Detecção de piscadas a partir de Eletroencefalograma utilizando Redes Neurais Artificiais

Giovanna Amorim Nascimento * **João Carlos Tonon Campi** ** **Vinícius Rocha Caetano** ***

* *Departamento de Engenharia Elétrica, Universidade Federal de São Carlos, SP, (e-mail: giovannanascimento@estudante.ufscar.br)*
** *Departamento de Engenharia Elétrica, Universidade Federal de São Carlos, SP, (e-mail: campijoao@estudante.ufscar.br)*
*** *Departamento de Engenharia Mecânica, Universidade Federal de São Carlos, SP, (e-mail: vinicius.caetano@estudante.ufscar.br)*

Palavras-chaves: Eletroencefalograma, Piscada, Redes Neurais Artificiais, Detecção, Classificação

Resumo:
O estudo apresentado a seguir, visa a implementação de redes neurais artificiais para a classificação de sinais de piscadas obtidos através de exames de eletroencefalograma realizados em 20 indivíduos. Além disso, também apresenta a comparação entre três redes implementadas com diferentes características dos sinais.
Palavras-chaves: Eletroencefalograma, Piscada, Redes Neurais Artificiais, Detecção, Classificação.

1. INTRODUÇÃO

O eletroencefalograma (EEG) tem sido uma poderosa ferramenta no estudo do comportamento do cérebro humano. Suas aplicações vão desde diagnósticos clínicos e médicos como deficit de atenção, coma, Alzheimer, epilepsia, entre outros distúrbios cerebrais, até aplicações mais avançadas de mapeamento cerebral, identificação de atenção e movimento e sistemas de neurofeedback.

Um dos desafios no uso desse tipo de ferramenta é a interferência de artefatos que podem influenciar nos resultados da EEG, como por exemplo a movimentação da cabeça e pescoço, a atividade elétrica dos músculos, o atrito dos sensores com o couro cabeludo ou cabelo, ou mesmo o movimento de piscar os olhos. Alguns desses artefatos não podem ser identificados para serem removidos por se tratarem de comportamentos individuas que mudam conforme a pessoa estudada, no entanto outros artefatos são provenientes de ações que podem ser identificadas e removidas, como é o caso do piscar de olhos, que é justamente o objetivo dessa pesquisa.

Ao longo desse estudo foi realizada a implementação de três redes neurais artificiais, que foram treinadas para identificar o sinal de piscada em sensores FP1 e FP2 de EEG. As redes foram treinadas com as características originais do sinal, a energia do sinal e a junção das duas características de ambos os canais. O intuito de identificar esse tipo de artefato é o reconhecimento dele em exames de EEG, para que possam ser removidos ou ignorados, e não influenciar em resultados realmente significativos para exames assim.

Com isso, nas seções subsequentes, serão expostas as metodologias empregadas juntamente com os resultados obtidos pelas redes utilizadas no processo de classificação do sinal.

2. METODOLOGIA

Ao longo dessa seção serão apresentadas as metodologias utilizadas para o desenvolvimento dessa pesquisa, além das referências e bases de dados utilizadas.

2.1 Base de dados

Para que a realização desta atividade fosse possível, foram utilizados dados obtidos de sensores EEG registrados para eletrodos frontais (FP1, FP2) de 20 indivíduos usando dispositivo OpenBCI e BIOPAC Cap100C, sendo que esta base de dados foi disponibilizada pelos pesquisadores Agarwal and Sivakumar (2019), Agarwal and Sivakumar (2020) e Gupta et al. (2020) da Escola de Engenharia Elétrica e de Computação da Georgia Tech [1] .

