

Classificação de Ondas EEG para Privação de Sono

1st Beatriz Matias Santana Maia
Colegiado do Curso de Ciência da Computação
Universidade Federal do Espírito Santo
Vitória, Brasil
beatriz.maia@edu.ufes.br

2nd Higor David Oliveira
Programa de Pós-Graduação em Informática
Universidade Federal do Espírito Santo
Vitória, Brasil
higor.d.oliveira@edu.ufes.br

Abstract—Nos últimos anos, os estudos sobre privação de sono têm revelado impactos significativos na saúde física e mental, afetando desde funções cognitivas até o sistema imunológico. Os efeitos colaterais da ausência de sono adequado são de curto a longo prazo, enfatizando a necessidade de profissionais da saúde gerenciar a saúde de seus pacientes junto a continuidade do sono. Esta pesquisa busca estudar o uso de modelos diferentes para a classificação de dados EEG em sono normal e privação de sono. Os modelos utilizados foram Máquina de Vetores de Suporte (SVM), K-Vizinhos Mais Próximos (KNN) e *Random Forest*. O modelo *Random Forest* se destacou, obtendo uma acurácia de 93%, mostrando como o uso de aprendizado de máquina é capaz de identificar o tipo de sono dado uma onda EEG.

Index Terms—Machine Learning, Sleep Deprivation, EEG, Classification

I. INTRODUÇÃO

Nos últimos anos, os estudos sobre privação de sono têm revelado impactos significativos na saúde física e mental [1], afetando desde funções cognitivas até o sistema imunológico [2]. Os efeitos colaterais da ausência de sono adequado são de curto a longo prazo, enfatizando a necessidade de profissionais da saúde gerenciar a saúde de seus pacientes junto a continuidade do sono [3]. Para investigar a existência da distúrbios relacionado ao sono é utilizado exames eletroencefalograma (EEG) para detectar atividade cerebral [4].

Paralelamente, o avanço da tecnologia de Aprendizado de Máquina tem revolucionado diversas áreas da saúde [5], [6], oferecendo novas perspectivas para diagnósticos mais precisos e tratamentos personalizados. Em particular, no campo dos sinais de EEG, recentes progressos têm destacado o potencial da Ciência de Dados e Aprendizado de Máquina para análise e interpretação desses sinais complexos [7], proporcionando conhecimento valiosos para entender melhor a atividade cerebral em condições normais e patológicas. Contudo, esses avanços não estão isentos de desafios significativos, incluindo a necessidade de lidar com a variabilidade dos sinais de EEG e a complexidade na interpretação dos padrões neurais capturados [8].

Dentro da área de sono, se destaca principalmente o uso de inteligência computacional para classificação dos diferentes estágios de sono utilizando sinais EEG, porém podendo depender de dados adicionais do tipo eletro-oculografia (EOG) e eletromiografia (EMG) [9], [10]. A escassez de dados é um dos fatores limitantes para avanços de pesquisa dentro de aprendizado profundo, com diversos não se disponibilizando

de forma aberta. As organizações *Sleep Research Society*¹ e *National Sleep Research Resource*², que visam avançar as pesquisas relacionada ao sono, levantam em conjunto a presença de 55 datasets, no qual diversos não se encontram de forma aberta ou de fácil acesso, com dados variando de sinais EEG até informação de censos sobre qualidade de sono.

Com essa variedade de dados a respeito do sono e a carência de estudos voltado a aplicação de métodos de aprendizado de máquina voltado a compreensão de sinais EEG dentro do domínio do sono, este artigo propõe:

- Aplicar métodos de aprendizado de máquina no *dataset* "A Resting-state EEG Dataset for Sleep Deprivation" [11].
- Analisar o impacto de diferentes hiperparâmetros no resultado dos modelos.

Este artigo está dividido em: Primeiramente, materiais e métodos, onde será descrito o dataset usado e a fundamentação teórica por trás das técnicas usadas no trabalhos. Em seguida, serão apresentados e discutido os resultados obtidos. E, por fim, serão feitas conclusões a respeito do trabalho, summarizando os resultados obtidos e indicando possíveis futuros passos.

