



UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ
ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS
CIÊNCIA DA COMPUTAÇÃO
(BACHARELADO)

**IDENTIFICAÇÃO DE ESFORÇO COGNITIVO COM AUXÍLIO DE DISPOSITIVOS
VESTÍVEIS DE QUALIDADE COMERCIAL E INTELIGÊNCIA ARTIFICIAL**

MATEUS ANTONIO FRANCESCHINA

CHAPECÓ, NOVEMBRO DE 2024

UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ
ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS
CIÊNCIA DA COMPUTAÇÃO
(BACHARELADO)

IDENTIFICAÇÃO DE ESFORÇO COGNITIVO COM AUXÍLIO DE DISPOSITIVOS
VESTÍVEIS DE QUALIDADE COMERCIAL E INTELIGÊNCIA ARTIFICIAL

Relatório do Trabalho de Conclusão de Curso
submetido à Universidade Comunitária da
Região de Chapecó para obtenção do título
de bacharelado no curso de Ciência da Com-
putação.

MATEUS ANTONIO FRANCESCHINA

Orientador: Prof. Dr. Felipe André Zeiser

CHAPECÓ, NOVEMBRO DE 2024

**IDENTIFICAÇÃO DE ESFORÇO COGNITIVO COM AUXÍLIO DE DISPOSITIVOS
VESTÍVEIS DE QUALIDADE COMERCIAL E INTELIGÊNCIA ARTIFICIAL**

MATEUS ANTONIO FRANCESCHINA

**ESTE RELATÓRIO, DO TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO
ADEQUADO PARA OBTENÇÃO DO TÍTULO DE:**

BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. Felipe André Zeiser
Orientador

BANCA EXAMINADORA:

Marcos Antônio Moretto, Dr.
Membro da banca

Giancarlo Dondoi Salton, Dr.
Membro da banca

Felipe André Zeiser, Dr.
Supervisor de TCC

Sandro Silva de Oliveira, Me.
Coordenador de Curso

CHAPECÓ, NOVEMBRO DE 2024

RESUMO

Este trabalho dedica-se à elaboração de um sistema de detecção de esforço cognitivo através de arquiteturas de inteligência artificial e dados psicofisiológicos coletados por dispositivos vestíveis comuns, amplamente disponíveis no mercado. Inicialmente, o conceito de esforço cognitivo refere-se a aplicação mental de algum indivíduo na realização de determinada tarefa, a qual exige certo grau de conhecimento específico aplicado. Dentre as diversas formas em que se pode identificar esforço cognitivo, o uso de indicadores psicofisiológicos destaca-se por tornar este processo independente da participação direta do sujeito avaliado ou de algum observador avaliando o desempenho na execução da tarefa em questão. Estes indicadores carregam informações sobre os sistemas internos do corpo humano, como volume de sangue em determinada região através de um sensor PPG, potencial de condutibilidade elétrica na pele por EDA ou a atividade de células no cérebro através de EEG. A partir destes sensores, uma série de processos foram aplicados para limpar os dados de possíveis interferências e artefatos para então extrair características descritivas em determinado período de tempo, com o resultado deste processo sendo direcionado para os modelos tradicionais de aprendizado de máquina SVM, KNN e GBDT. Ademais, outros modelos de aprendizado profundo, sendo eles CNN, FCN, LSTM e ResNet, foram aplicados em um fluxo sem intervenção humana nos sinais coletados, denominado aprendizado de ponta a ponta, no qual os modelos foram responsáveis por, internamente, remover ruídos dos sinais e extrair características relevantes de cada sinal, reduzindo, portanto, o viés de características geradas manualmente além da necessidade de conhecimento específico de processamento de sinais e cada indicador psicofisiológico em particular. Finalmente, todos os modelos foram testados para detecção de esforço cognitivo em dados de três voluntários a fim de comparar a capacidade de generalização para cada modelo. Concluindo, destaca-se a capacidade de modelos treinados em dados de dispositivos comerciais em identificar esforço cognitivo, especialmente através de engenharia de características, mas também através do aprendizado de ponta a ponta, o que contribui para o tanto para o monitoramento ativo deste estado mental, quanto para outras pesquisas de detecção de emoções através de sinais semelhantes.

Palavras-chave: Esforço Cognitivo. Inteligência Artificial. PPG. EEG. EDA.

ABSTRACT

This research aims the development of a cognitive load detection system through artificial intelligence architectures and psychophysiological data collected from common wearable devices, largely available on the market. Firstly, the concept of cognitive load refers to one's mental application for the resolution to any given task that requires certain degree of domain specific knowledge related to it. Among the several techniques employed on identifying cognitive load, the use of psychophysiological measures is highlighted due to its capacity for removing the need of direct intervention from the subject under evaluation or from any assessment reported from a trained practitioner, whom would be responsible to monitor the task's performance. These measurements provide us information about the human physiology, for example, the blood volume on a previously determined region through a PPG sensor, electrodermal activity with EDA or neuronal cells activation by means of EEG. Thenceforth, given the gathered sensor data, a series of processes were applied to clear out the data from any noise or possible artifacts in order to, then, extract features descriptive from a determined time window, piping this result into training traditional machine learning models, such as SVM, KNN and GBDT. Moreover, additional deep learning models, being CNN, FCN, LSTM and ResNet, were implemented on an end to end workflow, where there is no human intervention on the collected signal data, therefore holding the models responsible for, internally by its own means, handle signal noise and extract meaningful features for each given signal, as a consequence, reducing both the bias generated by manually selecting features, but also the requirement of specialized knowledge on signal processing and on the physiological meaning for each of the used measures. Lastly, all the models were tested for cognitive load detection on the data from three volunteers in order to compare their generalization capabilities. In conclusion, emphasizes the performance of the models trained on consumer grade wearables data on detecting cognitive load, specially throughout the feature engineering process, but also from the end to end approach, which contributes both for the monitoring of this specific mental state and also for further researches on emotion detection with similar signals.

Keywords: Cognitive Load. Artificial Intelligence. PPG. EEG. EDA.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Comparação entre aprendizado de ponta a ponta com engenharia de características | 16 |
| Figura 2 – Visão da estrutura elaborada | 21 |
| Figura 3 – Divisão do conjunto de dados | 24 |
| Figura 4 – Sinal Empatica EDA | 30 |
| Figura 5 – Sinal Empatica PPG | 32 |
| Figura 6 – Sinal Samsung PPG | 32 |
| Figura 7 – Sinal Muse EEG | 33 |
| Figura 8 – Organização de classes desenvolvidas | 42 |
| Figura 9 – Resultado FCN EEG para primeiro participante de teste | 43 |
| Figura 10 – Resultado FCN EEG para segundo participante de teste | 44 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Resumo dos dispositivos | 9 |
| Tabela 2 – Características por domínio | 14 |
| Tabela 3 – Resumo dos dados | 31 |
| Tabela 4 – Resumo dos modelos tradicionais | 38 |
| Tabela 5 – Resumo dos modelos | 40 |
| Tabela 6 – Resumo dos resultados | 41 |

LISTA DE ABREVIATURAS

- ACC** aceleração. 22
- BPM** batimentos por minuto. 36
- BVP** volume de pulso sanguíneo. 22, 23
- CNN** *convolutional neural network*. iv, v, 12, 13, 15, 20, 26, 38
- CNN-LSTM** *convolutional neural network long-short term memory*. 15, 39
- ECG** eletrocardiograma. 6, 18
- EDA** atividade elétrica cutânea. iv, v, 6–9, 18, 22, 30, 34, 35, 44, 46
- EEG** eletroencefalograma. iv, v, 7–9, 22, 29–32, 34, 35, 37–39, 42
- EMG** eletromiografia. 18
- FCN** *fully convolutional network*. iv, v, 12, 13, 15, 20, 26, 38, 42
- fMRI** *functional magnetic resonance imaging*. 6
- FN** falso negativo. 27
- fNIRS** *functional near infrared spectroscopy*. 6
- FP** falso positivo. 27
- GAP** *Global Average Pooling*. 39, 40
- GBDT** *gradient boosting decision tree*. iv, v, 10, 11, 17, 20, 25, 37
- HOC** *High-Order Crossing*. 14
- HR** ritmo cardíaco. 6, 7, 23, 34
- HRV** variabilidade do ritmo cardíaco. 6, 7, 22, 23
- IBI** intervalo entre batimentos. 22, 23, 36
- ICA** análise de componente independente. 14
- IIR** *infinite impulse response*. 35
- KNN** *k-nearest neighbors*. iv, v, 10, 11, 20, 25, 37

LOSOVC *leave-one-subject-out cross validation*. 17, 20, 23, 37, 39, 40

LSTM *long-Short term memory*. iv, v, 13, 20, 26, 39

MCDCNN *multi channel deep convolutional neural network*. 16

MLP *perceptron multicamada*. 15, 17

MLP-LSTM *multilayer perceptron long short term memory*. 16

MT-CNN *multi task convolutional neural network*. 15

PCA *análise de componente principal*. 15

PPG *fotopletismografia*. iv, v, 6, 8, 9, 17, 22, 23, 30, 34, 36, 42, 44, 46

PSD *espectro de densidade de energia*. 14, 36

ReLU *Rectified Linear Unit*. 13, 38, 39

ResNet *Residual Network*. iv, v, 13, 15, 20, 27, 39

RNN *Recurrent Neural Network*. 13

SCL *nível de condutividade cutânea*. 7, 8

SCR *resposta de condutividade cutânea*. 8, 35

ST-CNN *single task convolutional neural network*. 15

SVM *support vector machine*. iv, v, 10, 11, 20, 25, 37

TEMP *temperatura da pele*. 22

Time-CNN *time convolutional neural network*. 16, 38

TN *verdadeiro negativo*. 27

TP *verdadeiro positivo*. 27

TPE *tree-structured Parzen Estimator*. 24

WL *comprimento de onda*. 14

SUMÁRIO

| | | |
|------------------|--|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Contextualização | 1 |
| 1.2 | Delimitação do problema | 2 |
| 1.3 | Objetivos | 2 |
| 1.3.1 | Objetivo geral | 2 |
| 1.3.2 | Objetivos específicos | 2 |
| 1.4 | Justificativa | 2 |
| 1.5 | Procedimentos metodológicos | 3 |
| 1.6 | Estrutura do trabalho | 4 |
| 2 | REVISÃO BIBLIOGRÁFICA | 5 |
| 2.1 | Esforço cognitivo | 5 |
| 2.2 | Indicadores psicofisiológicos | 6 |
| 2.2.1 | Uso para medir esforço cognitivo | 7 |
| 2.2.2 | Sensores utilizados na coleta | 8 |
| 2.3 | Técnicas de inteligência artificial | 9 |
| 2.3.1 | Aprendizado de máquina | 9 |
| 2.3.1.1 | Aprendizado tradicional | 10 |
| 2.3.1.1.1 | SVM | 11 |
| 2.3.1.1.2 | KNN | 11 |
| 2.3.1.1.3 | GBDT | 11 |
| 2.3.1.2 | Aprendizado profundo | 11 |
| 2.3.1.2.1 | Convolucionais | 12 |
| 2.3.1.2.2 | Recorrente | 13 |
| 2.3.1.2.3 | Residual | 13 |
| 2.3.2 | Engenharia de características | 14 |
| 2.3.3 | Aprendizado de ponta a ponta | 15 |
| 2.3.4 | Validação | 16 |
| 2.4 | Trabalhos relacionados | 17 |
| 2.4.1 | Lacuna de pesquisa | 18 |
| 2.5 | Considerações do capítulo | 19 |
| 3 | MATERIAIS E MÉTODOS | 20 |
| 3.1 | Conjunto de dados | 20 |
| 3.1.1 | Indicadores coletados | 22 |
| 3.2 | Pré-processamento | 22 |
| 3.3 | Treinamento | 23 |