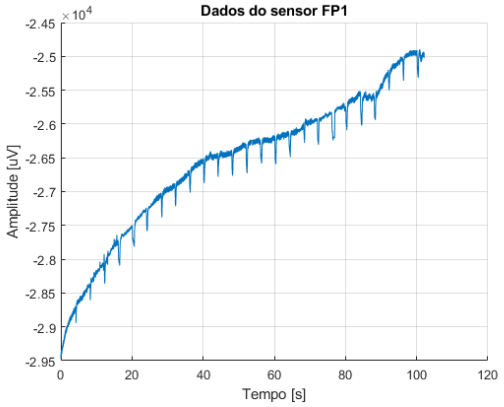
Além disso, os arquivos fornecidos foram categorizados da seguinte maneira: EEG-IO, EEG-VV e EEG-VR. O primeiro aborda piscadas oculares voluntárias, o segundo trata de piscadas involuntárias durante a visualização de um vídeo pelo voluntário, e o último diz respeito a piscadas involuntárias enquanto o voluntário lê um artigo. Além desses grupos, para cada um dos três tipos de coleta mencionados anteriormente, existe um conjunto de arquivos denominados *data* e *label*. Os arquivos *data* contêm os valores registrados pelos sensores FP1 e FP2

[1] https://gnan.ece.gatech.edu/eeg-eyeblinks/

de cada indivíduo durante todo o período de coleta, enquanto os arquivos *label* indicam o momento exato em que ocorreram as piscadas, bem como se há algum momento do arquivo que está corrompido.

A Figura 1 mostra um exemplo de como é um sinal obtido por meio desta base de dados.

Figura 1. Exemplo do sinal obtido através da base de dados



Fonte: Autores

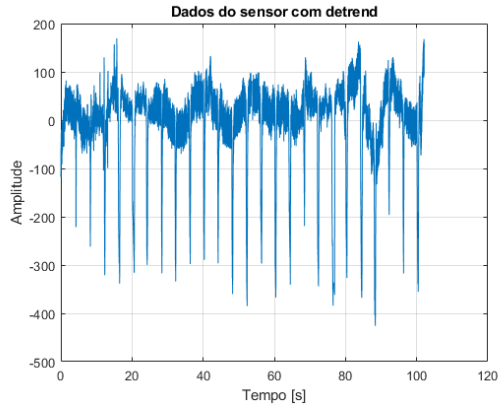
2.2 Pré processamento dos sinais

Para que os sinais da base de dados pudessem ser utilizados, foi necessário realizar um pré processamento dos mesmos.

Conforme mostra a Figura 1 o sinal dos sensores FP1 e FP2 apresentam uma tendência, vista pelo deslocamento do sinal no eixo vertical, ao longo do tempo. Para remover a tendência do sinal, foi utilizada a função *detrend* no MATLAB, que remove a tendência polinomial do sinal. No caso do sinal apresentado, o polinômio definido foi de grau 10.

Também foi realizada a normalização do sinal sem a tendência, dividindo o mesmo pela média do valor máximo e mínimo de forma a facilitar na detecção de possíveis outliers. A Figura 2, mostra o sinal sem tendência e normalizado.

Figura 2. Exemplo do sinal sem tendência e normalizado



Fonte: Autores

Esses dados foram separados em janelas, conforme será detalhado na próxima seção. Os sinais de cada sensor foram divididos em 400 janelas que foram concatenadas em vetores de 800 janelas, sendo as 400 primeiras relacionadas ao sensor FP1 e as demais ao sensor FP2. Nesse mesmo vetor foi adicionado uma classe relacionada ao momento da piscada, obtida através do arquivo *labels*, somando assim 801 classes por vetor. Esses dados foram salvos no arquivo amostras, sendo obtida dessa forma, a primeira base de dados utilizada.

2.3 Filtragem e obtenção da energia do sinal

A fim de obter mais características da piscada, foi realizada a FFT (*Fast Fourrier Transform*) nas janelas do sinal dos sensores, na qual se passa a fazer a análise no domínio da frequência.

Com o sinal no domínio da frequência, foi feito um banco de filtros para obter apenas as características desejadas. Como a frequência de piscadas é baixa o banco de filtros foi implementado de 0Hz a 10Hz, após isso não foram identificados dados relevantes para o sinal.