II. MATERIAIS E MÉTODOS

A. Dataset

O *dataset* "A Resting-state EEG Dataset for Sleep Deprivation" [11] apresenta sinais EEG a respeito de 71 indivíduos que foram submetidos a dois tipos de experimentos envolvendo sono normal e privação de sono. Os dados coletados foram feitos utilizando o sistema internacional 10-20, apresentado pela Figura ??, com informação de apenas 61 canais sendo armazenada. Durante o experimento de sono normal, os participantes dormiam em seus respectivos lares e poderiam utilizar um diário de sono para descrever sobre a sua noite. Para o experimento de privação de sono, os participantes eram obrigados a ficarem no laboratório. Após de cada exame, era feito a coleta de sinais EEG com olhos abertos e fechados. O dataset consiste em dados EEG dos diferentes experimentos e dos diferentes tipos de coletas. A ordem de experimentos varia por participantes. Além disso, também foi coletado resposta de diferentes avaliações de humor, porém a

¹<https://sleepresearchsociety.org/about/who-we-are/>

²<https://sleepdata.org/pages/about>

Tipo de Dado	Quantidade
Olhos fechados - sono normal	38
Olhos fechados - privação de sono	36
Olhos abertos - sono normal	68
Olhos abertos - privação de sono	70

TABLE I
QUANTIDADE DE TIPO DE DADO POR TIPO DE SONO.

coleta dessa informação não é consistente entre os diferentes participantes. Os dados de olhos fechados e olhos abertos estão desbalanceados, com dados de olhos abertos possuindo quase o dobro de exemplos comparado a olhos fechados, como mostra a Tabela I.

O processamento do sinal de EEG para densidade espectral de potência foi feito em Python usando o método de Welch disponibilizado pela biblioteca MNE, sendo anteriormente filtrado por um filtro passa banda entre 0.2Hz e 45Hz. Os dados então foram analisados na escala dB.

B. Transformada de Fourier

A transformada de Fourier tem como objetivo decompor um sinal com base na série de Fourier [12]. A série de Fourier consiste em uma soma complexa de senos e cossenos com as componentes de frequência do sinal [13]. As características relacionadas as componentes de frequências do sinal podem então ser usadas como *features* para representação do sinal original. A equação generalizada da série de Fourier se encontra abaixo.

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx) \quad (1)$$

sendo, a_0 o escalar associado a sinal de frequência 0Hz (sinal contínuo)

a_n , o escalar associado a componentes pares de frequência e b_n o escalar associado a componentes impares de frequência.

C. Método de Welch

O método de Welch é um método computacional que pode ser usado para calculo da potência espectral de um sinal. Também chamado de método de periodogramas, o método combina várias transformadas de Fourier em blocos sucessivos [14], chamados periodogramas. O periodograma identifica as frequências dominantes para aquela janela de tempo, servindo como um filtro para sinais com ruídos de largura de banda alta. Em seguida é calculado a média das frequências entre os blocos. O periodograma de um bloco m é definido por [14]:

$$P_{x_m, M}(w_K) = \frac{1}{M} |FFT_{N, k}(x_m)|^2 \quad (2)$$

$$P_{x_m, M}(w_K) \triangleq \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j2\pi n k / N} \right|^2 \quad (3)$$

Sendo, N o tamanho da janela

x_m , as amostras na janela m

k , o número de janelas disponíveis

A partir desta equação, calcula-se a densidade espectral de potência somando os valores dos periodogramas.

$$S_x^W(w_k) \triangleq \frac{1}{k} \sum_{m=0}^{K-1} P_{x_m, M}(w_K) \quad (4)$$

D. Máquina de Vetores de Suporte

Máquina de Vetores de Suporte (em inglês: Support Vector Machine - SVM) é um algoritmo de aprendizado supervisionado que visa separar valores dentro de um hiperplano de $N-1$ dimensão, onde N é o número de *features* de um dado [15], [16]. O hiperplano ótimo é capaz de representar a completa separação dos dados das diferentes classes, maximizando a distância (margem) entre os dados de classes diferentes mais próximos ao hiperplano. Para identificar o melhor hiperplano, utiliza-se pontos de dados próximo ao hiperplano, denominados de vetores de suporte. Além disso, o método utiliza uma função *kernel*, que através dos pontos de entradas e o método utilizado, é capaz de transformar o espaço em que será encontrado o hiperplano.