| | | |
|-------|--|----|
| 3.3.1 | Espaço de busca de hiperparâmetros | 24 |
| 3.4 | Avaliação | 27 |
| 3.5 | Considerações do capítulo | 28 |
| 4 | RESULTADOS E DISCUSSÃO | 29 |
| 4.1 | Análise Exploratória de Dados | 29 |
| 4.2 | Pré-processamento | 34 |
| 4.2.1 | EDA | 35 |
| 4.2.2 | EEG | 35 |
| 4.2.3 | PPG | 35 |
| 4.3 | Otimização de hiperparâmetros | 37 |
| 4.4 | Aprendizado tradicional | 37 |
| 4.5 | Aprendizado de ponta a ponta | 38 |
| 4.6 | Testes de modelos | 40 |
| 4.7 | Discussão | 40 |
| 4.8 | Contribuições | 44 |
| 5 | CONSIDERAÇÕES FINAIS | 46 |
| | REFERÊNCIAS | 47 |

1 INTRODUÇÃO

Neste capítulo, o projeto de identificação de esforço cognitivos será descrito através do contexto o qual está inserido, as presentes dificuldades para a classificação do mesmo, a meta de uma aplicação para resolver estas dificuldades e a motivação para elaboração da mesma, além das etapas realizadas para a sua construção.

1.1 Contextualização

Também referenciado como esforço ou carga mental, o esforço cognitivo é definido como uma construção multidimensional, a qual representa a carga de uma tarefa imposta ao sistema cognitivo de um indivíduo (Paas; Merriënboer, 1994). Ou seja, pode ser entendido como um estado mental, atingido ao realizar alguma tarefa que possua certo grau de exercício mental para sua realização, seja este esforço direto ou indiretamente atrelado à tarefa. Ainda, o grau de esforço cognitivo deriva da capacidade do sujeito com a demanda da tarefa, de modo que proficiência e experiência na tarefa possa reduzir a sua quantidade. No segundo capítulo, este conceito será explorado em mais detalhes.

Atualmente, existem quatro formas principais para estimar o esforço cognitivo, sendo elas: Avaliação subjetiva; Performance da tarefa; Comportamento; e Métricas psicofisiológicas (Paas et al., 2003). No entanto, conforme relatado por Fleming, Robinson e Roiser (2023), grande parte da pesquisa sobre estes métodos de mensuração do esforço cognitivo, até o presente momento, dedicou-se a formas que dependem da realização de alguma tarefa específica ou profissionais qualificados, algo inviável para realização em escala ou de modo automático por qualquer indivíduo. Como solução para este problema, e devido a maior disponibilidade computacional, métricas psicológicas ganham força no uso em detecção de esforço cognitivo. Estas métricas consistem em mudanças inconscientes no indivíduo, como movimento dos olhos, dilatação da pupila, resposta galvânica da pele, medidas de eletromiografia e eletroencefalografia, variação de batimentos cardíacos entre outros para identificação do esforço cognitivo (Cinaz, 2013).

Neste cenário, o presente trabalho busca elaborar um sistema capaz de utilizar dados de sensores instalados em dispositivos vestíveis de qualidade comercial, responsáveis pela coleta de métricas psicofisiológicas, para a inferência do estado mental do indivíduo em esforço cognitivo ou não. De forma a realizar a identificação, serão utilizadas tecnologias de inteligência artificial, através tanto de algoritmos tradicionais de classificação, quanto ao uso de modelos de aprendizado profundo, os quais permitem formar um vínculo entre medidas fisiológicas do usuário e o esforço cognitivo, para classificar se no presente momento o sujeito executa alguma tarefa mentalmente exigente.

1.2 Delimitação do problema

A detecção de esforço cognitivo, por muito tempo, ocorreu exclusivamente pelos métodos subjetivos e comparação de performance, no entanto, para a avaliação em tempo real é necessário o uso de métricas psicofisiológicas, até o presente momento, pouco estudadas (Ayres; Paas, 2012). Apesar de haver uma orientação de quais métricas são alteradas em situações de alto esforço cognitivo, não há amplo entendimento de todos os outros fatores externos que também podem estar envolvidos (Conway et al., 2013).

Ademais, o fato de humanos também terem dificuldade em aferir seu estado emocional representa um desafio por si só, a compreensão de mecanismos envolvido no sentimento de emoções não é claro e possui variações individuais. Deste modo, acaba por ser ainda mais desafiador associar os dados coletados de sensores ao estado psicológico do indivíduo, haja vista que isso não ocorre de forma explícita, como é com estados físicos, por exemplo diferenciar corrida de uma caminhada (Gjoreski et al., 2021).

1.3 Objetivos

1.3.1 Objetivo geral

Elaborar um modelo de inteligência artificial para detecção de esforço cognitivo em tempo real através de dados psicofisiológicos coletados por dispositivos vestíveis de qualidade comercial.

1.3.2 Objetivos específicos

- Realizar uma análise exploratória dos dados coletados;
- Treinar algoritmos de aprendizado profundo a partir dos dados puros coletados;
- Identificar características descritoras de esforço cognitivo;
- Treinar algoritmos de inteligência artificial através de dados pré-processados;
- Avaliar diferentes modelos de inteligência artificial e sua performance;

1.4 Justificativa

Desde o último século, a humanidade vem passando por uma grande mudança de paradigma da forma de mão de obra, partindo de linhas de produção completamente dependentes de trabalho manual para o ganho de produtividade e escala oriundo do ganho de conhecimento e capacitação pessoal. *Knowledge worker* é o termo utilizado para referir-se ao novo perfil do profissional moderno, voltado a tarefas muito menos manuais e mais abstratas (Drucker, 2006).

Além disso, a expansão da globalização no mundo causou um aumento na demanda de esforço cognitivo devido a intensa carga horária e pressão para cumprir prazos, possibilitando um cenário de estresse crônico, exaustão ou até depressão ocasionados pelo acúmulo de trabalho mental sem o descanso apropriado (Cinaz, 2013). Ademais, neurotransmissores, como dopamina, demonstram afetar a disposição de um indivíduo em engajar em atividades com maior demanda cognitiva, neste cenário, o monitoramento de esforço cognitivo pode tanto prestar auxílio na avaliação da performance de psicostimulantes catecolaminérgicos, quanto em identificar a necessidade do uso de alguma substância do tipo (Westbrook et al., 2020).

O monitoramento diário de esforço cognitivo pode contribuir para detecção de doenças que comprometem o sistema cognitivo, evitar acidentes no meio por profissionais exaustos mentalmente além de melhorar a produtividade de trabalhadores ao bloquear intervenções em momentos de foco (Gjoreski et al., 2021). Para isto ser possível, faz-se necessário o desenvolvimento de um sistema que não dependa de avaliações subjetivas ou de um profissional qualificado, assim justificando a utilização de dispositivos vestíveis aliados à inteligência artificial.

1.5 Procedimentos metodológicos

Para a execução do trabalho, inicialmente foi conduzida uma pesquisa bibliográfica em livros e artigos sobre os efeitos de esforço cognitivo nas características psicofisiológicas de um indivíduo. Esta busca teve como objetivo identificar quais métricas podem ser utilizadas e quais indicadores podem ser derivados dos dados coletados para gerar melhores resultados e compreensão do estado de esforço cognitivo.

O conjunto de dados *CogWear*¹, disponibilizado por Grzeszczyk et al. (2023), foi selecionado para esta pesquisa. Técnicas de exploração de dados foram empregadas para avaliar a qualidade das informações obtidas, classificando quais seriam os desafios para extração de características relevantes e gerando elementos visuais para compreensão da qualidade dos sensores, além de elaborar um fluxo de tratamento dos dados.

Posteriormente, para o treinamento dos algoritmos de inteligência artificial, dois processos distintos foram elaborados com base em desenvolvimentos de bibliografias relacionadas. O primeiro conta com uma etapa de engenharia de dados, realizando filtros para remoção de ruído e artefatos dos sinais coletados com o objetivo de extrair informações descritivas do estado psicofisiológico em janelas de tempo. Enquanto no segundo processo, modelos de aprendizado profundo serão responsabilizados por realizar a detecção de esforço cognitivo a partir dos dados sem nenhum pré-processamento aplicado, no denominado aprendizado de ponta a ponta. Em ambos os processos, a seleção dos algoritmos utilizados ocorreu com base nos modelos utilizados em trabalhos relacionados. Para a busca destas informações, termos como *wearable sensors*, *affective computing*, *emotion detection* e *cognitive load detection* foram usados para pesquisa em ferramentas de busca de artigos acadêmicos.

¹ Disponível em: <<https://physionet.org/content/consumer-grade-wearables/1.0.0/>>. Acesso em: 01 mar. 2024

1.6 Estrutura do trabalho

Este projeto conta com uma revisão de literatura no segundo capítulo, descrevendo, primordialmente, esforço cognitivo como um conceito de estado mental, formas de quantificar esforço cognitivo e detalhando o uso de métricas psicofisiológicas para a medição. Na sequência, é conduzida uma análise de indicadores psicofisiológicos e sua relação com esforço cognitivo, além de explicar o processo de coleta destas informações, listando quais métricas foram coletadas e com quais aparelhos. Após isto, tópicos de inteligência artificial são abordados, inicialmente pela análise do funcionamento de sistemas de aprendizado de máquina e os diferentes algoritmos utilizados, para detalhar, em sequência, o fluxo de uso destes algoritmos em um processo de engenharia de características e aprendizado de ponta a ponta e então passando por métodos de avaliação de performance dos modelos. Concluindo a revisão, alguns trabalhos relacionados e sua lacuna com este projeto são explorados, finalizando com considerações finais.

