$ corresponde a correlação amostral de X .

Nessa abordagem de extração de atributos na forma matricial, há a formação de blocos por frequência/classe, onde cada bloco possui $N_j \times N_t$ amostras, com dimensão $N_c \times N_f$, ou seja, para cada uma das $N_j \times N_t$ amostras do bloco, concatena-se os coeficientes da FFT referentes às N_f classes de cada N_c eletrodo. De forma geral, concatenando os blocos referentes as N_f frequências, forma-se, então, uma matriz de atributos $(N_f \times N_j \times N_t) \times (N_c \times N_f)$. No caso deste trabalho, como estamos empregando $N_f = 4$, $N_j = 5$, $N_t = 6$ e $N_c = 1$, a dimensão da referida matriz é (120×4) . Tem-se que N_j equivale ao número de janelas, N_t ao número de testes, N_c ao número de canais e N_f ao número de frequências. A esta matriz, aplica-se, subsequentemente, a divisão equilibrada das amostras em conjuntos de treinamento e teste, seguindo a abordagem de *holdout*, com proporção de 80/20.

2.4.2 Support vector machine

Segundo (RASCHKA; MIRJALILI, 2019), a máquina de vetor de suporte (SVM, do inglês: Support Vector Machine) tem como objetivo maximizar a margem, definida como a distância entre o hiperplano de separação (fronteira de decisão) e os exemplos de treinamento que estão mais próximos desse hiperplano, denominados como vetores de suporte.

Figura 3 – SVM - Maximizando a margem.



Fonte: Adaptado de (RASCHKA; MIRJALILI, 2019).

Intuitivamente, percebe-se que quando as fronteiras de decisão (do inglês, Decision boundary) possuem margens mais amplas, tende a ocorrer um erro de generalização menor. Por outro lado, modelos com margens pequenas são mais propensos a se ajustarem de forma excessiva aos dados de treinamento (Overfitting). Os hiperplanos positivos e negativos, paralelos à fronteira de decisão podem ser expressos da seguinte forma:

$$w_0 + w^t x_{pos} = 1 \quad (2.8)$$

$$w_0 + w^t x_{neg} = -1 \quad (2.9)$$

Subtraindo as equações 2.8 e 2.9 tem-se que:

$$w^t (x_{pos} - x_{neg}) = 2 \quad (2.10)$$

Pode-se normalizar a equação 2.10 por meio do comprimento do vetor w , definido como:

$$\|w\| = \sqrt{\sum_{j=1}^m w_j^2} \quad (2.11)$$

Dessa forma, obtem-se a seguinte equação:

$$\frac{w^T (x_{pos} - x_{neg})}{\|w\|} = \frac{2}{\|w\|} \quad (2.12)$$

O lado esquerdo da equação 2.12 pode ser interpretado como a distância entre os hiperplanos positivo e negativo, ou seja, equivale a margem que se deseja otimizar. Considerando esta mesma equação, a SVM tem como objetivo maximizar a margem por meio da maximização do termo $\frac{2}{\|w\|}$ sob a condição de que os exemplos sejam classificados de maneira correta. Esses exemplos são detalhados a seguir:

$$w_0 + w^T x^{(i)} \geq 1 \text{ if } y^{(i)} = 1 \text{ para } i = 1 \dots N \quad (2.13)$$

$$w_0 + w^T x^{(i)} \leq -1 \text{ if } y^{(i)} = -1 \text{ para } i = 1 \dots N \quad (2.14)$$

Nas equações 2.13 e 2.14, N se refere ao número de exemplos do conjunto de dados. Essas equações indicam que todos os exemplos da classe negativa devem estar de um lado do hiperplano negativo, enquanto todos os exemplos da classe positiva devem estar atrás do hiperplano positivo. De forma mais concisa, isso pode ser expresso da seguinte maneira:

$$y^{(i)}(w_0 + w^T x^{(i)}) \geq 1 \text{ Para todo } i \quad (2.15)$$

No caso do SVM, em vez de aplicarmos a etapa de extração de atributos, o que resultaria, conseqüentemente, em uma redução da cardinalidade do problema, todas as amostras do sinal foram consideradas em versões temporais. Em outras palavras, a dimensão do espaço de atributos correspondeu a 250.

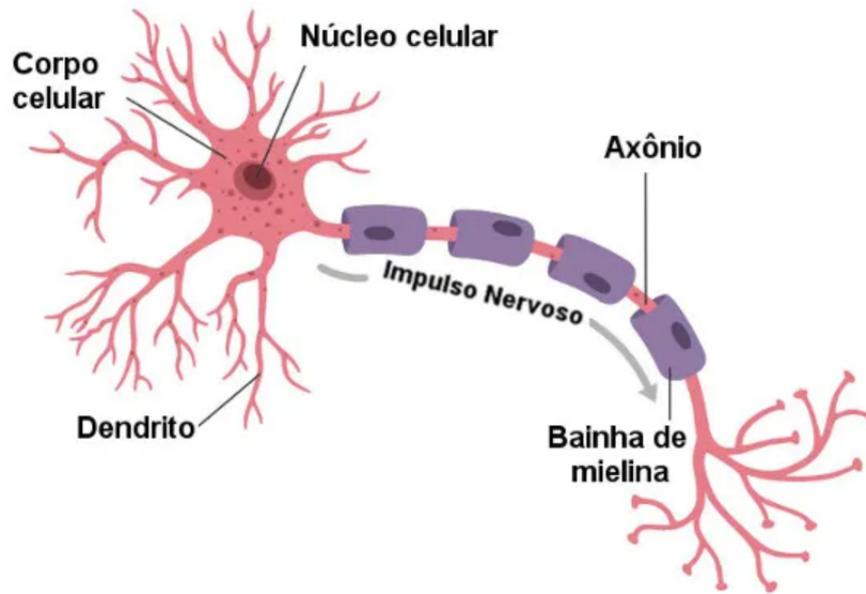
2.5 Redes neurais artificiais

As redes neurais possuem uma habilidade fundamental: a capacidade de aprender padrões e representações a partir de dados. Essa capacidade as torna incrivelmente poderosas ao executar tarefas de aprendizado de máquina. As redes neurais são formadas por nós interligados, chamados neurônios, que têm a capacidade de processar e modificar os dados de entrada usando camadas de computação (GURNEY, 2018).

Os modelos de redes neurais foram inspirados na funcionalidade de um cérebro humano, em que bilhões de neurônios interconectados entre si processam informações paralelamente (WANG, 2003). Neurônios biológicos são células nervosas interconectadas no cérebro que estão envolvidas no processamento e transmissão de sinais químicos e elétricos (RASCHKA; MIRJALILI, 2019). De forma simplificada, os neurônios, são constituídos por três partes, conforme Figura 4:

1. Dendritos: Região responsável por captar informações e por realizar o transporte desta informação para o corpo celular.
2. Corpo celular: Processa as informações que são transmitidas a outros neurônios através do axônio.

Figura 4 – Neurônio biológico.

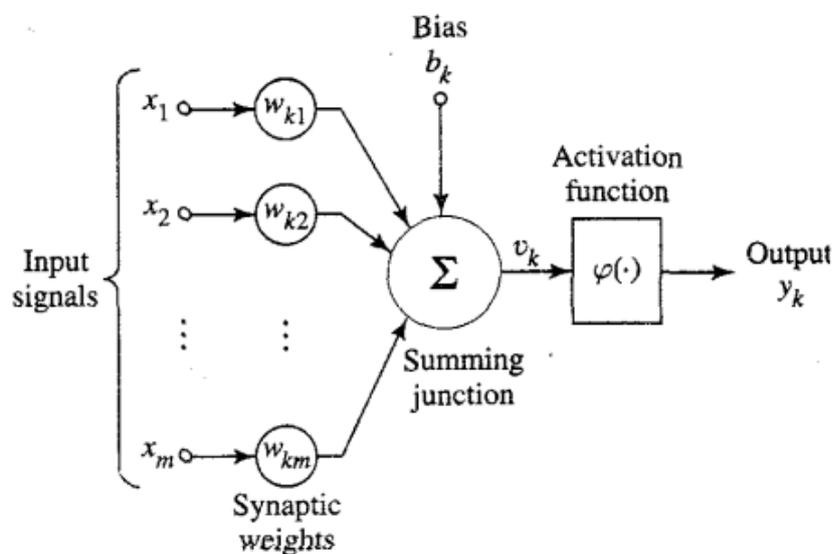


Fonte: Adaptado de (BORGES et al., 2015).

3. Axônio: Realiza a distribuição das informações que foram processadas até o dendrito de outros neurônios.

O neurônio artificial pode ser representado conforme Figura 5, denominado também como Perceptron (SANGER; BALJEKAR, 1958).

Figura 5 – Neurônio artificial.