E. K-Vizinhos Mais Próximos

O algoritmo de K-Vizinhos Mais Próximos (em inglês: K-Nearest Neighbor - KNN) é uma técnica de clusterização de dados, agrupando em K diferente conjuntos. Com o valor de K conhecido, é possível utilizar o KNN como um algoritmo de classificação. Durante a etapa de treino, o modelo aprende a localização espacial dos diferentes dados e a classe a qual pertence. Para inferência, o modelo utiliza esse mapeamento de pontos conhecidos para classificar o dado de entrada. O dado de entrada é comparado ao K dados mais próximos utilizando uma métrica de distância. O dado é então classificado a classe majoritária [17].

F. Árvore de Decisão

O algoritmo de aprendizado supervisionado Árvore de Decisão (em inglês: Decision Tree- DT) se comporta como um grafo direcionado não cíclico, com diferentes nós que apontam para outros. Os nós podem ser divididos em três tipos: nó raiz, nó de interno e nó folhas [18].

Os nós de decisão são divididos em nó raiz e nó interno e são os responsáveis pela criação de regras em que serão feitas inferências sobre, utilizando os atributos da base de dados. O nó raiz simboliza o início da árvore, não possuindo parentes. Os nós de folhas são os resultados finais, podendo representar a classe do resultado das regras. Para medir a qualidade de uma regra, utiliza-se diferentes métodos para calcular, como a Impureza de Gini ou a Entropia.

A Impureza de Gini mede a probabilidade de uma instância ser classificada erroneamente quando escolhida de forma aleatória e é definida por:

$$Gini(t) = 1 - \sum_{i=1}^j (P(i))^2 \quad (5)$$

no qual t representa a árvore, j representa o número de classes existentes e P representa a razão das classes no i -ésimo nó [18].

A Entropia mede a desordem, o nível de "incerteza", de um nó, levando em consideração a variância das classes das instâncias em um determinado nó. A Entropia é determinada por:

$$Entropia(t) = - \sum_{i=1}^j P(i) \log_2(P(i)) \quad (6)$$

no qual $P(i)$ é a probabilidade de selecionar um exemplo da classe i .

G. Random Forest

O algoritmo *Random Forest* é composto por *ensemble* de DT [19]. A técnica de *ensemble* utilizada para unificar as diferentes DT é denominada de *bagging*. Cada modelo recebe uma combinação reduzida de dados do conjunto de treino, de forma que cada modelo é exposto a conjuntos de dados diferentes. A resposta do *ensemble* leva em consideração a resposta majoritária dos diversos modelos. Os modelos de DT são muitos suscetíveis a *overfitting* aos dados de treino [20]. A técnica de *bagging* introduz variabilidade de modelos, diminuindo o impacto de modelos com índices de *overfitting* [19].

H. Métricas

Para avaliação dos modelos, foi utilizado as métricas de acurácia, *recall* e precisão, calculadas por:

$$Acurácia = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$Precisão = \frac{TN}{TN + FP} \quad (9)$$

Os valores TP , TN , FP , e FN significam, respectivamente, verdadeiros positivos, verdadeiros negativos, falso positivos, e falsos negativos.