Posteriormente, é contemplado no terceiro capítulo uma descrição dos métodos e tecnologias utilizadas para a elaboração do sistema proposto. Esta descrição inicia-se pelo conjunto de dados utilizado, entendendo quais foram os aparelhos utilizados para a coleta das informações e a composição dos voluntários desta coleta. Na sequência é apresentado o processo de pré-processamento e quais técnicas podem ser utilizadas para extrair mais informações a partir dos dados. Seguinte, o modo de treinamento e avaliação dos modelos são elaborados, finalizando por considerações finais sobre este capítulo.

Então, no quarto capítulo é relatado o processo de implementação da solução de detecção de esforço cognitivo, passando pelos detalhes de exploração e visualização dos dados obtidos, compilando métricas e figuras para aferir a qualidade das informações. Tendo este entendimento, é detalhado quais foram as técnicas para pré-processamento dos dados, para cada tipo distinto de sinal utilizado. A partir deste momento, iniciou-se o trabalho com algoritmos de inteligência artificial, primeiramente com um processo para busca dos melhores hiperparâmetros para cada modelo, seguindo, então, para o treinamento através de aprendizado tradicional e de ponta a ponta. Finalmente, os melhores modelos para cada algoritmo e sinal foram comparados e os resultados discutidos, ressaltando detalhes sobre o comportamento dos algoritmos ao classificar dados novos e sumarizando as contribuições feitas pelo trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo é direcionado à fundamentação teórica do projeto. Será explorado o conceito de esforço cognitivo e como pode ser feita a medição do mesmo, com ênfase em métricas psicofisiológicas. Ademais, alguns indicadores psicofisiológicos utilizados para reconhecimento de esforço cognitivo são exemplificados, em conjunto dos sensores utilizados para coleta. Na sequência é elaborada uma análise do uso de inteligência artificial no processo de detecção de esforço cognitivo. Finalmente, apresenta-se trabalhos relacionados a este, seguinte pelas considerações finais do capítulo.

2.1 Esforço cognitivo

Originalmente, a teoria de esforço cognitivo buscou compreender quais são os métodos mais eficientes que podem ser utilizados para transferir ou obter conhecimento, dado em um contexto no qual recursos mentais para a aquisição de novas informações são limitados. Este cenário baseia-se em uma arquitetura mental que divide o intelecto humano em duas fontes, simplesmente referenciadas por primária e secundária, onde esta é o foco do estudo de esforço cognitivo e consiste de todo o conhecimento necessário ou desenvolvido a partir de um contexto social, enquanto aquela trata do conhecimento que somos preparados biologicamente para obter durante evolução como espécie. Nesta definição de organização mental, nós como espécie, possuímos uma memória de trabalho limitada e com curta duração que interage com a memória de longo prazo, perceptivelmente ilimitada, no sentido de que representa assuntos que temos familiaridade e não é possível definir um limite de armazenamento ou de tempo para busca das informações mantidas lá (Sweller; Ayres; Kalyuga, 2011).

Dado o conceito, as primeiras tentativas de quantificar este estado mental ocorreram pela construção de modelos computacionais, os quais imitassem o processo de aprendizado, de modo que quanto mais complexo fosse a busca que o modelo executasse, maior seria considerado o esforço cognitivo naquela tarefa. Futuramente, outras técnicas foram estabelecidas, que mediam o esforço cognitivo através de experimentos ou uma autoavaliação do indivíduo que executou a tarefa. No entanto, estes métodos dependem de uma medição subjetiva e não poderiam ser conduzidos de forma simultânea à tarefa (Sweller; Ayres; Kalyuga, 2011).

Nesta definição, o grau de esforço cognitivo na realização de alguma tarefa é composto por dois fatores principais, intrínseco, diretamente relacionado a estrutura da tarefa, e alheio, o qual refere-se a carga adicionada pela forma como a tarefa ocorre, ou seja, seu ambiente. Portanto, a distinção entre ambos em experimentos para avaliar a carga de esforço cognitivo ocorre ao manter um dos fatores constante. Nestes cenários, os voluntários teriam sua performance avaliada, seja na tarefa principal ou alguma secundária caso exista, através de formulários para avaliação subjetiva (Sweller; Ayres; Kalyuga, 2011).

Apenas no início do século XXI iniciou-se o uso de indicadores fisiológicos para medir o grau de esforço cognitivo. Nesse contexto foram desenvolvidos estudos sobre reação pupilar proporcional ao uso de memória, ressonância magnética funcional e eletroencefalograma detectando ondas cerebrais, rastreamento do movimento dos olhos e variação do movimento, complexidade de fala e densidade léxica entre outros. Não obstante, todos estes métodos são prejudicados, em maior ou menor grau, na coleta de dados para inferência da presença de esforço cognitivo no dia a dia por não existir um equipamento de uso corriqueiro para fazer a coleta destas informações, o que vai contra o propósito do uso destes indicadores (Sweller; Ayres; Kalyuga, 2011).

2.2 Indicadores psicofisiológicos

Por muito tempo as técnicas para identificar esforço cognitivo dependiam exclusivamente de avaliações subjetivas, isto é, restritas a um parecer individual e não necessariamente auditável. Apenas com avanços tecnológicos recentes é que o uso de indicadores psicofisiológicos começou a ter relevância para esta tarefa, com indicadores agrupados em quatro categorias, relacionadas à órgãos do corpo, sendo coração, pele, pulmão e cérebro (Ayres et al., 2021).

Relacionado ao coração, a variabilidade do ritmo cardíaco (HRV) foi demonstrada como capaz em diferenciar níveis de esforço cognitivo. Este valor pode ser entendido como a oscilação do intervalo entre batimentos consecutivos e pode ser obtida através da atividade elétrica do coração com uso de um eletrocardiograma (ECG) ou por diferenças no volume sanguíneo em vasos capilares por meio de um sensor fotopletismografia (PPG). No entanto, o principal problema desta métrica é o fato dela ser influenciada por outras questões como emoções ou atividade física. Ademais, para longos períodos de medição, o ritmo cardíaco (HR) costuma ser mais significativo do que HRV (Ayres et al., 2021).

Em contraste, não há muita pesquisa sobre a influência de esforço cognitivo no sistema respiratório. Assim como para o sistema cardíaco, diversas condições podem influenciar sua medição. Ademais, atividades verbais podem interromper a medição com estes indicadores, o que dificulta o seu uso para monitoramento (Ayres et al., 2021).

De outro modo, indicadores relacionados a pele contam com um longo histórico de uso para aferir estados emocionais, condições de estresse ou de esforço cognitivo. Está informação é coletada através de um sensor de atividade elétrica cutânea (EDA), o qual é influenciado pela secreção de suor, que diminui a resistência elétrica da pele, por sua vez aumentando a condutividade. Este sinal é formado por dois componentes, um tônico, que refere-se à mudanças lentas e reflete uma ativação psicofisiológica e um componente fásico, que mostra mudanças abruptas e está relacionado a situações de estresse ou excitação (Ayres et al., 2021).

Por último, métricas atreladas ao cérebro são comumente presentes em ambientes laboratoriais, devido ao maior requisito técnico de equipamentos, mas que vem sendo flexibilizado com dispositivos mais baratos e menos invasivos para coleta. Dentre as técnicas utilizadas constam *functional magnetic resonance imaging* (fMRI), *functional near infrared spectroscopy* (fNIRS)

e eletroencefalograma (EEG). Com foco em EEG, por ser o sensor utilizado na composição do conjunto de dados em uso neste trabalho, o indicador mede a atividade elétrica do cérebro através de diversos eletrodos posicionados em diferentes regiões do cérebro, medindo oscilações de atividade, agrupadas por frequência. O esforço cognitivo foi observado por influenciar, principalmente, a faixa alpha no lobo parietal e a faixa theta no mesencéfalo frontal, causando um aumento na atividade medida (Ayres et al., 2021).

2.2.1 Uso para medir esforço cognitivo

Devido ao relacionamento dos diferentes sistemas do corpo humano, a observação de mudanças nos indicadores psicofisiológicos pode ser utilizada para identificar o estado de esforço cognitivo. Conforme Ahmad et al. (2023), há achados prévios vinculando uma redução na HRV e aumento no HR ao entrar em estado de esforço cognitivo. Esta relação também podendo ser observada por um aumento de atividades de onda *Theta* na parte fronta do cérebro em uma leitura de EEG ou aumento de nível de condutividade cutânea (SCL), detectável por um sensor EDA.

Contudo, apesar da existência de referências prévias usando estes sinais para classificação de estados mentais, o uso destes indicadores costuma ser superestimado. A razão disto deve-se ao fato de que existe a impressão, na qual o estado mental sempre pode ser corretamente inferido por estas métricas, pensamento que costuma ser originado de conclusões não propriamente fundamentadas e generalizações problemáticas. Ademais, estas características impõem novas complicações relacionadas a diferentes áreas de estudo para validar a medição de esforço cognitivo. Por exemplo, para uma demonstração de qualificação do estado mental bem sucedida, faz-se necessário conhecimento em tecnologia de sensores, processamento de sinais, modelagem matemática, desenho experimental, psicofisiologia, engenharia de sistemas e conhecimento sobre o grupo alvo do estudo (Brouwer et al., 2015).

Em ordem de garantir a qualidade na pesquisa, há seis recomendações no estudo de indicadores psicofisiológicos relacionados a estados mentais a serem tomadas, sendo elas: Definição precisa do estado mental de interesse e quais são suas evidências empíricas; Elaborar hipóteses que vinculam o estado mental as métricas; Eliminar variáveis de confusão; Utilizar análises estatísticas apropriadas; Fornecer explicações sobre o comportamento dos dados; Clarificar o benefício em utilizar medidas psicofisiológicas (Brouwer et al., 2015).