Fonte: (HAYKIN, 1999)

Na Figura 5, o vetor $x = [x_1, \dots, x_m]^T$ representa os sinais de entrada do neurônio artificial. Já w representa os pesos sinápticos em relação as entradas e são ajustados conforme as iterações que ocorrem no período de treinamento. v_k representa a combinação dos pesos e entradas. O bias b_k possibilita um ajuste no grau de liberdade da rede. Assim, este limiar é adicionado a função de ativação que é realizada por meio de uma soma ponderada. Esta função tem como objetivo realizar a adequação da saída do neurônio em uma amplitude estabelecida previamente. Portanto, a saída do perceptron pode ser denominada como y (HAYKIN, 1999).

Matematicamente, o neurônio pode ser representado pelo seguinte par de equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.16)$$

e

$$y_k = \varphi(u_k + b_k) \quad (2.17)$$

Onde x_1, x_2, \dots, x_m representam os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ; u_k é a combinação linear da saída devido aos sinais de entrada; b_k representa o bias; φ é a função de ativação e y_k é a saída do neurônio. Como visto, o ajuste do bias surte efeito sobre a saída u_k da combinação linear do modelo da Figura 5. Nesse contexto, tem-se que:

$$v_k = u_k + b_k \quad (2.18)$$

Na Equação (2.17), o bias b_k é um parâmetro externo do neurônio artificial k . De forma equivalente, pode-se obter uma equação equivalente por meio das equações (2.16) e (2.18). Assim, é possível obter o seguinte par de equação:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.19)$$

e

$$y_k \approx \varphi(v_k) \quad (2.20)$$

A função de ativação, denotada por $\varphi(v)$ defini a saída do neurônio pelo potencial de ativação v_k do neurônio k . A função de ativação permite estabelecer o momento de disparo do neurônio. Na Figura 6 pode-se observar várias dessas funções de ativação. A escolha dessas funções está atrelada ao problema a ser resolvido.

Figura 6 – Funções de ativação.

Função de ativação	Gráfico	Equação	Função
Linear		$f(x) = x$	$f'(x) = 1$
Degrau		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Sigmoide		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
Softmax		$f(x) = \ln(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$
ReLU		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Leaky ReLU		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
PReLU		$f(\alpha, x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(\alpha, x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
ELU		$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$	$f'(\alpha, x) = \begin{cases} f(\alpha, x) + \alpha & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$

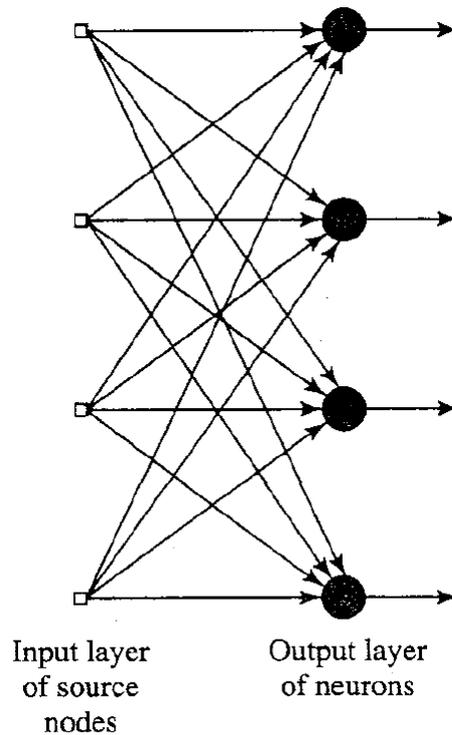
Fonte: (LEITE, 2021)

Conforme mencionado por (GÉRON, 2022), a função de ativação mais amplamente utilizada em redes neurais artificiais atualmente é a *ReLU*. Quanto à camada de saída, a seleção da função está associada à tarefa em questão. Portanto, em problemas de classificação multi-classe, a função *softmax* é empregada. Essa função gera uma distribuição de probabilidade que indica a pertinência de cada classe.

Quanto à arquitetura de uma rede neural, ela é determinada pela quantidade de neurônios e pela maneira como estão organizados e conectados. Em geral, uma rede é composta por uma camada de entrada, que captura informações externas; uma camada intermediária, responsável por extrair padrões em relação ao processo/sistema e uma camada de saída que representa as saídas finais da rede como resultado do processamento das camadas anteriores (GURNEY, 2018).

Segundo (HAYKIN, 1999), as redes podem ser classificadas em três classes, sendo elas, redes *feedforward* com camada única, *feedforward* com múltiplas camadas e redes recorrentes. A rede *feedforward* de camada simples, ilustrada na Figura 7, possui apenas uma camada de entrada de nós de fonte que se projetam para camada de saída de neurônios subsequente.

A rede *feedforward* com múltiplas camadas, em contraste com a anterior, contém uma ou mais camadas intermediárias entre a entrada e a saída, conforme ilustrado na Figura 8. Nessa rede, a camada de entrada tem a função exclusiva de receber os dados de

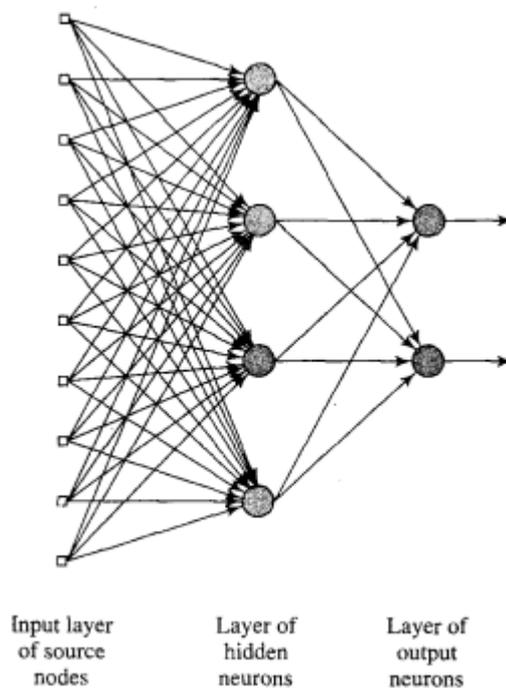
Figura 7 – Rede *feedforward* de camada única.

Fonte: (HAYKIN, 1999)

entrada e transferir seus atributos para os primeiros neurônios da camada escondida. As camadas escondidas, por sua vez, executam mapeamentos mais complexos entre a camada de entrada e a de saída. Ademais, a quantidade de camadas e o número de neurônios na rede são definidos de maneira experimental.

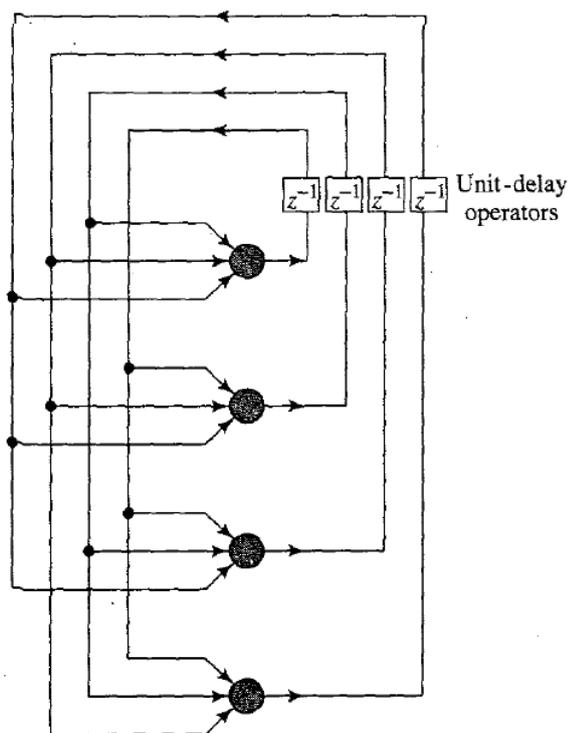
Por fim, em uma rede do tipo recorrente, os neurônios de saída realimentam a entrada de outros neurônios. A Figura 9 ilustra essa característica.

Figura 8 – Rede *feedforward* de múltipla camada.



Fonte: (HAYKIN, 1999)

Figura 9 – Rede recorrente.



Fonte: (HAYKIN, 1999)

O treinamento das redes neurais tem como objetivo torná-las capazes de aprender com os dados, tendo como consequência a melhoria do seu desempenho ao realizar tarefas específicas. Dessa forma, esse processo permite que a rede consiga generalizar sua compreensão além dos dados de treinamento e consiga realizar previsões de forma precisa em novos dados não vistos anteriormente.

Assim, segundo (HAGAN et al., 2014), em um aprendizado supervisionado, a regra de aprendizado é fornecida por meio de um conjunto de exemplos (o conjunto de treinamento) de comportamento adequado da rede, onde (2.21), p_q é a entrada da rede e t_q é seu correspondente correto.

$$p_1, t_1, p_2, t_2, \dots, p_Q, t_Q, \quad (2.21)$$

Quanto ao aprendizado não supervisionado, os pesos e viés são modificados em resposta apenas às entradas da rede. Não há saídas alvo disponíveis. Os algoritmos de treinamento desse tipo de rede realizam operações de agrupamento e aprendem a categorizar padrões da entrada em classes. (HAGAN et al., 2014)

Sendo assim, uma forma bem conhecida de realizar o treinamento da rede é através do algoritmo de *backpropagation*, introduzido inicialmente por (RUMELHART; MCCLELLAND; GROUP, 1986), sendo um procedimento de aprendizagem para redes *feedforward* de múltiplas camadas (MLP, do inglês *Multi-layer Perceptron*).