I. Configuração experimental

O experimento foi feito utilizando validação cruzada de 5 subconjuntos (*folds*). Os dados inicialmente foram identificados por participante, em vez de por experimento ou tipo de coleta, de forma que um participante poderia participar apenas de um conjunto ou um *fold*. Esse cuidado é para evitar que possíveis características individuais coletadas no dado EEG pudesse contaminar outros conjuntos, afetando o resultado. Os dados estavam na escala de 10^{-5} , com dados temporais em 150.000 instâncias. Para diminuir a dimensionalidade e obter mais informações, foi aplicado uma transformação

SVM	
Hiperparâmetros	Valores
Parâmetro de regularização (C)	[0.1, 1, 10, 100]
Kernel	['linear', 'poly', 'rbf', 'sigmoid']
Coeficiente do kernel (Gamma)	[1, 0.1, 0.01, 0.001, 'scale']
KNN	
Hiperparâmetros	Valores
Peso	['uniform', 'distance']
Algoritmo	['ball_tree', 'kd_tree', 'brute']
Métrica	['euclidean', 'manhattan', 'minkowski']
Random Forest	
Hiperparâmetros	Valores
Número de árvores	[10, 50, 100, 200]
Função de qualidade dos <i>splits</i>	['gini', 'entropy', 'log_loss']
Profundidade máxima	[None, 5, 10, 20]
Quantidade de exemplos mínimo por <i>split</i>	[2, 5, 15]

TABLE II
HIPERPARÂMETROS UTILIZADOS EM BUSCA DE GRADE.

Hiperparâmetro	Valor
Tamanho do batch	16
Épocas	10
Taxa de aprendizado	0.001
<i>Momentum</i>	0.9

TABLE III
HIPERPARÂMETROS UTILIZADOS PARA O TREINO DO MODELO RESNET50.

considerando a densidade espectral. Para a transformação, foi utilizado uma frequência mínima e máxima de 0.2 Hz e 45 Hz, respectivamente. A escolha desses valores foi baseada em frequências estudadas e os tipos de ondas identificadas por sinais EEG [21].

Os experimentos foram então divididos em três partes

- Treinamento de modelos utilizando apenas os dados coletado com olhos abertos.
- Treinamento de modelos utilizando apenas os dados coletado com olhos fechados.
- Treinamento de modelos utilizando os dois tipos de dados.

Dentro de cada experimento foi feito uma busca em grade para os modelos SVM, KNN e *Random Forest*. Os hiperparâmetros analisados estão descrito na Tabela II. A escolha dos hiperparâmetros foi baseado no objetivo de estudar sobre os diferentes modelos, com os valores escolhidos levando em consideração as possibilidades implementadas por o Scikit-Learn [22], além de informações sensíveis ao tamanho do dataset (como por exemplo, quantidade de exemplos mínimo por *split*). Para o treino utilizando o KNN, foi considerado $k = 2$, visto o seu uso como um classificador de sono normal (0) ou privação de sono (1).

Além disso foi feito um treino com o modelo ResNet50 e otimizador SGD para poder comparar os esforços e resultados aos modelos anteriores. A Tabela III apresenta os hiperparâmetros utilizados.

Hiperparâmetros	
Variável	Significado
SVM	
C	Parâmetro de regularização
Gamma	Coefficiente do kernel
Kernel	Kernel
KNN	
Algn	Algoritmo para computar vizinho mais próximo
Dist	Métrica de distância
P	Tipo de abordagem relacionada a peso dos dados
Random Forest	
Crit	Função para medir a qualidade dos nós
D	Profundidade máxima
Min_S	Quantidade mínima de exemplos por nó interno
N_DT	Número de árvores de decisão

TABLE IV

LEGENDA DE VARIÁVEIS UTILIZADAS PARA ANOTAR RESULTADOS E SEUS SIGNIFICADOS.

OLHOS FECHADOS			
Configuração	Acurácia	Recall	Precisão
SVM			
C=100 Gamma=scale Kernel=Poly	0.87	1.00	0.78
C=1 Gamma=1 Kernel=Linear	0.73	0.71	0.71
C=10 Gamma=1 Kernel=Poly	0.87	1.00	0.78
C=1 Gamma=0.001 Kernel=Poly	0.87	1.00	0.78
C=1 Gamma=0.001 Kernel=Linear	0.73	0.71	0.71
KNN			
Algm=brute Dist=minkowski P=dist	0.75	0.83	0.71
Algm= ball_tree Dist=minkowski P=dist	0.75	0.83	0.71
Algm= ball_tree Dist=minkowski P=unif	0.75	0.50	1.00
Algm= brute Dist=minkowski P=unif	0.75	0.50	1.00
Algm= ball_tree Dist=minkowski P=dist	0.75	0.83	0.71
Random Forest			
Crit=log_loss D=10 Min_S=2 N_DT=100	0.87	1.00	0.78
Crit=gini D=2 Min_S=15 N_DT=10	0.67	0.71	0.63
Crit=log_loss D=10 Min_S=2 N_DT=200	0.80	0.86	0.75
Crit=entropy D=5 Min_S=5 N_DT=200	0.80	0.86	0.75
Crit=log_loss D=10 Min_S=5 N_DT=10	0.93	1.00	0.88