O primeiro filtro implementado foi um passa-baixa, do tipo Butterworth, com a frequência de corte em 2Hz, seguido por quatro filtros passa-banda, do mesmo tipo, com as frequências de 2 a 4Hz, 4 a 6Hz, 6 a 8Hz e 8 a 10Hz, respectivamente.

Com os dados dos filtros foi extraída a energia do sinal, conforme apresentado na Equação 1, na qual N é o número de componentes frequenciais e $S(f)$ o sinal de piscada no domínio da frequência.

$$E = \frac{1}{N} \sum_{i=1}^{N} S(f)^2 \tag{1}$$

A energia do sinal de cada sensor foi concatenada em vetores de 10 classes, sendo as 5 primeiras relacionadas ao sensor FP1 e as demais ao sensor FP2. Para completar a base de dados, foi acrescentada uma classe referente ao momento da piscada, obtida através do arquivo *labels*.

2.4 Criação da rede

Para realizar a confecção das redes neurais deste trabalho, foi utilizado um aplicativo da MathWorks, que por sua vez fica disponível dentro do ambiente de programação da linguagem MATLAB, chamado *Neural Net Pattern Recognition*. Os dados de treinamento são preparados com entradas e saídas desejadas, divididos em conjuntos de treinamento, validação e teste. A arquitetura da rede é criada especificando o número de neurônios e funções de ativação, no entanto o número de camadas ocultas é definido pelo aplicativo. A rede é treinada usando a função *train* com um algoritmo de otimização, monitorando o desempenho com dados de validação e ajustando hiperparâmetros para evitar *overfitting*. Após o treinamento, o desempenho é avaliado com dados de teste usando métricas como acurácia. A rede treinada é então usada para fazer previsões em novos dados, destacando a importância da configuração adequada da rede, dos dados de treinamento e dos parâmetros de treinamento para obter resultados eficazes.

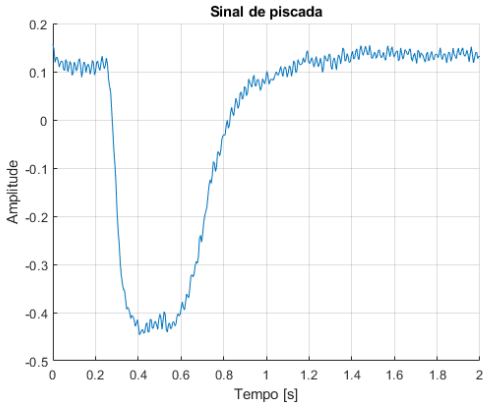
3. RESULTADOS E DISCUSSÕES

Esta seção irá apresentar todos os resultados e discussões acerca do treinamento da rede para a detecção de piscadas por um sinal de EEG, utilizando os sensores FP1 e FP2. O código fonte utilizado neste projeto pode encontrado no GitHub [2] .

3.1 Criação da base de dados e arquitetura de Rede

A primeira etapa de todo o projeto é observar e identificar os momentos em que ocorrem a piscada. A Figura 3 trás um exemplo do que é um sinal de piscada nos sensores FP1 ou FP2.

Figura 3. Exemplo de um sinal de EEG para a Piscada



Fonte: Autores

A partir da Figura 3, é possível definir uma janela de recorte de 400 amostras, na qual é possível englobar todo o período de piscada. Considerando que a frequência de amostragem é de $F_s = 250$ Hz, a janela observada possui aproximadamente $t_s = 1.6$ segundos.