A fim de validar o uso destes indicadores e esclarecer as recomendações propostas por Brouwer et al. (2015), definimos o estado mental a ser identificado como de esforço cognitivo, o qual está sendo estimulado através da realização de um teste mentalmente exigente. Vinculando esforço cognitivo às métricas utilizadas, está a evidência apontada anteriormente por Ahmad et al. (2023), que também deve ser observada neste trabalho. Para eliminar variáveis de confusão, a coleta de dados foi realizada em dois dias diferentes com um intervalo de duas semanas entre eles, também, antes do início da tarefa mensurada, todos os participantes passaram por um período de leitura base em um estado de relaxamento em postura confortável, sendo todos estes fatores contribuintes para isolar o estado de esforço cognitivo de qualquer outro estado emocional, o

qual possa interferir no experimento. Para análises apropriadas, os algoritmos de inteligência artificial serão treinados em dados de dez participantes e posteriormente avaliados em dados novos de outros três voluntários, com a finalidade de garantir a detecção em pessoas distintas. Por estarem em estado de repouso, com objetivo de realização de uma única tarefa, alterações no comportamento psicofisiológicos devem ter originado por alguma mudança na carga de esforço cognitivo. Por último, o benefício do uso de métricas psicofisiológicas para a detecção deste estado mental deve-se ao fato de ser a única maneira de viabilizar o monitoramento desta métrica sem a necessidade de outro profissional qualificado e eliminando fatores subjetivos da qualificação do estado mental.

2.2.2 Sensores utilizados na coleta

A forma de coleta do EDA acontece pela passagem de uma pequena quantidade de corrente elétrica entre dois eletrodos em contato com a pele. O primeiro componente do sinal trata de valores basais e que possuem uma variação lenta sendo sua principal característica o SCL, enquanto o segundo refere-se a mudanças rápidas, com seu principal componente sendo a resposta de condutividade cutânea (SCR) (Boucsein et al., 2012). Neste trabalho, o sensor EDA utilizado foi o equipado no dispositivo *Empatica E4*, um *smartwatch* voltado para uso em pesquisas clínicas.

Sinais relacionados ao coração são coletados por PPG. Esta é uma técnica de coleta bastante conveniente pois possui baixo custo e serve para a realização de medições não invasivas. Seu mecanismo de funcionamento consiste em uma fonte de luz iluminando uma região de pele e um detector luminoso que mede pequenas variações de intensidade na luz, associada a diferença de volume sanguíneo no tecido (Allen, 2007). Apesar da confiabilidade deste sinal poder ser comprometida por conta de artefatos de movimento, quando emitido com luz verde a diferença na relação de sinal-ruído é consideravelmente pequena e apresenta qualidade comparável a métodos mais robustos como eletrocardiograma (Lee et al., 2013). Neste trabalho, dois sensores PPG distintos foram utilizados, um coletado pelo *Empatica E4* e outro pelo *smartwatch Samsung Galaxy Watch4*, um relógio comercial de entrada.

Para o EEG, os eletrodos atuam como antenas medindo campos elétricos que são resultados da sinalização eletroquímica de um neurônio ativando-se e propagando a ativação para outro. Quando centenas de milhões destes sinais ocorrem simultaneamente em grupos de neurônios alinhados espacialmente e geometricamente, em uma posição piramidal, em conjunto com movimentações de cálcio e sódio e ativação de células da glia, então os eletrodos são capazes de identificar a atividade cerebral, que em conjunto de algoritmos de mapeamento anatômico, identifica quais regiões do cérebro foram ativadas (Cohen, 2017). Neste trabalho, foi usado um sensor de contato seco EEG, equipado no dispositivo *Muse S headband*.

Esta atividade detectada pelo EEG é composta por ritmos que são agrupados em um número pequeno de frequências características, ademais, os ritmos originam de flutuações da excitabilidade de populações de neurônios e ocorrem em complexos padrões espaço-temporais.

Tabela 1 – Resumo dos dispositivos

| Aparelho | Sinal | Taxa de coleta | Uso |
|------------------------------|--------------|-----------------------|------------|
| <i>Samsung Galaxy Watch4</i> | PPG | 25Hz | Comercial |
| <i>Empatica E4 wristband</i> | PPG | 64Hz | Clinico |
| <i>Empatica E4 wristband</i> | EDA | 4Hz | Clinico |
| <i>Muse S headband</i> | EEG | 256Hz | Comercial |

Fonte: Elaborado pelo autor.

Interações entre estes ritmos e características dos ritmos podem dar informações sobre os processos neuronais sendo executados no momento. Dados ritmos são costumeiramente classificados em Delta, Theta, Alpha, Beta e Gamma, conforme sua frequência (Siegel; Donner; Engel, 2012).

2.3 Técnicas de inteligência artificial

Historicamente, diversas definições de inteligência artificial foram elaboradas, desde estabelecendo um vínculo entre inteligência como fidelidade à performance humana ou utilizando termos mais abstratos, como racionalidade (Russell; Norvig, 2016). Ademais, o modo em que a inteligência atua também influencia na definição, podendo ser encarado como um processo interno de raciocínio, como um encadeamento de pensamentos, ou através de uma caracterização externa de comportamento inteligente (Russell; Norvig, 2016).

A partir desta definição, quatro combinações podem ser formadas para exemplificar o conceito do que é uma inteligência artificial. Primeiramente, um modelo que tenha um comportamento semelhante ao humano, o qual, por exemplo, é o foco de análise de um Teste de Turing, onde o objetivo é parecer-se com um humano. O segundo é um modelo que pense como um humano, onde deve-se elaborar uma teoria da mente precisa o suficiente para ser expressa em termos computacionais, através de métodos de introspecção, experimentos psicológicos e observação de atividades cerebrais. O terceiro modelo seria uma inteligência que pensa de modo racional, ou seja, elabora estruturas lógicas através de um processo de raciocínio irrefutável, fazendo uso de predicados, sentenças e silogismos. Por último, há o modelo que comporta-se de modo racional, ou seja, um agente que opera de forma autônoma para atingir um objetivo específico, como maximizar lucros em um sistema financeiro (Russell; Norvig, 2016).

2.3.1 Aprendizado de máquina

Para Langley e Simon (1995), aprendizado de máquina é o estudo de métodos computacionais capazes de automatizar a melhoria de performance através do ganho de experiência. Sendo o foco desta área fornecer cada vez maiores níveis de independência no processo de engenharia de informações, ao substituir mão de obra humana por técnicas automáticas que apresentam melhor acurácia e eficiência.

Desta maneira, considera-se que um agente esta aprendendo quando a partir do momento em que sua performance para determinada tarefa aumenta após ser exposto à exemplos, e o uso de modelos computacionais neste processo permite que realizemos tarefas que não necessariamente tenham sido propriamente antecipadas, ou que não possuímos um conhecimento detalhado em como resolver (Russell; Norvig, 2016).

A fim de formalização, algoritmos de aprendizado ganham experiência E em alguma tarefa T , com sua performance na tarefa T mensurada por P , sendo que a performance aumenta com E . Uma tarefa é descrita em como um algoritmo deve processar um conjunto de características medidas quantitativamente. Um tipo de tarefa é a classificação, na qual o algoritmo tem um número específico de grupos a ser encontrado. A performance é um conjunto de métricas que podem ser calculadas com o resultado do algoritmo, como o valor de acurácia ou erro. Por último, a experiência é obtida através de um conjunto de dados, que pode classificar como aprendizado supervisionado ou não supervisionado, onde no primeiro, o conjunto de características é associado à uma categoria, diferentemente do segundo, onde o modelo aprenderá algo relevante relacionado à estrutura dos dados (Goodfellow; Bengio; Courville, 2016).

Ademais, aprendizado de máquina pode ser subdividido em algoritmos tradicionais e algoritmos de aprendizado profundo. Há diversos tipos distintos de algoritmos tradicionais, alguns são baseados em estatística bayesiana, classificando valores conforme uma distribuição de probabilidades, outros são algoritmos lineares, como *support vector machine* (SVM), que busca vetores que separam os dados, arvores de decisão formam uma sequência de regras, separando os dados em diferentes regiões ou algoritmos de vizinhos, os quais categorizam os dados com base na distância, ou similaridade, entre cada informação (Goodfellow; Bengio; Courville, 2016).

Enquanto os algoritmos de aprendizado profundo são compostos por uma vasta gama de técnicas de aprendizado de máquina, as quais podem ser representadas por complexas estruturas algébricas com parâmetros configuráveis. A origem destes algoritmos deve-se à tentativa de replicar a estrutura do cérebro em forma de circuitos, por isso chamados de redes neurais. Estes modelos destacam-se pela capacidade de lidar com dados mais complexos, pois, diferentemente dos modelos tradicionais, onde poucas interações são feitas com os dados de entrada até resultar em uma saída, nas redes neurais, as informações passam por diversas camadas, as quais podem interagir entre si de maneiras diversas (Russell; Norvig, 2016).

2.3.1.1 Aprendizado tradicional

Há três algoritmos de aprendizado tradicional utilizados no trabalho, sendo SVM, *k-nearest neighbors* (KNN) e *gradient boosting decision tree* (GBDT). Todos eles tem um funcionamento distinto, o primeiro é um algoritmo linear, o segundo baseado em vizinhos e o terceiro é uma evolução sobre arvores de decisão.

2.3.1.1.1 SVM

O funcionamento de SVM é baseado na transformação dos dados de entrada para um sistema multidimensional chamado de espaço vetorial de características através de uma função *kernel*. Neste novo espaço vetorial, o algoritmo busca traçar um hiperplano de modo que seja possível maximizar a distância entre ele e retas paralelas, também chamadas de vetores de suporte, tangenciais ao ponto mais próximo de cada classe distinta. Nota-se que, apesar deste ser um algoritmo com barreira de decisão originalmente linear, devido a transformação dos valores de entrada para outro espaço vetorial, este algoritmo também torna-se capaz de lidar com problemas não lineares (Hearst et al., 1998).

2.3.1.1.2 KNN

Enquanto que para o KNN, o grupo pertencente de um novo dado obtido corresponde aos K vizinhos mais próximos do próprio ponto. Ou seja, para associar os vizinhos, a semelhança entre os pontos é estimada pela métrica de Minkowski, que pode ser igual, por exemplo, a distância de Manhattan, Euclidiana ou de Chebyshev, conforme o valor de p na equação. Neste algoritmo, ao selecionar um valor muito pequeno para K resulta na formação de pequenas vizinhanças espalhadas, enquanto que a seleção de um valor muito alto fará com que grupos menores sejam ignorados (Kramer; Kramer, 2013).