TABLE V

TOP 5 MELHORES CONFIGURAÇÕES POR MODELO PARA A TAREFA DE CLASSIFICAÇÃO E O SEU DESEMPENHO EM ACURÁCIA, *recall* E PRECISÃO. MODELOS TREINADOS EM DADOS DE OLHOS FECHADOS APENAS.

III. RESULTADOS

Para cada experimento rodado, houve uma grande quantidade de modelos executados. Para simplificar a apresentação dos resultados, foram informados os cinco melhores modelos avaliados durante o treino. Devido a limitação de espaço, foi necessário utilizar abreviações para representação dos resultados. As Tabelas V-VI podem ser lidas com o auxílio da Tabela IV. É possível encontrar a Tabela V. A Tabela V apresenta os 5 melhores modelos treinados com dados apenas de olhos fechados. A Tabela VI apresenta os 5 melhores modelos treinados com dados apenas de olhos abertos. A Tabela VII apresenta os 5 melhores modelos treinados com dados de olhos fechados e abertos.

Quando se trata do modelo SVM, notou-se que os experimentos que utilizavam o kernel do tipo *sigmoid* apresentaram uma tendência em sofrer *overfitting*. Na Tabela VII é possível observar que embora durante o treinamento houve três modelos que se destacaram com o uso do kernel *sigmoid*, os valores

OLHOS ABERTO			
Configuração	Acurácia	Recall	Precisão
SVM			
C=0.1 Gamma=0.1 Kernel=rbf	0.80	0.87	0.76
C=100 Gamma=0.1 Kernel=rbf	0.83	0.87	0.81
C=100 Gamma=0.1 Kernel=rbf	0.80	0.87	0.76
C=0.1 Gamma=0.1 Kernel=rbf	0.83	0.87	0.81
C=1 Gamma=0.1 Kernel=rbf	0.83	0.87	0.81
KNN			
Algm= kd_tree Dist=minkowski P=dist	0.83	0.67	1.00
Algm= ball_tree Dist=euclidian P=dist	0.83	0.67	1.00
Algm= ball_tree Dist=minkowski P=dist	0.83	0.67	1.00
Algm= kd_tree Dist=euclidian P=dist	0.83	0.67	1.00
Algm= brute Dist=minkowski P=dist	0.83	0.67	1.00
Random Forest			
Crit=gini D=5 Min_S=2 N_DT=10	0.70	0.73	0.68
Crit=gini D=Sem Min_S=15 N_DT=10	0.70	0.87	0.65
Crit=log_loss D=5 Min_S=15 N_DT=100	0.83	0.81	0.81
Crit=gini D=20 Min_S=15 N_DT=200	0.83	0.87	0.81
Crit=entropy D=5 Min_S=2 N_DT=100	0.83	0.87	0.81

TABLE VI

TOP 5 MELHORES CONFIGURAÇÕES POR MODELO PARA A TAREFA DE CLASSIFICAÇÃO E O SEU DESEMPENHO EM ACURÁCIA, *recall* E PRECISÃO. MODELOS TREINADOS EM DADOS DE OLHOS ABERTOS APENAS.