Para criar a base de dados necessária, indicando quais são os dados considerados piscadas ou não, é necessário utilizar o arquivo da base de dados *labels* que possuí a marcação temporal de quando ocorre a piscada. Logo, os dados são separados utilizando a janela anteriormente citada, a movendo com um passo de 100 amostras e detectando se ela engloba mais de 50% de uma piscada. Caso essa condição seja verdadeira o conjunto é dado como piscada, mas caso contrário não é considerado. Utilizando-se então de 19 exemplos de dados dos sensores FP1 e FP2 separados da forma citada acima, gerando 2275 elementos diferentes, três tipos de base de dados são formadas descritas da seguinte forma:

B1: 801 classes, sendo destas 400 relacionadas aos dados do sensor FP1, 400 aos dados do sensor FP2 e 1 para a classificação de piscada;

B2: 11 classes, sendo destas 5 relacionadas à energia do sinal pelo banco de filtros do sinal de FP1, 5 referentes à energia do sinal de FP2 e 1 para classificação de piscada;

B3: 811 classes, sendo a união das duas anteriores.

[2] https://github.com/joaocarloscampi/Projeto_ProcessamentoDeSinais/tree/main

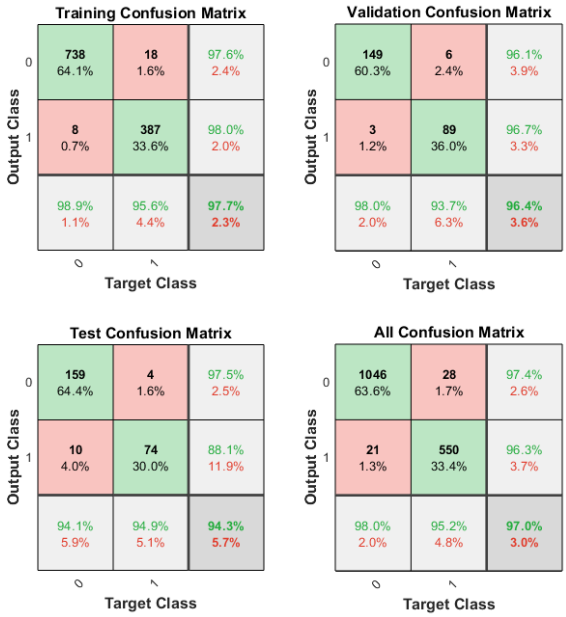
A energia do sinal presente em duas das bases de dados é calculada a partir das parcelas obtidas pelo banco de filtros pela Equação 1. O banco para esse sinal é definido por filtros passa-faixa centrado nas frequências $f_c = [1, 3, 5, 7, 9]$ Hz, com banda de $\Delta f = 2$Hz.

Para todas as bases aqui apresentadas em teste, é definido que a arquitetura de rede a ser utilizada é a FeedFoward, possuindo uma camada de entrada com o número de neurônios correspondentes a quantidade de entradas da base, uma camada intermediária com 50 neurônios fixa e um neurônio na sua camada de saída. Além disso, as bases de dados para as redes são dividas em 1645 amostras de treinamento e 650 amostras para validação.

3.2 Resultados para a base B1

Utilizando uma rede neural 800-50-1, os resultados de treinamento e validação de rede podem ser vistos nas Figuras 4 e 5 a seguir.

Figura 4. Treinamento da rede 1



Fonte: Autores

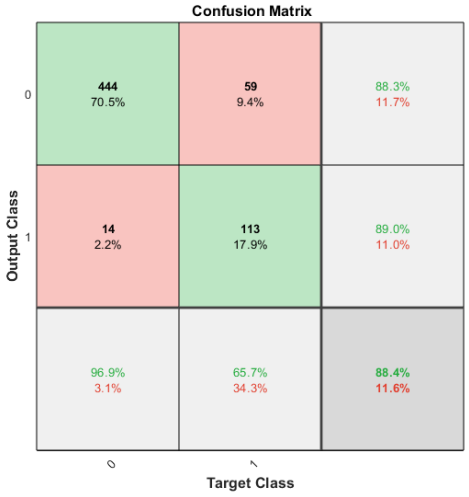
A comparação das duas figuras apresentadas mostra uma grande precisão nos dados de treinamento, mas uma queda de rendimento para os dados de teste. Isso é um efeito esperado, ainda mais com as precisões estando em torno de 96.8% e 88.4%, respectivamente, indicando uma boa generalização da rede para os dados de validação apresentados.