Este método ocorre por meio de ajustes iterativos dos pesos da rede, a fim de que as diferenças entre a saída atual e a desejada sejam minimizadas. Nesse processo, um vetor de entrada é repassado à rede e percorre por toda a rede com o intuito de obter o valor de saída. Assim, esse vetor de saída é comparado ao vetor de saída desejado, produzindo um vetor de erro. Com isso, o vetor de erro é retropropagado pela rede, possibilitando o ajuste dos pesos. Esse processo iterativo ocorre até que a rede retorne para os valores de entrada valores de saída que sejam realmente próximos dos valores desejados (RUMELHART; MCCLELLAND; GROUP, 1986).

O ajuste dos pesos na iteração ocorre de acordo com a equação:

$$w \leftarrow w - \eta \frac{\partial E}{\partial w} \quad (2.22)$$

Na equação 2.22, tem-se o peso da iteração atual subtraído do peso da iteração anterior, corrigida pelo valor proporcional ao gradiente. O parâmetro $\frac{\partial E}{\partial w}$ dessa expressão é de suma importância visto que realiza os cálculos das derivadas parciais da função de erro E em relação a cada peso correspondente ao vetor w .

As redes neurais, então, permitem receber sinais brutos da SSVEP como entrada, eliminando a necessidade de extração de características realizadas de forma manual, assim como as etapas de pré-processamento do sinal. Com isso, possibilitam uma vantagem crítica, visto que permitem que sinais ou características implícitas de EEG não sejam

perdidos na etapa de pré-processamento ou extração de características (XU et al., 2023; LAWHERN et al., 2018).

2.5.1 Autoencoders variacionais

Antes de nos aprofundarmos nos autoencoders variacionais, vamos revisitar brevemente o conceito de autoencoders. Autoencoders consistem, basicamente, de um codificador \mathcal{E} , que mapeia os dados de entrada x para uma representação de espaço latente z , e um decodificador \mathcal{D} que reconstrói a entrada a partir da representação latente:

$$z = \mathcal{E}(x) \quad (2.23)$$

$$\hat{x} = \mathcal{D}(z) \quad (2.24)$$

Em uma extensão da versão tradicional do autoencoder, o autoencoder variacional adiciona uma interpretação probabilística do espaço latente. Isso significa que, em vez de existir um mapeamento direto e preciso entre dados de entrada e representação latente, o encoder lança mão do que é conhecido como truque da reparametrização. Nesse truque, os dados de entrada são mapeados em uma distribuição de probabilidade, $q(z|x)$, geralmente representada por uma distribuição gaussiana $p(z)$, de forma que valores de média e desvio padrão correspondam ao mapeamento de destino. Subsequentemente, o decoder recebe versões amostradas da distribuição definida pelo encoder, $z \sim q(z|x)$, de forma que ele atue de forma geracional. Desta forma, há uma proposta de regularização do espaço latente. Em sua saída, então, o decoder produz uma versão reconstruída dos dados de entrada, $p(x|z)$, e, de acordo com a função perda, representada a seguir, os pesos das duas redes são ajustados simultaneamente.

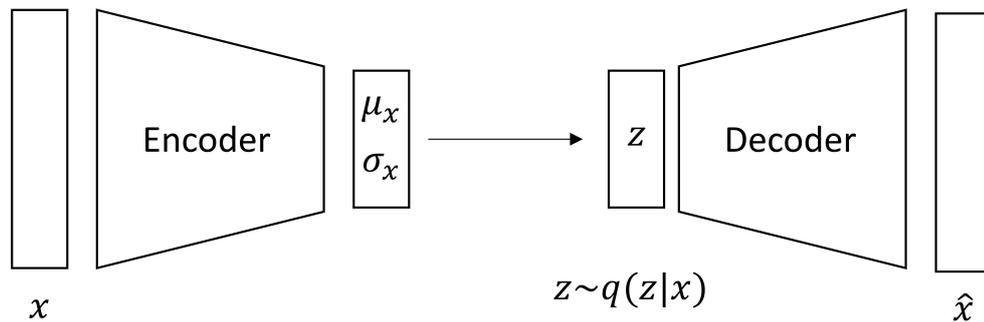
$$\mathcal{L} = E_{q(z|x)}[\log p(x|z)] - KL(q(z|x)||p(z)) \quad (2.25)$$

onde o termo KL representa a Divergência de Kullback-Leibler.

Neste trabalho, tanto o encoder quanto o decoder, possuem acesso aos rótulos dos dados, de forma que a arquitetura empregada seja considerada um autoencoder variacional condicional, conforme ilustrado na Figura 11. Com isso, nas duas redes, os dados de entrada correspondentes são concatenados às representações vetoriais que representam cada uma das quatro classes envolvidas no problemas – os quatro estímulos cerebrais avaliados. Em termos matemáticos, a função perda supracitada sofre uma ligeira modificação, com a adição da contribuição do vetor de rótulos y referentes às classes, conforme indiciado abaixo:

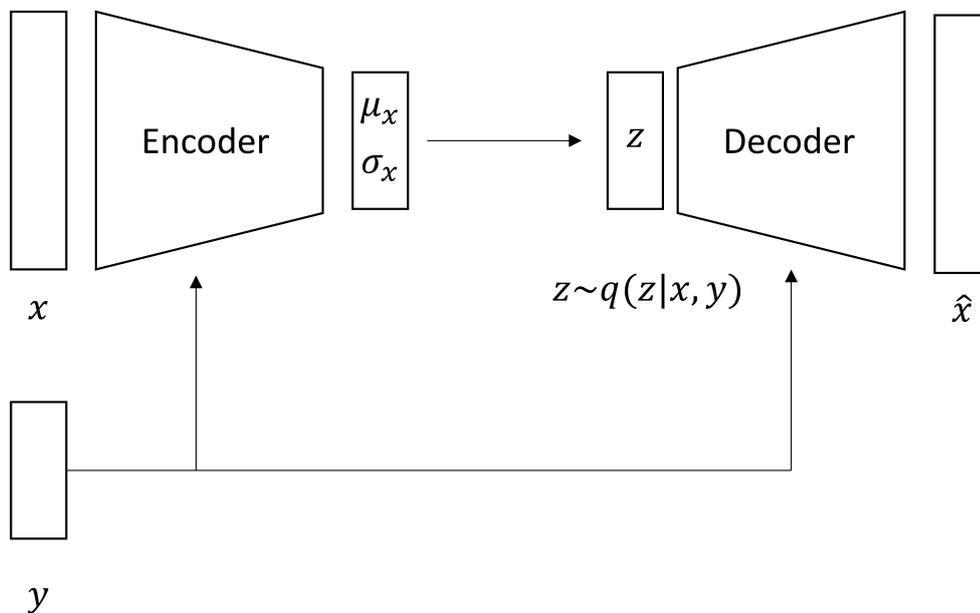
$$\mathcal{L} = E_{q(z|x,y)}[\log p(x|z,y)] - KL(q(z|x,y)||p(z)) \quad (2.26)$$

Figura 10 – Autoencoder variacional.



Fonte: Do autor.

Figura 11 – Autoencoder variacional condicional.



Fonte: Do autor.

Em termos de arquitetura, por se tratar de uma versão condicional, o encoder recebe a concatenação dos dados de entrada, com dimensão 250, e a parcela referente a label, que possui dimensão 4. Com isso, as camadas desta rede são responsáveis por fazer o seguinte processamento: $(250 + 4) \rightarrow 512 \rightarrow 256 \rightarrow 64$, onde a última camada representa o tamanho do espaço latente. O decoder, por sua vez, receberá esta dimensão latente juntamente com a parcela de label, com dimensão também igual a 4. Assim, o processamento se dá: $(64 + 4) \rightarrow 512 \rightarrow 256 \rightarrow 250$. Ambas as redes foram treinadas com otimizador Adam, com *learning rate* igual a 0,00001, por 100 épocas, com tamanho de *batch* de 8 amostras.

2.5.2 Modelo de difusão

Segundo (CAO et al., 2022), modelos generativos profundos são uma abordagem que possibilitam a geração de dados de alta qualidade em várias disciplinas. O modelo de difusão, uma nova classe que surgiu dos modelos generativos profundos, tem sido discutido recorrentemente devido a sua ótima qualidade na geração de amostras realistas. Este modelo tem recebido muita atenção pelo fato de superar os desafios enfrentados por modelos generativos profundos como Autoencoder variacionais (VAE, do inglês *Variational Autoencoders*), Modelos baseados em energia (EBM, do inglês *Energy-Based Models*) e GAN's.