OLHOS FECHADOS E ABERTOS			
Configuração	Acurácia	Recall	Precisão
SVM			
C=100 Gamma=scale Kernel=sigmoid	0.54	0.56	0.54
C=1 Gamma=scale Kernel=rbf	0.80	0.78	0.82
C=100 Gamma=0.01 Kernel=sigmoid	0.50	0.48	0.50
C=1 Gamma=0.01 Kernel=sigmoid	0.50	0.48	0.50
C=10 Gamma=0.01 Kernel=linear	0.83	0.87	0.80
KNN			
Algm=ball_tree Dist=euclidian P=dist	0.87	0.83	0.90
Algm=kd_tree Dist=euclidian P=dist	0.87	0.83	0.90
Algm=ball_tree Dist=minkowski P=dist	0.87	0.83	0.90
Algm=brute Dist=minkowski P=dist	0.87	0.83	0.90
Algm=kd_tree Dist=minkowski P=dist	0.87	0.83	0.90
Random Forest			
Crit=gini D=20 Min_S=2 N_DT=10	0.80	0.74	0.85
Crit=gini D=10 Min_S=15 N_DT=10	0.89	0.91	0.88
Crit=gini D=5 Min_S=5 N_DT=10	0.85	0.87	0.83
Crit=gini D=10 Min_S=2 N_DT=10	0.87	0.78	0.95
Crit=log_loss D=Sem Min_S=15 N_DT=10	0.85	0.87	0.83

TABLE VII

TOP 5 MELHORES CONFIGURAÇÕES POR MODELO PARA A TAREFA DE CLASSIFICAÇÃO E O SEU DESEMPENHO EM ACURÁCIA, *recall* E PRECISÃO. MODELOS TREINADOS EM AMBOS OS DADOS DE OLHOS FECHADOS E ABERTOS.

obtidos durante o teste eram extremamente baixos. Quando o kernel era de um diferente método, a acurácia aumentava significativamente.

Em todos os experimentos, com exceção de olhos fechados, destacou o uso da distância entre os diferentes pontos como critério de peso em questão de empate para os modelos KNN. Isso possivelmente está relacionado a natureza dos dados. O dado possui uma relação voltada ao tempo e a frequência, de forma que cada "feature" representa um momento no tempo. Os resultados para dados do tipo de olhos aberto ou que possui ambos, apresenta um impacto pequeno dos parâmetros em relação dos resultados. As diferentes combinações apresentavam os mesmos resultados para os testes.

Enquanto seja possível encontrar comportamentos similares dentro de diferentes conjuntos ao analisar os resultados do modelo *Random Forest*, é difícil generalizar o seu comportamento. O experimento apresentado pela Tabela VII mostra uma repetição do uso de impureza de Gini como métrica de avaliação de qualidade. Além disso, todos os ensemble eram compostos de dez DT. Esse comportamento não se repetia nos outros experimentos.

Considerando os diferentes modelos por experimento, o modelo que mais se destacou em questão de acurácia foi o *Random Forest*, atingindo 93% em uma das configurações para dados de olhos fechados. Também se destacou em questão dos experimentos com dados de olhos abertos e fechados, chegando em 89% em uma das configurações.

IV. CONCLUSÃO

O uso de modelos de aprendizado de máquinas se mostram capazes de identificar se um indivíduo teve um sono normal ou se sofreu privação de sono. O *Random Forest* se destaca em relação a acurácia, atingindo 93% quando utiliza dados ambos de olhos fechados quanto de olhos abertos.

Não foi possível identificar comportamentos específicos para hiperparâmetros dos diferentes modelos, com exceção do tipo de Kernel do SVM e o peso dos pontos de KNN. Com o Kernel *sigmoid*, o modelo sofria facilmente *overfitting*. Já a respeito do KNN, devido a natureza do dado apresentar uma continuidade entre as sua "features", o uso da distância como critério de peso influenciou positivamente o desempenho do modelo.

O dado EEG apresenta uma alta complexidade, demandando de um trabalho multidisciplinar com especialistas para melhor uso dentro da inteligência computacional. Neste estudo, focamos na utilização do sinal e na aplicação de transformações, porém há diversas outras abordagens que poderiam ser aplicadas. Métodos como extração ou seleção de *features* demanda um maior conhecimento a respeito do estudo de ondas e de atividade cerebral de determinados canais. Para trabalhos futuros, com a representação e do dados mais sofisticada, seria interessante repetir o experimento da busca de grade e adicionando ao estudo diferentes redes neurais.