3.3 Resultados para a base B2

Neste caso, utilizando uma rede neural 10-50-1, os resultados de treinamento e validação de rede podem ser vistos nas Figuras 6 e 7 a seguir.

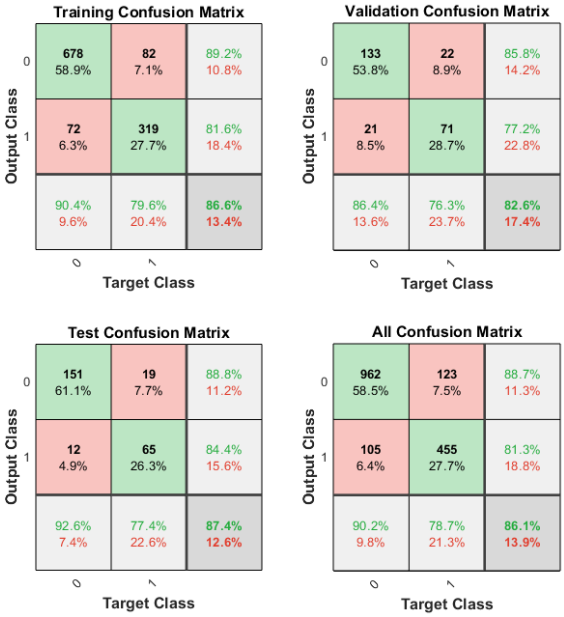
Os resultados desta rede se mostram menos distantes, tendo como precisão 86.1% na base de treino e 75.7% na base de testes. Entretanto, em comparação com a

Figura 5. Validação da rede 1



Fonte: Autores

Figura 6. Treinamento da rede 2



Fonte: Autores

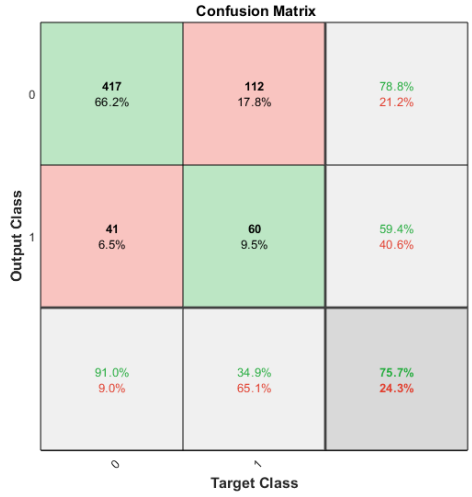
base anterior é evidenciado uma dificuldade maior de treinamento e, além disso, um desempenho inferior em questão de generalização dos dados.

3.4 Resultados para a base B3

Por fim, utilizando uma rede neural 810-50-1, os resultados de treinamento e validação de rede englobando os dois casos anteriores podem ser vistos nas Figuras 8 e 9 a seguir.

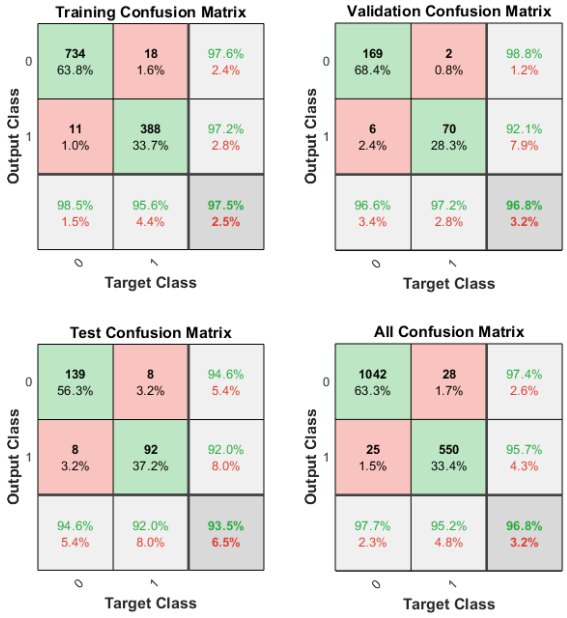
Para este caso é possível perceber um desempenho um pouco inferior ao apresentado na primeira base, sendo 96.8% para o treinamento e 87.5% para a validação. Isso pode ser um indicador de que os dados de energia do sinal não ajudam no treinamento da rede, tendo em vista que