2.3.1.1.3 GBDT

Finalmente, algoritmos do tipo GBDT podem ser entendidos em duas partes, inicialmente, o processo de aprendizado para máquinas de *gradient boosting* expande o modelo consecutivamente para melhorar a acurácia de predição da variável alvo. Este conceito é uma extensão de métodos de *ensemble*, onde novos estimadores são adicionados sequencialmente e aprendem com base no erro de todo o conjunto (Natekin; Knoll, 2013). Especificamente para o caso de GBDT, estes novos estimadores são árvores de decisão, algoritmos que agrupam, sucessivamente, em regiões distintas os dados com base em valores dos campos de entrada, de modo que é possível traçar similaridades entre elementos agrupados dentro de um nó em específico ou de alguma sub-árvore dentre as decisões do modelo (Ville, 2013).

2.3.1.2 Aprendizado profundo

Esta técnica de aprendizado de máquina fornece um poderoso arcabouço para resolução de problemas complexos. A estrutura dos algoritmos fornece um controle modular, ao permitir a adição de camadas com diferentes características para compor o modelo. O exemplo mais característico para estes modelos são redes *feedforward*, que possuem como objetivo aproximar uma função f , com mapeamento $y = f(x, \theta)$ onde a rede aprende o valor para o parâmetro θ

que resulte na melhor aproximação para o resultado, levando o nome de *feedforward* porque as informações fluem a partir da função f através das computações intermediárias até chegar no resultado y pelas diferentes camadas (Goodfellow; Bengio; Courville, 2016).

Cada camada é composta por unidades, chamadas de neurônios, que recebem um vetor como entrada e calculam um valor escalar como saída. Tradicionalmente, cada neurônio computa a soma ponderada de todas as entradas dos neurônios anteriores e então aplicar uma função de ativação não linear para gerar seu resultado. Vale ressaltar que a função de ativação não ser linear é um fator crucial para que a rede neural possa representar qualquer outra função arbitrária. Caso a função de ativação fosse linear, qualquer combinação de redes e neurônios poderia ser representada por um *perceptron* de uma camada, pois a soma ponderada das funções resultaria em outra função linear (Russell; Norvig, 2016).

2.3.1.2.1 Convolucionais

A fim de aprimorar a detecção de padrões nos dados de entradas redes *convolutional neural network* (CNN) foram desenvolvidas. Geralmente aplicadas com o uso de imagens, estas redes neurais aplicam um processo de convolução nos dados de entrada, mantendo um relacionamento de localidade das informações e reduzindo a quantidade de parâmetros que precisam ser aprendidos. Por exemplo, para dados que possuam uma relação espacial ou temporal entre si, o simples uso de uma rede neural *feedforward* tradicional criaria um peso para cada ponto de dados, o que representa um crescimento exponencial no consumo de memória. Ademais, uma rede tradicional apresentaria o mesmo resultado para qualquer permutação entre os dados de entrada, pois, diferentemente das redes convolucionais, elas não mantêm um vínculo entre a localidade dos dados de entrada (Russell; Norvig, 2016).

O funcionamento básico destas redes ocorre pela troca de um peso escalar único no neurônio por um *kernel* de algum tamanho, seja d , que possuirá d^n pesos, onde n é a quantidade de dimensões do *kernel*. Deste modo, o resultado de um neurônio será a convolução do *kernel* aplicado no vetor de entrada e então o resultado da função de ativação, produzirá o que é chamado de *feature map*. Ademais, uma rede CNN também costuma apresentar camadas de *pooling*, que funcionam de forma semelhante à uma convolução no sentido de que, para localidades específicas do vetor de dados, estas camadas elegem algum valor para compor o valor de saída, seja por um cálculo de valor médio ou máximo, por exemplo. Estas camadas de *pooling* servem para diminuir a dimensionalidade dos dados e não possuem parâmetros para serem aprendidos associados, nem aplicam funções de ativação. Por final, é feita uma operação para achatar as dimensões dos dados, a partir deste momento, são aplicadas camadas densas, conforme uma rede neural tradicional para classificar o resultado final (Russell; Norvig, 2016).

Uma *fully convolutional network* (FCN), é bastante semelhante de uma rede CNN, também apresentando camadas de convolução, com *pooling*, mas estas não fazem a parte final de achatar os *feature maps*. A operação que substitui o processo de achatar é uma camada de *pooling*

global, ou seja, o tamanho da janela de *pooling* é todo o *feature map*. Após isto, a rede FCN refaz o processo inverso de convolução, que aumentará o tamanho do *feature map* até voltar para a dimensão original de entrada. Portanto, enquanto uma CNN é capaz de exclusivamente identificar uma classe para o dado de entrada, FCN, quando aplicadas com o processo de deconvolução, podem ser utilizadas para fazer uma segmentação semântica do valor de entrada, destacando as classes encontradas por região (Long; Shelhamer; Darrell, 2015).

2.3.1.2.2 Recorrente

Há um outro modelo de arquitetura para aprendizado profundo, o qual difere ligeiramente de uma rede *feedforward*, este modelo é chamado de *Recurrent Neural Network* (RNN), e o que a torna especial é a presença de ciclos entre as camadas do modelo. A presença destes ciclos permite com que o resultado produzido por um neurônio seja reutilizado em conjunto como a entrada do mesmo, criando uma espécie de memória, onde dados passados influenciam no aprendizado para valores futuros. No entanto, estes modelos tendem a sofrer de problemas de explosão ou desvanecimento de gradiente, ao calcular o gradiente da função de perda para determinados pesos (Russell; Norvig, 2016).

Uma forma de combater o problema no calculo de gradiente foi proposto através de redes *long-Short term memory* (LSTM), que, diferentemente de RNN, onde o acumulo de memória ocorre de modo composto para cada informação, para LSTM novas informações são adicionadas aos valores presentes através de um componente chamado de *memory cell*. Neste componente, há três operações que controlam informações que devem permanecer na rede ou que devem ser esquecidas (Russell; Norvig, 2016).

2.3.1.2.3 Residual

O último tipo de rede neural abordado neste trabalho é o Residual Network (ResNet). Neste modelo, diferentemente das redes *feedforward* normais, onde cada camada substitui a representação dos dados anteriores completamente, estas redes seguem a ideia de que uma camada deve "*perturbar*" esta representação, ao invés de substitui-la. O funcionamento para este modelo, em termos simples, soma a entrada de um neurônio com o valor computado na sua saída como resultado para o neurônio seguinte, ou formulando matematicamente, dado o valor residual calculado por um neurônio i com base na entrada z^i sendo $f(z^i) = W^{i+1}z^i + b^{i+1}$, então a representação para o próximo neurônio será $z^{i+1} = g_{i+1}(z^i + f(z^i))$. Este processo de transmitir a entrada de uma camada para a seguinte é chamado de atalho, ou *skip connection*. Por fim, vale ressaltar que esta estrutura faz com que, por padrão, a rede transmita informação, o que pode ser visto ao considerar os vetores de parâmetros e viés como nulos, e a função de ativação como Rectified Linear Unit (ReLU), o resultado deste modelo para qualquer número de camadas sera $y = ReLU(x)$ para x e y sendo entrada e saída, respectivamente (Russell; Norvig, 2016).

Tabela 2 – Características por domínio

| Domínio | Característica |
|------------------|---|
| Tempo | Máximo, mínimo, média, mediana, moda, desvio padrão, Hjorth (atividade, mobilidade, complexidade), entropia, <i>High-Order Crossing</i> (HOC), comprimento de onda (WL) |
| Frequência | Análise de faixas PSD |
| Tempo-frequência | Eficiência de energia recursiva, energia de transformadas <i>wavelet</i> discretas, espectrograma, espectro de Hilbert-Huang |

Fonte: Elaborado pelo autor.

2.3.2 Engenharia de características

No processo de aprendizagem, é comum que o conjunto de dados passe por diversos processos prévios ao treinamento do modelo que visam melhorar a qualidade dos dados, criar dados para simular cenários específicos, mapear diversas características em outros valores que possam trazer mais informações as quais não eram diretamente disponíveis e selecionar quais as características mais relevantes, filtrando valores redundantes que possam atrasar o processo de aprendizagem. Estes processos são organizados em etapas, chamadas de pré-processamento, extração de características e seleção de características, respectivamente. Ainda, conforme a origem dos dados e a tarefa a ser executada, algumas técnicas utilizadas podem variar. Sendo assim, este trabalho concentra-se apenas nos processos utilizados para tratamento de sinais psicofisiológicos e reconhecimento de estados mentais.

Inicialmente, sobre pré-processamento, por tratar de sinais coletados diretamente por sensores, é necessário remover artefatos e interferência que não são relacionados à tarefa de estudo, no caso, esforço cognitivo. A origem destes artefatos é diversa, como por exemplo movimentos do voluntário durante a coleta, variação do contato do sensor com a pele ou campos magnéticos externos, como do próprio dispositivo de coleta. Dentre as técnicas utilizadas na remoção de artefatos estão subamostragem, filtros passa-faixa, transformadas *wavelet*, normalização, winsorização e análise de componente independente (ICA) (Saganowski, 2022).

Na sequência é realizado o processo de extração das características, o qual é responsável pela redução da dimensão dos dados enquanto mantém informações importantes. Esta etapa permite maior eficiência para o treinamento do modelo, uma vez que extrai as propriedades relevantes de um segmento selecionado ou de janelas deslizantes sobre o sinal coletado. Ademais, vale pontuar que estas características são organizadas, principalmente, por domínio de tempo, frequência ou tempo-frequência, como demonstrado na Tabela 2. Dentre estes domínios, diversos métodos são aplicados para extração de características como a morfologia do sinal, parâmetros de Hjorth, espectro de densidade de energia (PSD), transformadas *wavelet*, decomposição de onda, média, mediana, curtose, entropia, planos de Poincaré, entre outros (Islam et al., 2021).

Adiante, após formar o conjunto das características relevantes dos dados, estas passaram por um processo de seleção, o qual busca diminuir o número de parâmetros que serão utilizados, removendo características redundantes ou pouco significativas. Como testar todas as possibilidades de combinação para cada subconjunto de características é inviável, em questão de tempo e custo de computação, os métodos aplicados não produzem o resultado ideal, de forma que possa ter variação do resultado entre diferentes métodos para seleção (Saganowski et al., 2023).