REFERENCES

- [1] Institute of Medicine (US) Committee on Sleep Medicine and Research, *Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem*. National Academies Press (US): Institute of Medicine (US) Committee on Sleep Medicine and Research, 2006, ch. 3. Extent and Health Consequences of Chronic Sleep Loss and Sleep Disorders, p. Page Range.
- [2] S. Garbarino, P. Lanteri, N. L. Bragazzi, N. Magnavita, and E. Scoditti, "Role of sleep deprivation in immune-related disease risk and outcomes," *Commun. Biol.*, vol. 4, no. 1, p. 1304, Nov. 2021.
- [3] G. Medic, M. Wille, and M. E. Hemels, "Short- and long-term health consequences of sleep disruption," *Nat. Sci. Sleep*, vol. 9, pp. 151–161, May 2017.
- [4] D. Petit, J.-F. Gagnon, M. L. Fantini, L. Ferini-Strambi, and J. Montplaisir, "Sleep and quantitative eeg in neurodegenerative disorders," *Journal of Psychosomatic Research*, vol. 56, no. 5, pp. 487–496, 2004.
- [5] T. Davenport and R. Kalakota, "The potential for artificial intelligence in healthcare," *Future Healthc. J.*, vol. 6, no. 2, pp. 94–98, Jun. 2019.
- [6] J. Bajwa, U. Munir, A. Nori, and B. Williams, "Artificial intelligence in healthcare: transforming the practice of medicine," *Future Healthc. J.*, vol. 8, no. 2, pp. e188–e194, Jul. 2021.
- [7] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (eeg) classification tasks: a review," *Journal of Neural Engineering*, vol. 16, no. 3, p. 031001, apr 2019.
- [8] Z. Khademi, F. Ebrahimi, and H. M. Kordy, "A review of critical challenges in mi-bci: From conventional to deep learning methods," *Journal of Neuroscience Methods*, vol. 383, p. 109736, 2023.
- [9] I. Lambert and L. Peter-Derex, "Spotlight on sleep stage classification based on EEG," *Nat. Sci. Sleep*, vol. 15, pp. 479–490, Jun. 2023.
- [10] C. Li, Y. Qi, X. Ding, J. Zhao, T. Sang, and M. Lee, "A deep learning method approach for sleep stage classification with EEG spectrogram," *Int. J. Environ. Res. Public Health*, vol. 19, no. 10, p. 6322, May 2022.
- [11] C. Xiang, X. Fan, D. Bai, K. Lv, and X. Lei, "'a resting-state eeg dataset for sleep deprivation,'" 2024.
- [12] E. W. Weisstein, *Fourier Transform*. MathWorld—A Wolfram Web Resource., accessed 2024, online book, Wolfram Research, Inc.
- [13] —, *Fourier Series*. MathWorld—A Wolfram Web Resource., accessed 2024, online book, Wolfram Research, Inc.
- [14] J. O. Smith, *Spectral Audio Signal Processing*. <http://ccrma.stanford.edu/ios/sasp/> / <http://ccrma.stanford.edu/~ios/sasp/>, accessed 2024, online book, 2011 edition.
- [15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995.
- [16] A. Shmilovici, *Support Vector Machines*. Boston, MA: Springer US, 2005, pp. 257–276.
- [17] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Ann. Transl. Med.*, vol. 4, no. 11, p. 218, Jun. 2016.
- [18] Y.-Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Arch. Psychiatry*, vol. 27, no. 2, pp. 130–135, Apr. 2015.
- [19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.
- [20] *Avoiding Overfitting of Decision Trees*. London: Springer London, 2007, pp. 119–134.
- [21] K. A. I. Aboalayon and M. Faezipour, "Multi-class svm based on sleep stage identification using eeg signal," in *2014 IEEE Healthcare Innovation Conference (HIC)*, 2014, pp. 181–184.
- [22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.