Figura 7. Validação da rede 2



Fonte: Autores

Figura 8. Treinamento da rede 3



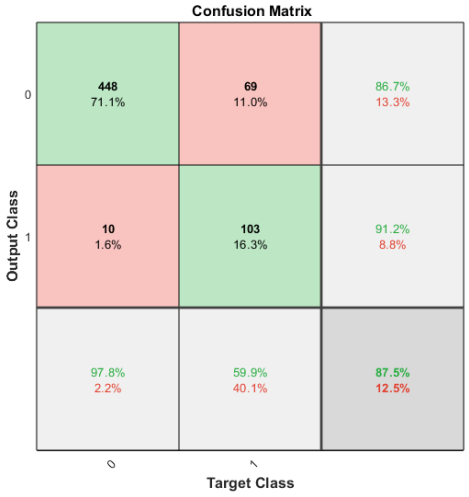
Fonte: Autores

ao acrescenta-los o desempenho da rede piorou em relação a primeira. Por conta do volume de informações contidas nos dois sensores, a diferença entre os dois desempenhos é mínima, indicando que os dados de energia também não possuem tanta relevância e peso no resultado final da rede.

4. CONCLUSÃO

Os resultados deste trabalho conduzem a algumas conclusões pertinentes. Primeiramente, observou-se uma alta precisão nos dados de treinamento em todas as bases de dados avaliadas. Contudo, uma pequena queda no desempenho de generalização ocorreu durante a validação, o que merece destaque. Além disso, é notável o achado em um dos *databases*, onde a adição de informações sobre a energia do

Figura 9. Validação da rede 3



Fonte: Autores

sinal não trouxe os benefícios esperados para o treinamento da rede, sendo que na verdade tal inclusão resultou em uma pequena queda de desempenho em comparação com os resultados obtidos na primeira base de dados investigada. Uma possível causa para este efeito deve-se ao fato de que a base B2 possui aproximadamente 10 vezes menos dados em comparação com a B1, no entanto ela ainda sim consegue um resultado satisfatório. Para que esses resultados possam tornar-se mais precisos, a adição de mais índices pode ser uma possível solução. Outrossim, uma provável causa para uma sucinta queda entre os resultados de B3 e B2 pode ser fruto da própria rede neural, visto que pelo fato dos dados serem muito diferentes entre si a rede pode priorizar os dados do sensor em relação as informações obtidas pelos filtros.

Portanto, a escolha de um banco de dados e a configuração otimizada dos parâmetros da rede neural emergem como fatores cruciais que impactam profundamente a capacidade de generalização das informações. Ademais, surgem como alternativas que podem enriquecer a exploração da rede neural e aprimorar ainda mais os resultados questões como a redução seletiva da quantidade de dados por sensor, a inclusão de métricas relacionadas ao espectro de frequência original e a consideração de indicadores como o valor RMS, bem como a busca por um equilíbrio na distribuição das categorias piscada e não piscada.

REFERÊNCIAS

Agarwal, M. and Sivakumar, R. (2019). Blink: A fully automated unsupervised algorithm for eye-blink detection in eeg signals. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1113–1121. IEEE.

Agarwal, M. and Sivakumar, R. (2020). Charge for a whole day: Extending battery life for bci wearables using a lightweight wake-up command. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14.

Gupta, E., Agarwal, M., and Sivakumar, R. (2020). Blink to get in: Biometric authentication for mobile devices using eeg signals. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.