No geral, existem quatro categorias que englobam métodos de seleção de características, sendo eles, filtro, transformação, envelope, e embutido. Filtro depende em alguma medida, como correlação ou ganho de informação em conjunto com um limite inferior, para seleção. A transformação produz uma projeção ortogonal das características, de forma a reduzir a variância. Envelope combina a seleção com classificação e avalia a performance de um subconjunto ao comparar as métricas da classificação com a de outro subconjunto de características. Finalmente, no embutido um algoritmo de aprendizado profundo gera automaticamente as características, com processos que removem as piores e adicionam melhores resultados (Saganowski et al., 2023).

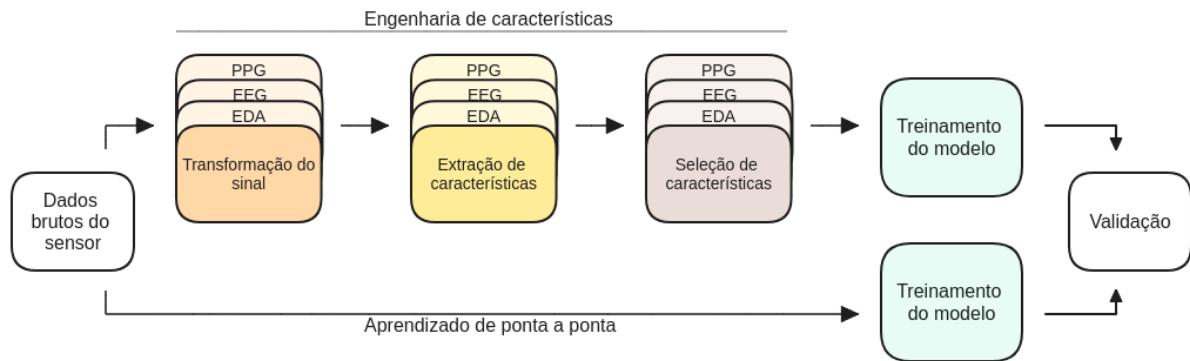
No entanto, informações sobre métodos de seleção de características não constam com frequência em trabalhos de reconhecimento de estados afetivos por sinais psicofisiológicos, uma vez que alguns algoritmos de aprendizado já possuem algum processo similar. Mesmo assim, alguns dos algoritmos observados são análise de componente principal (PCA), filtro por ganho de informação e embutido (Saganowski et al., 2023).

2.3.3 Aprendizado de ponta a ponta

Outra forma para desenvolver um modelo de aprendizado de máquina conta com o uso dos dados brutos coletados pelos sensores aplicados diretamente em algum modelo de classificação por aprendizado profundo. Nesse cenário, o objetivo é de que o próprio modelo seja capaz de aprender padrões ocultos nos dados da maneira que foram coletados, de modo que seja o suficiente para a classificação correta do estado mental.

Este método apresenta uma vantagem clara, uma vez que simplifica drasticamente o processo entre a coleta de dados e obter a inferência sobre o estado emocional. Assim, todas as etapas de engenharia de característica, como pré-processamento, extração e seleção passam a ser desnecessárias, o que por sua vez gera um consumo maior em computação mas reduz o requisito de conhecimento especializado sobre processamento de sinais e psicofisiologia do sinal. Outrossim, diferentes sensores possuem taxa de coleta diferente, logo não estão sincronizados no tempo, com aprendizado de ponta a ponta torna-se possível elaborar camadas específicas por sensor e simplificar a junção dos diferentes dados. Ademais, em comparação com métodos tradicionais, utilizando engenharia de características, o aprendizado de ponta a ponta obteve performance similar ou melhor (Schmidt et al., 2019).

Dentre as arquiteturas utilizadas estão perceptron multicamada (MLP), CNN, *convolutional neural network long-short term memory* (CNN-LSTM), *single task convolutional neural network* (ST-CNN), *multi task convolutional neural network* (MT-CNN), FCN, ResNet, Enco-

Figura 1 – Comparação entre aprendizado de ponta a ponta com engenharia de características

Fonte: Elaborado pelo autor.

der, *multi channel deep convolutional neural network* (MDCNN), *time convolutional neural network* (Time-CNN) e *multilayer perceptron long short term memory* (MLP-LSTM). Contudo, mais uma vez, a comparação entre trabalhos não é possível, uma vez que os conjuntos de dados são completamente distintos. Não obstante, em todos artigos, novamente, o aprendizado de ponta a ponta obteve performance igual ou superior quando comparado com o processo tradicional, através da engenharia de características (Dzieżyc et al., 2020; Kanjo; Younis; Ang, 2019).

2.3.4 Validação

Durante o treinamento, espera-se que o modelo seja capaz de atingir um nível adequado de compreensão dos dados, com capacidade para inferir em informações novas em uma taxa de erro aceitável, para isso, busca-se aumentar a quantidade de dados disponíveis para o treinamento. No entanto, caso simplesmente o modelo seja apresentado à mais dados, pode acontecer algo chamado *overfitting*, um cenário em que o modelo não aprendeu uma generalização do comportamento para os dados, mas sim exatamente como o conjunto de dados de treinamento está estruturado. Para lidar com este problema, técnicas de validação são utilizadas (Berrar, 2019).

Dentre as diversas técnicas para aplicar a validação do modelo, todas tem em comum o comportamento de dividir a quantidade total dos dados disponíveis em conjuntos distintos, os quais também são usados para funções diferentes. Ao usar *one-time split*, o total de dados disponíveis vai ser dividido uma única vez em dois conjuntos, um para treino do modelo e outro para testes, onde o modelo deve tentar classificar os valores e ter sua performance medida. Nesta técnica, a quantidade de dados dedicada para treino costuma variar entre 90 e 70%, sendo o resto usado para treino (Berrar, 2019).

Outra técnica é *k-fold*, na qual primeiramente é estabelecido um valor de k , que será responsável por determinar o número de rodadas, também chamadas de *folds*, em que um modelo será treinado e testado para conjuntos novos de treino e teste. Em cada uma destas rodadas, o total de dados será dividido entre os dois conjuntos conforme uma proporção definida por $1/k$,

ou seja, caso o valor de k seja 10, o conjunto de treino e testes serão compostos por 90 e 10% dos dados totais, respectivamente. Vale ressaltar que, enquanto um par de conjuntos não tem sobreposição, dois conjuntos de treino ou teste entre rodadas diferentes podem ser sobrepostos. Ademais, uma evolução a partir do k -fold é um método chamado de k -fold cross validation, onde k rodadas são estabelecidas da mesma forma, no entanto, para cada fold sendo executado, o conjunto de testes é distinto dos demais, isto é, sem sobreposição. Esta é uma das formas mais utilizadas para validação de modelos (Berrar, 2019).

Dentre as técnicas citadas, diversas variações de cada método foram elaboradas, principalmente do k -fold cross validation. Estas variações costumam focar em otimizar algum aspecto específico, geralmente dependente da distribuição dos dados e problema sendo resolvido (Berrar, 2019). Uma vez que tem-se como objetivo generalizar o modelo para o maior número possível de pessoas, a técnica de *leave-one-subject-out cross validation* (LOSOCV) é estabelecida, na qual, para um conjunto de dados composto por informações obtidas por diversas pessoas distintas e que estas informações variam de forma intrínseca ao indivíduo, como por exemplo é o caso de dados psicofisiológicos, cada fold terá como conjunto de testes os dados relacionados a uma pessoa em particular. Este método tende a fornecer melhores resultados para generalização de comportamento ao classificar indivíduos novos, pois garante que o modelo será testado em cada rodada com dados de pessoas desconhecidas, sendo uma melhor representação do cenário real de uso (Saganowski et al., 2023).

2.4 Trabalhos relacionados

Shu et al. (2020) elaboraram um mecanismo para reconhecimento de emoções a partir de dados da frequência cardíaca coletados a partir de um sensor PPG de bracelete. Neste trabalho, 25 voluntários foram submetidos a um conjunto de vídeos curtos representando alguma emoção entre neutro, feliz ou triste. A partir dos dados coletados, um processo de normalização foi aplicado e o processo de engenharia de características feito com os dados normalizados e brutos em conjunto. Para classificação, algoritmos tradicionais foram utilizados, com validação LOO. Os algoritmos baseados em árvores tiveram um desempenho maior.

Borisov, Kasneci e Kasneci (2021) desenvolveram um sistema para detecção de esforço cognitivo a partir de bracelete, ademais, a solução proposta foi a vencedora da competição *CogLoad@UbiComp* 2020. Para o conjunto de dados, 23 voluntários resolveram tarefas com diferentes níveis de complexidade, sendo os dados classificados de forma binária, como esforço cognitivo, ao realizar a tarefa, e ausência de esforço em períodos de descanso. Os dados passaram por um processo de engenharia de características, restrito a estatísticas básicas no domínio do tempo devido ao tamanho limitado do número de amostras coletadas. Para a classificação, algoritmos tradicionais foram utilizados além de MLP, quanto para validação, uma variação de k -fold. Os melhores resultados obtidos foram com métodos *Ensemble*, o qual no caso foi composto por um conjunto de GBDT.

Ding et al. (2020) utilizou métodos multimodais para detectar esforço cognitivo. Neste projeto, 18 voluntários foram submetidos a séries de tarefas aritméticas mentais enquanto tinham aparelhos de alta precisão medindo sinais de EDA, eletromiografia (EMG) além de ecocardiograma ECG, respondendo questionários para indicar o esforço durante a atividade. Os sinais foram submetidos a diversas técnicas de pré-processamento, como filtros passa-faixa, transformadas *wavelet* e normalização, mas nenhuma outra etapa de engenharia de características foi mencionada. Diversas rodadas de classificação foram realizadas, com permutações de sinais coletados e diversos algoritmos distintos com validação *k-fold* para todos os casos.

2.4.1 Lacuna de pesquisa

No trabalho de Shu et al. (2020), apesar do sucesso no reconhecimento de estados emocionais, a exata definição de emoção utilizada na pesquisa não foi estabelecida, e uma vez que esforço cognitivo lida especificamente com um tipo de memória operacional, com recursos limitados, os conceitos não podem ser livremente intercambiáveis. Neste cenário, apesar de uma possível interação e sobreposição dos dois construtos (Scheffel; Gärtner, 2024), a busca do reconhecimento de esforço cognitivo segue como uma demanda, na qual os métodos utilizados por Shu et al. (2020) podem fornecer auxílio no tratamento de indicadores psicofisiológicos.

Para Borisov, Kasneci e Kasneci (2021), enquanto a pesquisa foi direcionada diretamente à construção de esforço cognitivo com bons resultados, o conjunto de dados utilizados possui tamanho limitado, conforme também constatado pelos autores, de 1.4MB¹, dividido em quatro sensores diferentes, o que limita as técnicas que podem ser utilizadas para engenharia de características e modelos de aprendizado profundo. Ademais, a taxa de coleta das informações de 1Hz utilizada para compor o conjunto de dados apenas seria suficiente para sinais com largura de banda com, no mínimo, dois segundos de duração (Shannon, 1949).