Os modelos de difusão são constituídos de dois processos que são interconectados. Primeiro, em um processo avançado pré-definido que transforma a distribuição dos dados em uma distribuição anterior mais simples, como uma Gaussiana. O segundo processo ocorre de forma reversa e utiliza uma rede neural treinada para desfazer de forma progressiva o processo avançado por meio de simulações de equações ordinárias ou estocásticas (ODE/SDE, do inglês *Ordinary or Stochastic Differential Equations*). Como consequência, o modelo de difusão proporciona um objetivo de treinamento mais estável quando se compara com o objetivo adversarial nas redes GAN's. O modelo de difusão, apesar de ser superior na geração de dados em comparação com VAE, EBM e fluxos de normalização, sofrem naturalmente um processo de amostragem mais demorado em comparação com VAE ou GAN. Isso ocorre devido à transformação iterativa da distribuição anterior em uma distribuição complexa de dados por meio de processos ODE/SDE/Markov, que exige um grande número de avaliações de funções durante o processo reverso.

Neste trabalho, o modelo de difusão implementado foi baseado no modelo probabilístico (DDPM, do inglês *Denoising Diffusion Probabilistic Models*) inicializado por (SOHL-DICKSTEIN et al., 2015) e então proposto por (HO; JAIN; ABBEEL, 2020).

2.5.2.1 Processo de difusão

De forma resumida, o modelo de difusão insere gradualmente um ruído gaussiano em uma imagem de entrada x_0 , por meio de uma série de T etapas. Isso é denominado como processo direto e não está relacionado ao processo forward de uma rede neural. Esta etapa é necessária para gerar os alvos para a rede neural. Após isso, uma rede neural é treinada para recuperar os dados originais revertendo o processo de adição de ruído. Ao ser capaz de modelar o processo reverso, pode-se gerar novos dados e isso é chamado de processo reverso de difusão ou, de maneira geral, o processo de amostragem de um modelo generativo.

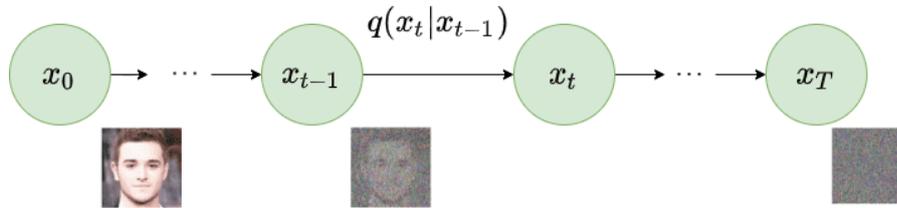
2.5.2.2 Processo direto

Os modelos de difusão podem ser interpretados como modelos com variáveis não observadas, conhecidas como latentes. O termo "latente" refere-se a um espaço de características ocultas contínuas. Isso os aproxima dos autoencoders variacionais. Na prática, esses modelos são estruturados usando uma cadeia de Markov com várias etapas. Aqui, "cadeia de Markov" implica que cada passo depende apenas do passo anterior, o que é uma suposição razoável. É importante ressaltar que não há uma restrição para a escolha de um tipo específico de rede neural, ao contrário do que ocorre nos modelos baseados em fluxo.

Dado uma entrada individual x_0 amostrada de um conjunto de dados de uma distribuição real de dados $q(x)(x_0 q(x))$, pode-se definir o processo de difusão direta por meio da adição de ruído. Em cada etapa da cadeia de Markov, adiciona-se ruído gaussiano com variância β_t a x_{t-1} produzindo uma nova variável latente x_t , com distribuição $q(x_t|x_{t-1})$. Conforme (HO; JAIN; ABBEEL, 2020), esse processo pode ser expresso matematicamente de acordo com a equação 2.27:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \mu_t = \sqrt{1 - \beta_t}x_{t-1}, \Sigma_t = \beta_t I) \quad (2.27)$$

Figura 12 – Etapa *forward* da difusão.



Fonte: (HO; JAIN; ABBEEL, 2020)

Em 2.27, considerando um cenário multi-dimensional, I é a matriz identidade, indicando que cada dimensão tem o mesmo desvio padrão β_t . Observa-se também que $q(x_t|x_{t-1})$ ainda é uma distribuição normal, definida pela média μ e pela variância Σ onde $\mu_t = \sqrt{1 - \beta_t}x_{t-1}$ e $\Sigma_t = \beta_t I$. O somatório Σ corresponderá sempre a matriz diagonal de variâncias, sendo aqui equivalente a β_t

Sendo assim, uma transformação tratável pode ser aplicada para mover os dados iniciais de entrada x_0 até o ponto final de dados x_t de forma fechada. Matematicamente, isso é definido como a probabilidade posterior, ou seja, a probabilidade condicional de um evento ou hipótese ocorrer após a consideração de evidências ou informações anteriores conforme a equação 2.28.

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (2.28)$$

Na equação 2.28, o símbolo ":", no termo à esquerda, indica que foi aplicado o operador q repetidamente do passo de tempo 1 até T . Isso é denominado como trajetória. No entanto, a fim de melhorar o tempo de amostragem aplica-se um truque de reparametrização (do inglês, reparametrization trick). Em modelos de difusão, que descrevem a evolução de dados ao longo do tempo, as etapas de propagação envolvem a introdução controlada de ruído nos dados. A reparametrização é uma estratégia que permite transformar a amostragem estocástica em um processo determinístico, tornando-o diferenciável e compatível com a propagação de gradientes (KINGMA; WELLING, 2013).

Dessa maneira, considerando $\alpha_t = 1 - \beta_t$, $\bar{\alpha} = \prod_{s=0}^t \alpha_s$ em que $\epsilon_0, \dots, \epsilon_{t-2}, \epsilon_{t-1} \sim N(\mathbf{0}, \mathbf{I})$, de forma recursiva, pode-se provar que:

$$x_t = \sqrt{1 - \beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} \quad (2.29)$$

$$= \sqrt{\alpha_t}x_{t-2} + \sqrt{1 - \alpha_t}\epsilon_{t-2} \quad (2.30)$$

...

$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_0 \quad (2.31)$$

Como forma de simplificação, utiliza-se o parâmetro ϵ , visto que todos os passos de tempo possuem o mesmo ruído gaussiano. Portanto, a fim de produzir uma amostra x_t , pode-se usar a seguinte distribuição:

$$x_t \sim q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0; (1 - \bar{\alpha}_t)\mathbf{I}) \quad (2.32)$$

Visto que β_t é um hiper parâmetro, pode-se pré-calcular α_t e $\bar{\alpha}_t$ para todos os passos temporais. Essas manipulações demonstram que o ruído poder ser amostrado em qualquer passo de tempo t e que é possível obter x_t de uma vez. Logo, conclui-se que é possível amostrar a variável latente x_t em qualquer passo de tempo arbitrário.

2.5.2.3 Variância

O parâmetro de variância β_t pode ser fixado como uma constante ou escolhido para ser como um cronograma ao longo dos T passos de tempo. Pode-se definir um cronograma de variância, podendo ser linear, quadrático, cosseno etc. Em (NICHOL; DHARIWAL, 2021), o autor demonstra a eficácia do uso do cosseno, como ilustrado na figura 13.

2.5.2.4 Processo reverso

A medida que T tende ao infinito, a variável latente x_t é próxima de uma distribuição de probabilidade gaussiana, também conhecida como distribuição normal, que é isotrópica. Dessa forma, com o aprendizado do processo de distribuição reversa $q(x_{t-1}|x_t)$, é possível

Figura 13 – Amostras latentes de cronogramas lineares (superior) e cosseno (inferior).



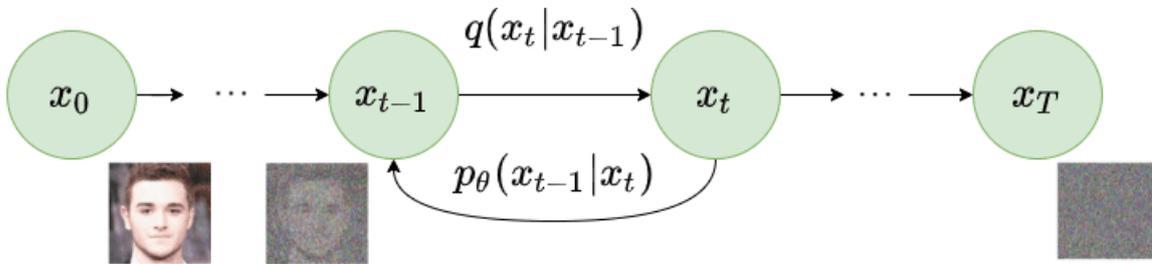
Fonte: (NICHOL; DHARIWAL, 2021)

amostrar x_t a partir de $\mathcal{N}(0, \mathbf{I})$, executar o processo reverso e obter uma amostra de $q(x_0)$, gerando um novo ponto de dados a partir da distribuição original dos dados.

Primeiramente, pode-se aproxima o processo reverso por meio de uma rede neural. Isso pode ser feito aproximando $q(x_{t-1}|x_t)$ com um modelo de rede neural parametrizada p_θ . Tendo em vista que $q(x_{t-1}|x_t)$ também será gaussiana, para valores suficientemente pequenos de β_t , pode-se escolher a distribuição p_θ de forma que também seja gaussiana e assim, parametrizar apenas a média e a variância. A equação 2.33 define isso.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sum_{\theta} \theta(x_t, t)\right) \quad (2.33)$$

Figura 14 – Processo reverso da difusão.



Fonte: (HO; JAIN; ABBEEL, 2020)

Aplicando a fórmula reversa para todos os passos de tempo $p_\theta(x_{0:T})$, denominado anteriormente como trajetória, pode-se ir de x_T a distribuição de dados:

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad (2.34)$$

Ao condicionar o modelo nos passos de tempo t , ele aprenderá a prever os parâmetros gaussianos, ou seja, a média $\mu_\theta(x_t, t)$ e a matriz de covariância $\sum_{\theta} \theta(x_t, t)$ para cada intervalo de tempo.

2.5.2.5 Treinamento do modelo de difusão

No treinamento do modelo de difusão, o objetivo é minimizar a função de perda, a qual é representada pela soma dos quadrados mínimos entre a predição do ruído e o ruído propriamente dito (KOTELNIKOV et al., 2023). A equação 2.35 define que:

$$L_t^{simple} = E_{x_0, \epsilon, t} \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \quad (2.35)$$

Assim como no autoencoder, o modelo de difusão recebeu informação da labels referentes aos sinais, de forma a ser possível escolher previamente a classe cujos sinais serão gerados e, com isso, apenas um modelo é treinado com o os sinais referentes a todas as frequências envolvidas. A equação acima é, então, reescrita como:

$$L_t^{simple} = E_{x_0, \epsilon, t} \|\epsilon - \epsilon_\theta(x_t, t, y)\|_2^2 \quad (2.36)$$

A minimização da equação (2.36) se deu por 100 épocas de treinamento por uma arquitetura composta pela seguinte configuração de camadas: $(64 + 4) \rightarrow 500 \rightarrow 1000 \rightarrow 500 \rightarrow 250 \rightarrow 64$. O otimizador foi o Adam com *learning rate* igual a 0,0001. Neste ponto, é importante destacar que, antes da amostra ser apresentada diretamente à rede difusora, ela é codificada pelo encoder previamente treinado, de forma a existir um mapeamento entre as dimensões 250 e 64.

Dada a convergência do modelo durante o treinamento, o processo consiste em, a partir de amostras aleatórias extraídas de uma distribuição normal e, de acordo com o processo reverso de difusão, aplicar iterativamente a equação (2.37):

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t, y) \right) + \sigma_t z \quad (2.37)$$

Como a saída do difusor possui dimensão 64, necessitamos, então, de refazer o mapeamento para a dimensão original dos dados, de 250. Com isso, analogamente ao processo anterior, usamos agora o decoder para reconstruir o sinal, de forma a, finalmente, obtermos um sinal no mesmo espaço dos dados originais. Por fim, é estabelecido que somente o difusor é ajustado durante esta etapa; os pesos das redes encoder e decoder são mantidos congelados e utilizados unicamente como modelos pré-treinados.

3 Resultados e discussão

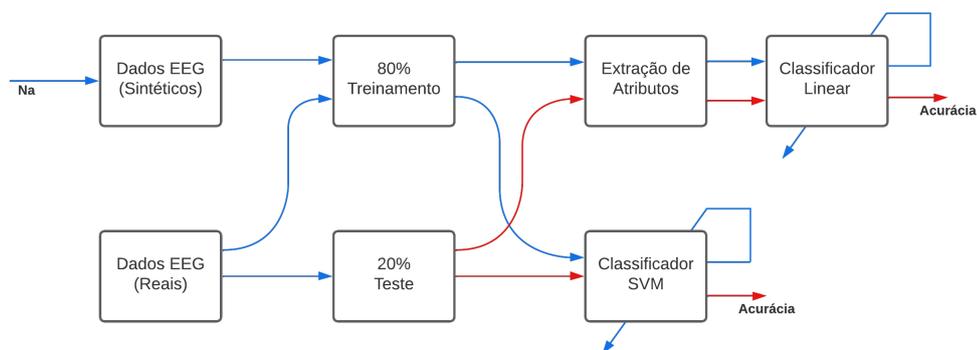
3.1 Procedimento experimental

Para avaliar as contribuições do modelo de difusão no âmbito de ganhos de desempenho para o sistema BCI-SSVEP, delineou-se um procedimento experimental responsável por adicionar amostras sinteticamente geradas junto aos dados de treinamento do classificador empregado. Esse processo se deu de forma gradual, com a adição de N_a amostras por classe, sendo $N_a = \{1, 2, 3, 4, 5, 10\}$. Com isso, o número de amostras totais inseridas correspondeu a 4, 8, 12, 16, 20 e 40, respectivamente.

Considerando que os sinais analisados foram segmentados em janelas de 1 s, totalizando 5 segmentos por *trial*, e que foram empregados 6 *trials* durante a coleta dos dados, cada frequência possui 30 amostras de sinais cerebrais. Como o processo de separação entre treinamento e teste seguiu um padrão de 80/20, 24 amostras por classe foram destinadas ao treinamento, enquanto 6 por classe para o teste. Observa-se aqui, então, que com a adição de amostras sinteticamente geradas pelo modelo difusor, somente o conjunto de dados de treinamento tem sua quantidade de amostras alterada.

A Figura 15 ilustra o fluxograma experimental adotado. É importante destacar que, como mencionado anteriormente, existe uma ligeira diferença de abordagem entre os classificadores linear e SVM. No primeiro caso, o espaço de atributos é reduzido para 4, ou seja, as magnitudes da FFT são extraídas nas frequências correspondentes às classes, enquanto no último caso, os sinais são processados em sua totalidade, com 250 pontos cada.

Figura 15 – Fluxograma do processo.



Fonte: Do autor.

Em termos numéricos, para o processo de adição de sinais sintéticos supracitado, os seguintes cenários experimentais, com as seguintes proporções entre dados reais e sintéticos,

foram obtidos:

Cenário	Número de amostras	Proporção %
1	1	4
2	2	8
3	3	11
4	4	14
5	5	17
6	10	29

Tabela 1 – Cenários experimentais.

3.2 Análise de resultados

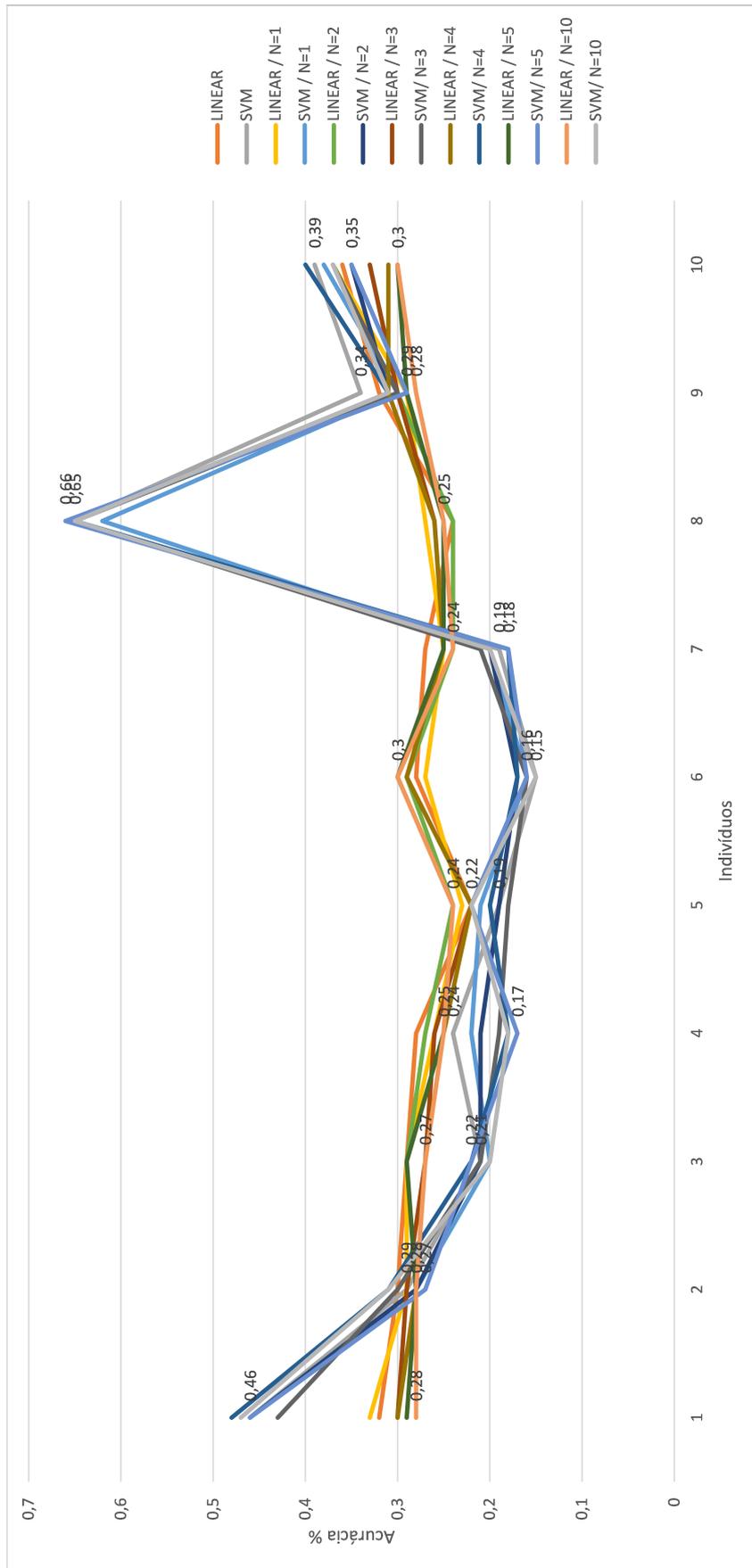
De acordo com o procedimento experimental apresentado, avaliou-se o comportamento da BCI-SSVEP para os 10 primeiros indivíduos da base de dados utilizada, não havendo, portanto, artifícios para a escolha de voluntários que pudessem apresentar resultados mais expressivos e, conseqüentemente, que favorecessem a premissa que permeia o presente estudo.