Em contrapartida, Ding et al. (2020) elaborou seu próprio conjunto de dados com uma constelação de equipamentos de alta qualidade, com eletrodos presentes no trapézio, antebraço e dedos, além de equipamentos no torso para monitorar a respiração. Desta forma, o uso deste mecanismo para monitoramento cotidiano é inexecutável, uma vez que acesso ao equipamento é limitado e o uso pode obstruir movimentos.

¹ Acesso e download do conjunto de dados realizado no dia 11 de Maio, 2024, através do link <<https://www.ubittention.org/2020/#papers>>

2.5 Considerações do capítulo

Este capítulo introduziu o conceito e origem do esforço cognitivo e diferentes categorias de métodos que podem ser utilizados para sua detecção, além de boas práticas para estudo envolvendo indicadores psicofisiológicos. Adiante, foram apresentados os órgãos de origem dos sinais coletado e informações básicas sobre os sensores utilizados neste procedimento.

Por seguinte, uma breve contextualização sobre inteligência artificial foi realizada, elencando algumas de suas áreas de atuação e aprofundando o conceito de aprendizagem de máquina, para a qual foram descritas diversas etapas que compõem o processo de construir um modelo funcional de aprendizado de máquina. Vale ressaltar que todas as etapas descritas foram contextualizadas no cenário de indicadores psicofisiológicos e reconhecimento de estados mentais. Posteriormente, uma forma distinta para realizar a mesma tarefa foi apresentada, a qual simplifica diversas operações nos dados a custo de utilizar um modelo mais complexo de aprendizado profundo para classificação. Então, métodos para validar a performance do modelo são descritos, com destaque para técnica que melhor simula um cenário de uso real.

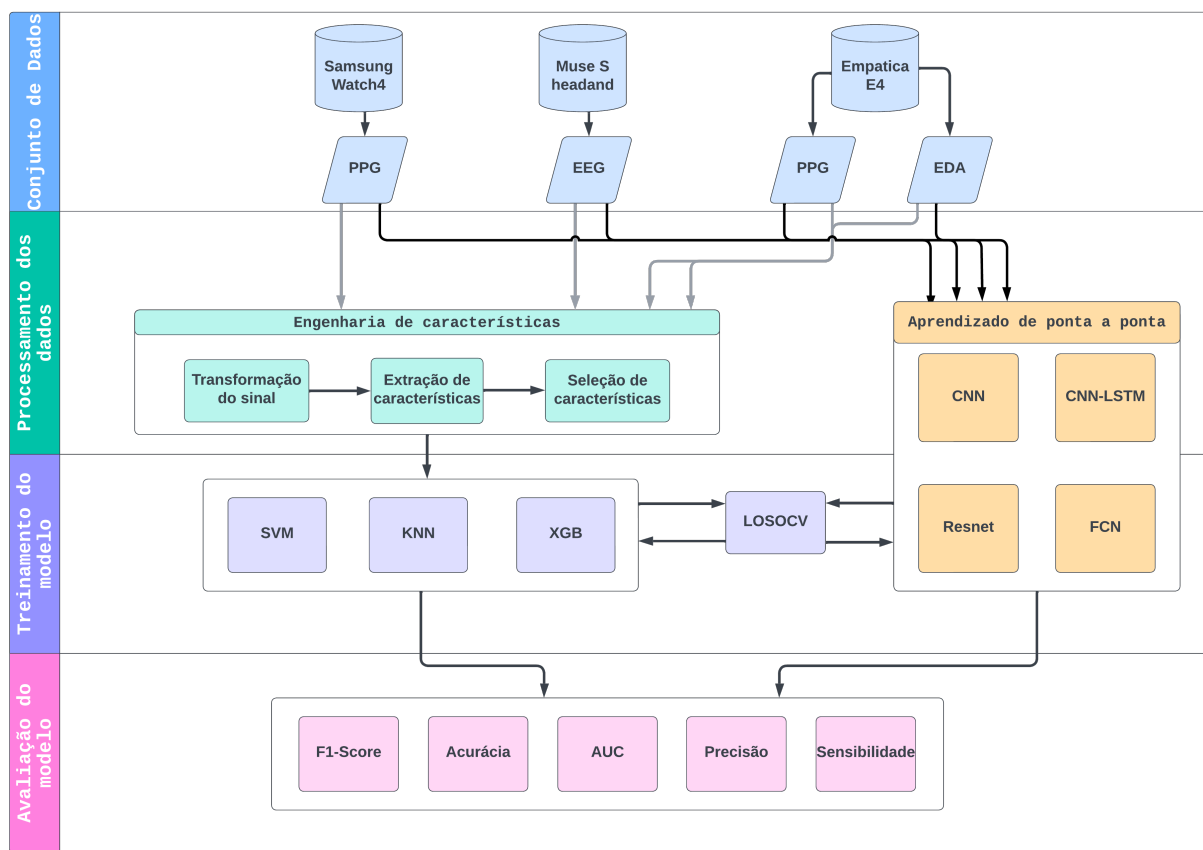
3 MATERIAIS E MÉTODOS

O processo para desenvolver um modelo de inteligência artificial capaz de detectar esforço cognitivo é composto por quatro etapas. A primeira delas é o estabelecimento do conjunto de dados a ser utilizado, detalhando os dispositivos e sensores utilizados para a coleta, além do sinal que será utilizado. A partir dos dados definidos, uma vez que o objetivo é utilizar um dispositivo vestível comum, com bastante presença no mercado, cada sinal será utilizado de forma individual, semelhante ao cenário disponível com o dispositivo da *Samsung*. Na próxima etapa, cada sinal será de-multiplexado para dois fluxos distintos, o primeiro aplicará o processo de engenharia de características, transformando o sinal para redução de ruídos e interferências, extraindo informações a partir do sinal psicofisiológico e selecionando os valores mais relevantes, enquanto o segundo fluxo utilizará aprendizado de ponta a ponta, ou seja, a partir do sinal bruto, aplicará uma rede neural para classificar o resultado final, o que engloba as etapas de processamento de dados e treinamento do modelo. O fluxo de engenharia de características, por sua vez, na etapa de treinamento do modelo, fornecerá o conjunto de dados transformado para algoritmos tradicionais para classificação.

Na etapa de treinamento do modelo, os algoritmos selecionados para o fluxo de engenharia de características foram SVM, KNN e GBDT, enquanto que para o processo de aprendizado de ponta a ponta, as redes neurais utilizadas serão CNN, LSTM, ResNet e FCN. Todos os algoritmos foram selecionados com base em trabalhos relacionados, devido sua popularidade de uso e performance obtida para a classificação de estados mentais. Ainda, ambos caminhos de treinamento passarão por uma validação LOSOCV, para que então sejam submetidos à avaliação de performance, onde serão avaliados o F_1 , *Precisão*, *Sensibilidade*, *Acurácia* e *AUC*, obtidos ao classificar os dados do conjunto de testes, o qual é composto por valores que não estavam presentes nos conjuntos de treinamento. O diagrama na Figura 2 demonstra as etapas executadas em sequência.

3.1 Conjunto de dados

O conjunto de dados *CogWear*, elaborado por Grzeszczyk et al. (2023), foi desenvolvido com o objetivo de avaliar a possibilidade de utilizar sensores de sinais fisiológicos equipados em dispositivos vestíveis de qualidade comercial para realizar a detecção de esforço cognitivo. Neste conjunto de dados, existem dois grupos de valores, um é o piloto, onde foi realizado um experimento com intuito de testar o processo de coleta de dados, utilizando todos os aparelhos disponíveis e também simulando um cenário de esforço cognitivo. Enquanto para o segundo, também foi realizado em conjunto com um questionário *gamificado*, com viés de identificar a influencia do processo de *gamificação* no estado de esforço cognitivo.

Figura 2 – Visão da estrutura elaborada

Fonte: Elaborado pelo autor.

Inicialmente, o conjunto piloto possui indicadores psicofisiológicos de 11 voluntários enquanto eles realizavam o teste de Stroop, uma tarefa cognitivamente exigente onde são apresentados nomes de cores escritas em uma fonte colorida e o objetivo é informar qual a cor que a palavra foi escrita. Este conjunto piloto não foi utilizado para treinamento ou testes dos modelos, seu uso serve exclusivamente para experimentação de processos, especialmente por ter seu tamanho bastante reduzido. Para o segundo grupo de dados, 13 voluntários participaram de um experimento com gamificação para responder questionários sobre saúde e bem estar. Neste experimento, os participantes também realizaram o teste de Stroop antes de responder aos questionários, em duas ocasiões diferentes para identificar o impacto da gamificação no esforço cognitivo. Como o objetivo do trabalho é exclusivamente a identificação de esforço cognitivo, os resultados dos questionários serão todos ignorados.

Portanto, neste conjunto três aparelhos distintos foram utilizados simultaneamente para a coleta de dados, sendo eles um dispositivo de qualidade clínica *Empatica E4 watch* que serve como controle para comparar os resultados obtidos, e dois dispositivos comerciais *Muse S EEG headband* e *Samsung Galaxy Watch4*, ambos medindo informações distintas. Cada um dos participantes usou no punho esquerdo o *Samsung Galaxy Watch4*, no punho direito o *Empatica*

E4 e na testa o *Muse S EEG headband* enquanto sentados confortavelmente em frente ao monitor de um computador, onde realizaram o teste.

Tendo em vista que o objetivo é elaborar um modelo capaz de detectar esforço cognitivo em dados comuns, os quais podem facilmente ser obtidos através de algum dispositivo popular, o foco deste trabalho será no aparelho *Galaxy Watch4*, por tratar-se do cenário mais comum de dispositivo vestível disponível. Mas todos os aparelhos serão utilizados a fim de comparação.