Em um primeiro momento, realizou-se uma análise quantitativa dos cenários supracitados. Para isso, a Tabela 2 apresenta uma comparação de desempenho entre os cenários considerados como linha de base (*baseline*), ou seja, sem a inclusão de sinais gerados de forma sintética, tanto para o classificador linear quanto para o classificador SVM. Para cada cenário avaliado, considerando que a divisão entre dados de treinamento e teste da abordagem *holdout* ocorre de forma aleatória e, com isso, diferentes partições podem ser obtidas, tomou-se a média de 50 realizações experimentais independentes para representar a acurácia média e os valores de desvio-padrão correspondentes.

BASELINE INDIVÍDUO	LINEAR SVM													
	LINEAR	SVM	N=1		N=2		N=3		N=4		N=5		N=10	
1	0,32 ± 0,07	0,46 ± 0,11	0,33 ± 0,07	0,46 ± 0,10	0,30 ± 0,06	0,46 ± 0,10	0,30 ± 0,05	0,43 ± 0,10	0,30 ± 0,04	0,48 ± 0,10	0,29 ± 0,05	0,46 ± 0,09	0,28 ± 0,03	0,47 ± 0,10
2	0,30 ± 0,07	0,29 ± 0,09	0,29 ± 0,08	0,28 ± 0,09	0,28 ± 0,06	0,28 ± 0,09	0,29 ± 0,05	0,30 ± 0,09	0,28 ± 0,06	0,31 ± 0,10	0,28 ± 0,05	0,27 ± 0,08	0,28 ± 0,03	0,31 ± 0,09
3	0,29 ± 0,08	0,21 ± 0,05	0,29 ± 0,08	0,20 ± 0,07	0,29 ± 0,06	0,21 ± 0,05	0,27 ± 0,05	0,21 ± 0,07	0,27 ± 0,05	0,22 ± 0,06	0,29 ± 0,04	0,22 ± 0,07	0,27 ± 0,03	0,2 ± 0,08
4	0,28 ± 0,09	0,24 ± 0,07	0,26 ± 0,06	0,22 ± 0,06	0,27 ± 0,07	0,21 ± 0,06	0,26 ± 0,05	0,19 ± 0,06	0,25 ± 0,05	0,18 ± 0,07	0,25 ± 0,04	0,17 ± 0,05	0,25 ± 0,03	0,18 ± 0,07
5	0,22 ± 0,07	0,19 ± 0,09	0,23 ± 0,06	0,21 ± 0,09	0,24 ± 0,06	0,19 ± 0,09	0,22 ± 0,06	0,18 ± 0,08	0,22 ± 0,04	0,20 ± 0,11	0,24 ± 0,04	0,22 ± 0,09	0,24 ± 0,03	0,22 ± 0,11
6	0,28 ± 0,07	0,15 ± 0,04	0,27 ± 0,06	0,16 ± 0,05	0,29 ± 0,05	0,17 ± 0,05	0,29 ± 0,05	0,16 ± 0,06	0,29 ± 0,05	0,17 ± 0,05	0,30 ± 0,04	0,16 ± 0,05	0,30 ± 0,03	0,15 ± 0,05
7	0,27 ± 0,1	0,19 ± 0,06	0,25 ± 0,06	0,20 ± 0,07	0,24 ± 0,05	0,20 ± 0,08	0,25 ± 0,05	0,21 ± 0,08	0,25 ± 0,04	0,18 ± 0,07	0,25 ± 0,05	0,18 ± 0,07	0,24 ± 0,03	0,2 ± 0,06
8	0,24 ± 0,09	0,65 ± 0,10	0,27 ± 0,06	0,62 ± 0,09	0,24 ± 0,06	0,65 ± 0,08	0,26 ± 0,06	0,65 ± 0,11	0,26 ± 0,05	0,65 ± 0,09	0,25 ± 0,04	0,66 ± 0,10	0,25 ± 0,03	0,65 ± 0,07
9	0,32 ± 0,08	0,34 ± 0,09	0,29 ± 0,07	0,30 ± 0,10	0,30 ± 0,07	0,31 ± 0,11	0,30 ± 0,06	0,30 ± 0,10	0,31 ± 0,06	0,31 ± 0,11	0,29 ± 0,06	0,29 ± 0,12	0,28 ± 0,04	0,31 ± 0,11
10	0,36 ± 0,08	0,39 ± 0,09	0,37 ± 0,07	0,38 ± 0,09	0,33 ± 0,06	0,35 ± 0,09	0,33 ± 0,06	0,37 ± 0,10	0,31 ± 0,05	0,4 ± 0,10	0,30 ± 0,05	0,35 ± 0,09	0,30 ± 0,03	0,37 ± 0,10

Tabela 2 – Tabela de comparativo de performance entre os cenários *baseline* e com adição de amostras sinteticamente geradas.

Figura 16 – Acurácia dos cenários testados por indivíduo.



Fonte: Do autor.

Dos 10 indivíduos analisados, em 7 deles houve melhora considerando a adição de sinais gerados sinteticamente aos dados de treinamento, com destaque para os indivíduos 1, 5 e 6, nos quais o ganho de desempenho foi de 2 pontos percentuais em relação ao maior valor *baseline*. Mais especificamente, em termos de proporção de desempenho, para o indivíduo 1, o aumento de 0,46 para 0,48 representou um acréscimo de 4,34%; para o indivíduo 5, que foi de 0,22 para 0,24, a melhora foi de 9,09%; por fim, para o indivíduo 6, o aumento foi de 7,14%.

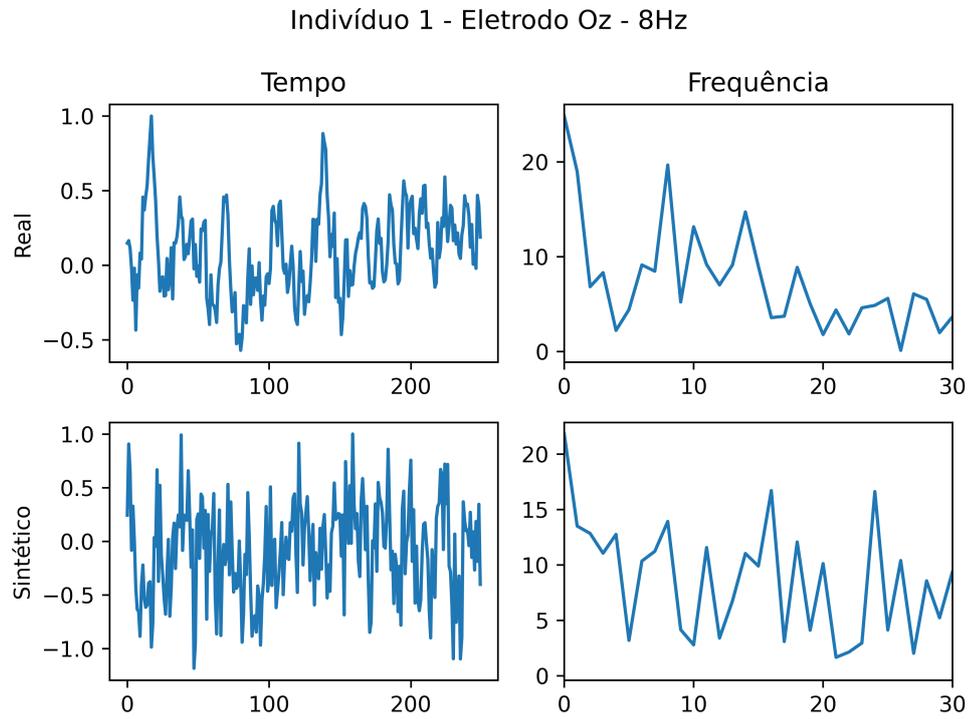
Nos demais casos, ou seja, para os indivíduos 2, 3, 8 e 10, os ganhos de desempenho representaram 1 ponto percentual. Além disso, em todas as 7 ocasiões em que a adição de dados artificiais proporcionou ganhos de desempenho ao sistema, o número de amostras inseridas foi maior ou igual a 4. Tal observação pode ser um indício importante para a investigação futura de cenários com uma quantidade maior de dados sintéticos por classe.

Quanto à comparação entre os classificadores empregados, linear e SVM, o SVM foi superior em 4 dos 7 cenários. Contudo, não é possível estabelecer uma relação de superioridade em relação ao classificador linear. Em uma análise complementar, também não há indicativos de que o sinal gerado deva ser completamente aproveitado pelo classificador, ou seja, todos os seus 250 pontos, como no SVM, ou se deve existir uma etapa prévia de extração de atributos, como no caso linear.

Como último passo na análise quantitativa, observou-se que, para os indivíduos 4, 7 e 9, contudo, não foi alcançado um aprimoramento no desempenho; pelo contrário, foi observada uma deterioração com a inclusão das amostras sintéticas. Ainda assim, considerando que em mais da metade dos cenários avaliados, mais precisamente em 70% deles, houve ganho de desempenho da BCI-SSVEP, é possível demonstrar quantitativamente a viabilidade da aplicação do modelo de difusão para a geração de dados sintéticos de sinais EEG-SSVEP.

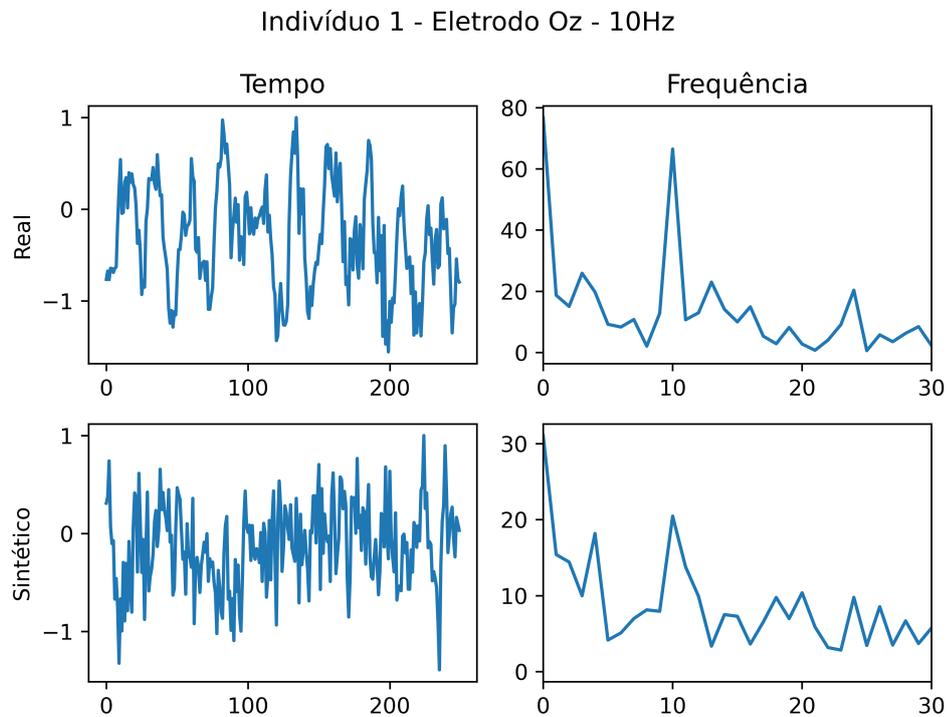
Em uma segunda etapa, procedeu-se a uma avaliação qualitativa das amostras geradas pelo modelo de difusão. Essa avaliação incluiu principalmente análises visuais, tanto no domínio do tempo quanto na frequência. Para isso, selecionou-se o indivíduo 1, para o qual foram obtidos ganhos mais expressivos, conforme indicado na Tabela 2. Os sinais sinteticamente gerados para as quatro frequências avaliadas (8, 10, 12 e 15 Hz) foram, então, analisados no eletrodo Oz, conforme ilustrado nas Figuras 17, 18, 19 e 20.

Figura 17 – Sinal real e sintético do indivíduo 1 na frequência de 8 Hz - Eletrodo Oz.



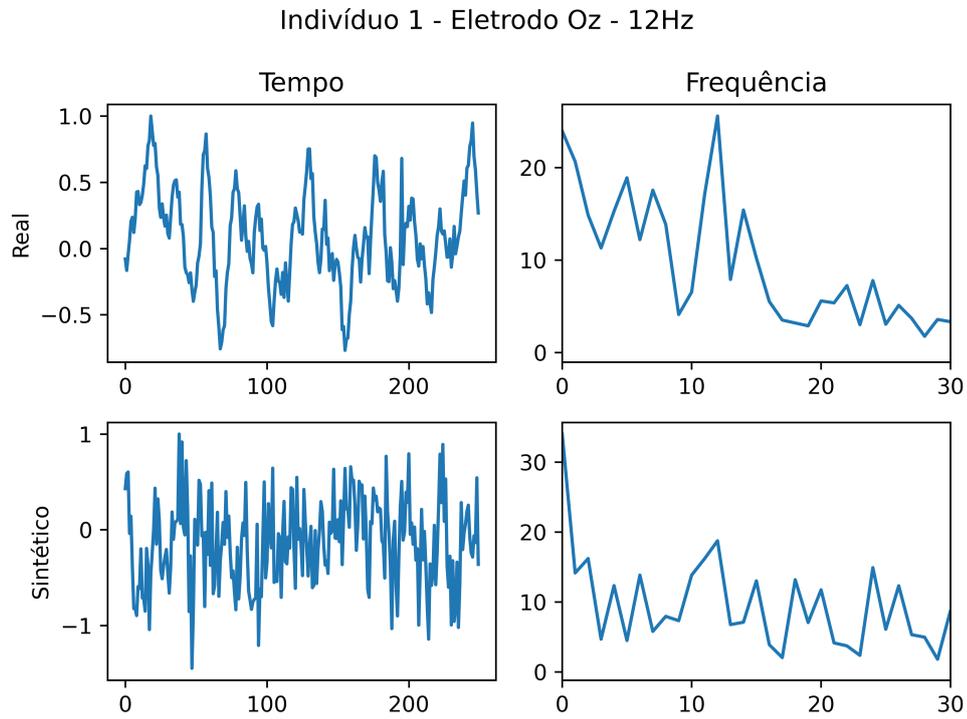
Fonte: Do autor.

Figura 18 – Sinal real e sintético do indivíduo 1 na frequência de 10 Hz - Eletrodo Oz.



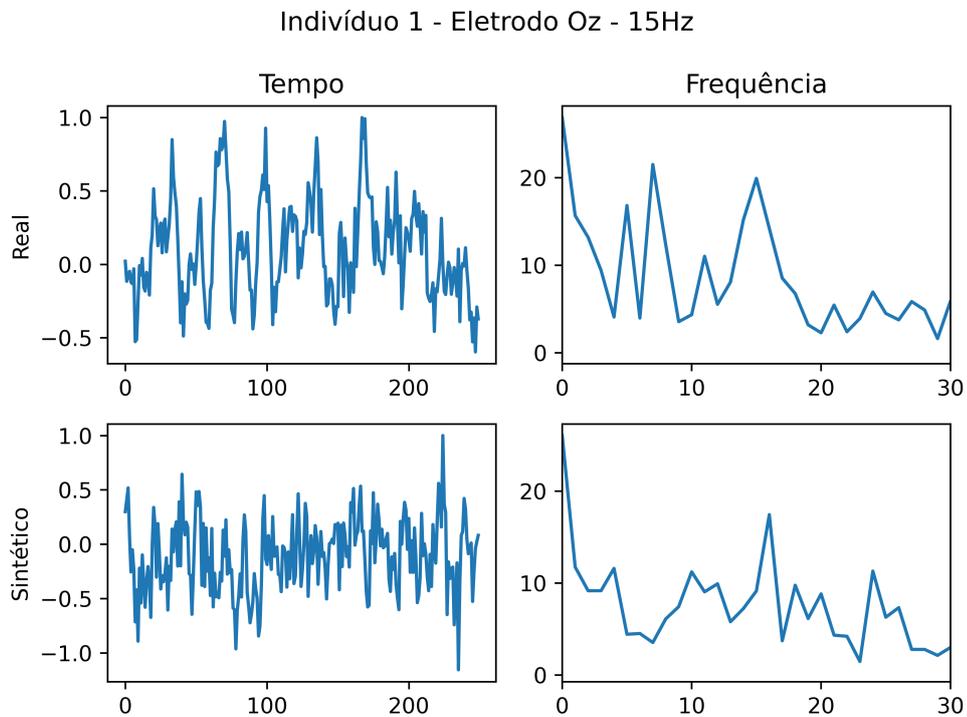
Fonte: Do autor.

Figura 19 – Sinal real e sintético do indivíduo 1 na frequência de 12 Hz - Eletrodo Oz.



Fonte: Do autor.

Figura 20 – Sinal real e sintético do indivíduo 1 na frequência de 15 Hz - Eletrodo Oz.



Fonte: Do autor.

Para todos os casos ilustrados, é possível perceber que, tanto no domínio do tempo quanto na frequência, os sinais sinteticamente gerados conseguiram capturar o comportamento geral do sinal. Mais especificamente no domínio do tempo, a dinâmica do sinal foi capturada pelo modelo de difusão e, na frequência, é possível identificar picos correspondentes às frequências dos estímulos, semelhantes ao sinal real utilizado como base de treinamento para o nosso modelo.

As análises visuais, atuando de modo complementar às análises quantitativas, também revelam que os sinais sinteticamente gerados se apresentam em uma versão aparentemente mais ruidosa do sinal de referência. Isso pode ter contribuído, de certa forma, para ganhos de desempenho menos expressivos nos classificadores avaliados.

Diante das observações e dos resultados expostos, é possível indicar a viabilidade do emprego de modelos de difusão para a geração de dados artificiais. Isso se deve ao fato de que, na grande maioria dos cenários analisados, os ganhos de desempenho foram proporcionados pela adição de sinais sinteticamente gerados.

4 Conclusão

Este trabalho introduziu o modelo de difusão como uma técnica emergente para a geração de dados de EEG-SSVEP, com a premissa de aprimorar o desempenho dos classificadores empregados. Para isso, avaliou o desempenho desses classificadores sem a inclusão de sinais sintéticos - denominado *baseline* - e em diversos cenários que envolvem a adição de diferentes quantidades de sinais cerebrais.

Foi observado que a utilização de sinais gerados de forma sintética resultou em melhorias de desempenho no sistema BCI projetado para a maioria dos indivíduos analisados. Em conjunto com a análise qualitativa, que avaliou a dinâmica do sinal tanto no domínio do tempo quanto na frequência, é possível fortalecer a viabilidade da aplicação de difusores para a geração de sinais sintéticos no campo de BCI-SSVEP.

É importante ressaltar que, especialmente por meio da análise qualitativa (visual), os sinais gerados pelo difusor demonstraram uma leve tendência a serem mais ruidosos em comparação ao sinal de referência. Isso indica que a área de pesquisa relacionada à utilização de modelos de difusão ainda possui um grande potencial a ser explorado, visto que existem poucas contribuições que empregam esse método para gerar sinais cerebrais. Investigações futuras devem ser realizadas, como a exploração de modelos mais robustos no contexto de aprendizado profundo e a utilização de mais canais durante o processo de treinamento e geração de dados.

Referências

- AZNAN, N. K. N. et al. Simulating brain signals: Creating synthetic eeg data via neural-based generative models for improved ssvep classification. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2019. p. 1–8. 3
- AZNAN, N. K. N. et al. Leveraging synthetic subject invariant eeg signals for zero calibration bci. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2021. p. 10418–10425. 4
- BORGES, R. et al. Sincronização de disparos em redes neuronais com plasticidade sináptica. *Revista Brasileira de Ensino de Física*, v. 37, p. 2310–1, 06 2015. 12
- CAO, H. et al. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*, 2022. 20
- CECOTTI, H. A self-paced and calibration-less ssvep-based brain–computer interface speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 18, n. 2, p. 127–133, 2010. 2
- COSTA, T. *Desenvolvimento de interface cérebro-computador baseada em potenciais evocados visualmente em regime estacionário*. Tese (Doutorado) — UNICAMP, 10 2015. 2
- CROITORU, F.-A. et al. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–20, 2023. 3
- DEMARRE, D. A.; MICHAELS, D.; VALDEZ, R. Bioelectronic measurements. *Journal of Clinical Engineering*, LWW, v. 8, n. 3, p. 234, 1983. 1
- DORNHEGE, G. et al. *Toward brain-computer interfacing*. [S.l.]: MIT press, 2007. 7
- GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. [S.l.]: "O'Reilly Media, Inc.", 2022. 14
- GURNEY, K. *An introduction to neural networks*. [S.l.]: CRC press, 2018. 11, 14
- HAGAN, M. et al. *Neural Network Design*. Martin Hagan, 2014. ISBN 9780971732117. Disponível em: <<https://books.google.com.br/books?id=4EW9oQEACAAJ>>. 17
- HARTMANN, K. G.; SCHIRRMEISTER, R. T.; BALL, T. Eeg-gan: Generative adversarial networks for electroencephalographic (eeg) brain signals. *arXiv preprint arXiv:1806.01875*, 2018. 5
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999. (International edition). ISBN 9780132733502. Disponível em: <<https://books.google.com.br/books?id=bX4pAQAAMAAJ>>. 12, 13, 14, 15, 16
- HECKBERT, P. Fourier transforms and the fast fourier transform (fft) algorithm. *Computer Graphics*, v. 2, n. 1995, p. 15–463, 1995. 8
- HO, J.; JAIN, A.; ABBEEL, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, v. 33, p. 6840–6851, 2020. 20, 21, 23

- KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 22
- KOTELNIKOV, A. et al. Tabddpm: Modelling tabular data with diffusion models. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2023. p. 17564–17579. 24
- KOUTROUMBAS, K.; THEODORIDIS, S. *Pattern recognition*. [S.l.]: Academic Press, 2008. 8
- KWON, J.; IM, C.-H. Novel signal-to-signal translation method based on stargan to generate artificial eeg for ssvep-based brain-computer interfaces. *Expert Systems with Applications*, v. 203, p. 117574, 2022. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417422008880>>. 4
- LAWHERN, V. J. et al. Eegnet: a compact convolutional neural network for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, IOP Publishing, v. 15, n. 5, p. 056013, jul 2018. Disponível em: <<https://dx.doi.org/10.1088/1741-2552/aace8c>>. 18
- LEBEDEV, M. A.; NICOLELIS, M. A. L. Interface cérebro-computador: novas perspectivas para a reabilitação. *Revista Neurociências*, n. 29(9), p. 267–269, 2009. 1
- LEITE, S. N. de C. *Contribuições ao desenvolvimento de interfaces cérebro-computador baseadas em potenciais evocados visualmente em regime estacionário*. Tese (Doutorado) — [sn], 2016. 7
- LEITE, S. N. de C. *Redes Neurais Artificiais e Deep Learning*. PPGEF - UFOP / ICEA, 2021. Disponível em: <<https://ppgef.ufop.br/>>. 14
- LIU, B. et al. Beta: A large benchmark database toward ssvep-bci application. *Frontiers in neuroscience*, Frontiers Media SA, v. 14, p. 627, 2020. 6
- NICHOL, A. Q.; DHARIWAL, P. Improved denoising diffusion probabilistic models. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2021. p. 8162–8171. 22, 23
- NICOLAS-ALONSO, L. F.; GOMEZ-GIL, J. Brain computer interfaces, a review. *Sensors*, n. 12, p. 1211–1279l, 2012. 1
- OPPENHEM, A. V. *Discrete-time signal processing*. 1999. 8
- PAN, Y.; CHEN, J.; ZHANG, Y. A survey of deep learning-based classification methods for steady-state visual evoked potentials. *Brain-Apparatus Communication: A Journal of Bacomics*, Taylor & Francis, v. 2, n. 1, p. 2181102, 2023. Disponível em: <<https://doi.org/10.1080/27706710.2023.2181102>>. 4
- RASCHKA, S.; MIRJALILI, V. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. [S.l.]: Packt Publishing Ltd, 2019. 9, 10, 11
- RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. [S.l.]: MIT press, 1986. 17
- SANGER, T.; BALJEKAR, P. N. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958. Disponível em: <<https://api.semanticscholar.org/CorpusID:12781225>>. 12

- SIULY, S.; LI, Y.; ZHANG, Y. Eeg signal analysis and classification. *IEEE Trans Neural Syst Rehabil Eng*, Springer, v. 11, p. 141–144, 2016. 6
- SOHL-DICKSTEIN, J. et al. Deep unsupervised learning using nonequilibrium thermodynamics. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 2256–2265. 20
- TYNER, F. S.; KNOTT, J. R.; MAYER, W. B. *Fundamentals of EEG technology: Clinical correlates*. [S.l.]: Lippincott Williams & Wilkins, 1989. v. 2. 1
- WANG, S.-C. Artificial neural network. In: _____. *Interdisciplinary Computing in Java Programming*. Boston, MA: Springer US, 2003. p. 81–100. ISBN 978-1-4615-0377-4. Disponível em: <https://doi.org/10.1007/978-1-4615-0377-4_5>. 11
- XIE, S. et al. Stimulator selection in ssvep-based spatial selective attention study. *Comput Intell Neurosci*, n. 2016: 6410718, 2016. 2
- XU, D. et al. An analysis of deep learning models in ssvep-based bci: A survey. *Brain Sciences*, v. 13, n. 3, 2023. ISSN 2076-3425. Disponível em: <<https://www.mdpi.com/2076-3425/13/3/483>>. 2, 18