3.1.1 Indicadores coletados

Os sensores em cada dispositivo captaram sinais psicofisiológicos diferentes, mas que podem ser utilizados para derivar as mesmas métricas a fim de comparação ou indicadores diferentes. O *Empatica E4* mediu valores de volume de pulso sanguíneo (BVP), do qual podem ser derivados valores para variabilidade de HRV e intervalo entre batimentos (IBI). Também possui sensor para temperatura da pele (TEMP), um para EDA e por último aceleração (ACC) em três eixos, mas este último não foi disponibilizado para uso no conjunto de dados. A taxa de coleta para BVP foi de 64 Hz através de um sensor PPG, 4 Hz para o EDA e TEMP também com 4 Hz, no entanto, a informação de temperatura não será utilizada. O dispositivo *Galaxy Watch4* da Samsung coletou sinal de PPG verde, semelhante ao dispositivo da *Empatica*, a uma taxa de 25 Hz. Finalmente, o sinal coletado pelo *Muse S EEG headband* foi um EEG através de eletrodos localizados nas regiões pré-frontal e temporal, com uma taxa de coleta de 256 Hz e possuem contato seco com a pele. Ademais, este aparelho também possui uma informação de giroscópio, que poderia ser utilizada para um processo avançado de remoção de artefatos de movimento, mas não caberá no escopo deste trabalho.

3.2 Pré-processamento

Nesta etapa, os dados passarão por uma análise exploratória, que visa buscar um melhor entendimento das informações a fim de tomar decisões mais assertivas sobre as transformações que serão imposta aos dados. De forma análoga ao descrito por Tukey et al. (1977), o próprio conjunto de dados será utilizado para responder perguntas relevantes no processo de identificação de quais características contribuem para observação de esforço cognitivo com auxílio de gráficos para identificar padrões. Ademais, análises quantitativas também podem ser empregadas em momentos que os recursos visuais não forneçam uma tendência clara (Buja et al., 2009).

O pré-processamento pode ser dividido em três fases, a primeira delas busca reduzir a presença de artefatos causados por movimento ou interferências e ruídos nos sinais de sensores que possam ter sido causados por outros dispositivos ou o próprio mecanismo interno do aparelho. Para este objetivo, técnicas como filtros passa-faixa e transformadas *wavelet* podem ser utilizadas, uma vez que tanto trabalhos relacionados costumam utilizá-las, quanto por estas demonstrarem eficácia comprovada no tratamento de sinais (Joseph et al., 2014).

A partir do sinal limpo, com quantidade de ruídos reduzida, o processo de extração de características, o qual buscará inferir o maior número possível de informações a partir do sinal psicofisiológico coletado. Entre as características buscadas, estão aquelas relatadas na Tabela 2, quanto valores psicofisiológicos como HRV, IBI, HR e BVP.

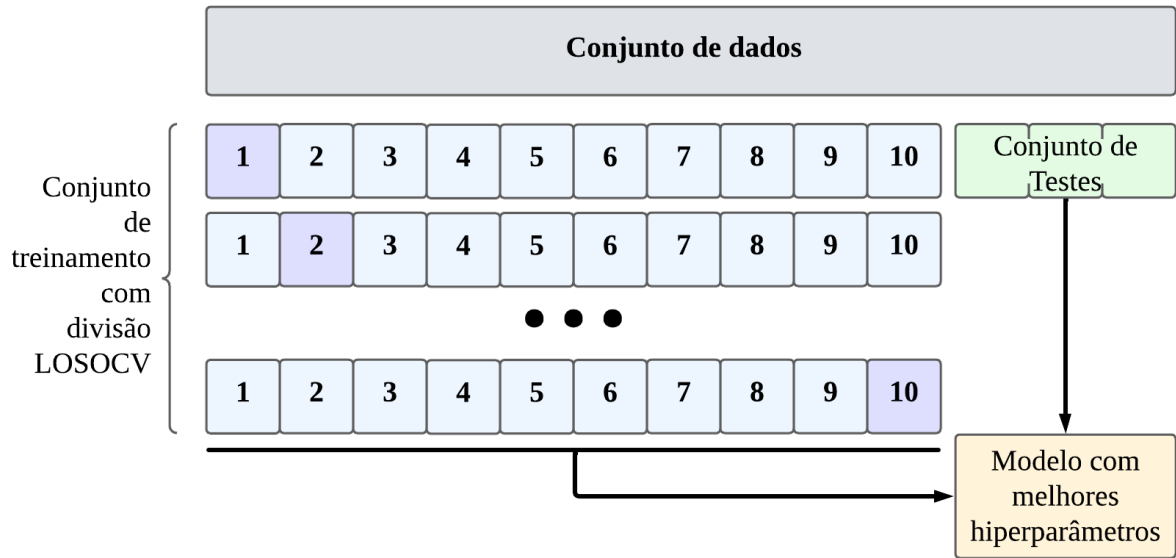
Para este trabalho, o processo de filtro e extração de características será realizado através de bibliotecas de código dedicadas ao processamento de sinais psicofisiológicos. A vantagem do uso destas ferramentas é a praticidade de aplicar várias destas técnicas de processamento de sinal de forma simples, através de um código já testado e validado. Em contrapartida, estas bibliotecas implementam um número limitado de funções, as quais não necessariamente cobrem todas as informações passíveis de serem obtidas. Devido à restrições de escopo do trabalho, outros métodos que possam complementar o processo realizado por estas ferramentas não serão profundamente explorados. Ademais, o foco para obter mais características do sinal recai sobre os sensores de PPG, novamente, por tratar-se do sensor mais popular do grupo.

Com o conjunto de características disponível, técnicas para selecionar as características mais relevantes e independentes entre si serão utilizadas. A redução do número de características permitirá maior agilidade no treinamento de diferentes modelos, minimizando a perda de informações, uma vez que apenas características com maior grau de redundância ou baixa correlação com esforço cognitivo serão removidas. Este processo de seleção de características apenas será realizado através de métodos embutidos nos algoritmos de classificação, caso aplicável.

3.3 Treinamento

O treinamento dos modelos será realizado a partir dos dados de cada sensor individualmente para cada um dos modelos e ambas técnicas de treinamento. Ou seja, tanto pelo processo de engenharia de características quanto por aprendizado de ponta a ponta, cada um dos quatro sensores serão utilizados para cada um dos três algoritmos tradicionais e para as quatro redes neurais. Para este processo, o conjunto de dados será dividido em dois, um dedicado ao treinamento do modelo, composto por dados de 10 voluntários, escolhidos aleatoriamente, e o conjunto de testes, composto pelos 3 voluntários restantes. O conjunto de testes será utilizado exclusivamente para avaliar a performance de cada modelo no final de todas as etapas de treinamento e validação, portanto serão dados considerados novos, aos quais o modelo não teve acesso prévio, simulando um cenário mais próximo de uso real.

Enquanto que, para o conjunto de treinamento uma técnica de LOSOCV será aplicada a fim de buscar o modelo com maior capacidade de generalização para indivíduos novos. Esta técnica consiste em dividir os dados de treinos dez vezes, sendo que em cada uma delas um voluntário será utilizado para validação do modelo que será treinado nos restantes nove voluntários. Este processo é realizado dez vezes, ou seja, o modelo foi treinado e validado com dez conjuntos distintos de dados, com o objetivo de utilizar os parâmetros internos do modelo que forneçam o melhor desempenho ao modelo final.

Figura 3 – Divisão do conjunto de dados

Fonte: Elaborado pelo autor.

Ademais, para cada conjunto de modelo e sensor será realizada a busca de hiperparâmetros, que são variáveis de cada algoritmo prévias ao processo de aprendizado e que podem influenciar a performance do modelo. Para este cenário, será utilizado um algoritmo *tree-structured Parzen Estimator* (TPE) de busca Bayesiana, o qual é um algoritmo estabelecido com performance e efetividade comprovada (Wu et al., 2019).

Finalmente, o processo completo de treinamento contará com sete algoritmos diferentes, três direcionados ao processo de aprendizado tradicional, através de engenharia de características e outros quatro serão utilizados no processo de aprendizado de ponta a ponta, portanto também sendo responsáveis por tratar ruídos de cada sinal e compreender as características de cada um. Uma vez que cada um dos quatro sensores será utilizado em cada algoritmo, existirá um total de 28 modelos distintos treinados.

3.3.1 Espaço de busca de hiperparâmetros

Para a etapa inicial de otimização dos hiperparâmetros, primeiramente é necessário definir quais são os intervalos de valores e para quais informações ocorrerá esta busca de melhores valores. Diferente algoritmos podem possuir parâmetros diferentes, especialmente os dedicados ao aprendizado tradicional, enquanto que para as redes neurais, os valores utilizados foram, majoritariamente, os mesmos. Todos os espaços utilizados estão listados a seguir.

Exemplo de código 1 – Espaço de busca para SVM

```

1     hyperparameter["svm"] = {
2         'C': hp.choice('C', [0.1, 0.01, 0.001, 0.0001, 1, 10, 100, 1000]),
3         'kernel': hp.choice('kernel', ['linear', 'poly', 'rbf', 'sigmoid']),
4         'degree': hp.randint('degree', 1, 5),
5         'gamma': hp.choice('gamma', ['scale', 'auto']),
6         'class_weight': hp.choice('class_weight', ['balanced', None])
7     }

```

Exemplo de código 2 – Espaço de busca para KNN

```

1     hyperparameter["knn"] = {
2         'n_neighbors': hp.randint('n_neighbors', 1, 10),
3         'weights': hp.choice('weights', ['uniform', 'distance']),
4         'algorithm': hp.choice('algorithm', ['ball_tree', 'kd_tree', 'brute']),
5         'leaf_size': hp.randint('leaf_size', 2, 100),
6         'p': hp.choice('p', [1, 2])
7     }

```

Exemplo de código 3 – Espaço de busca para GBDT

```

1     hyperparameter["xgb"] = {
2         'max_depth': hp.choice("max_depth", np.arange(1,20,1,dtype=int)),
3         'eta'       : hp.uniform("eta", 0, 1),
4         'gamma'      : hp.uniform("gamma", 0, 100),
5         'reg_alpha': hp.uniform("reg_alpha", 1e-8, 10),
6         'reg_lambda' : hp.uniform("reg_lambda", 0,1),
7         'colsample_bytree': hp.uniform("colsample_bytree", 0.5,1),
8         'colsample_bynode': hp.uniform("colsample_bynode", 0.5,1),
9         'colsample_bylevel': hp.uniform("colsample_bylevel", 0.5,1),
10        'n_estimators': hp.choice("n_estimators",
11        ⇨ np.arange(10,1000,10,dtype='int')),
12        'min_child_weight' : hp.choice("min_child_weight",
13        ⇨ np.arange(1,10,1,dtype='int')),
14        'max_delta_step' : hp.choice("max_delta_step",
15        ⇨ np.arange(1,10,1,dtype='int')),
16        'subsample' : hp.uniform("subsample",0.1,1),
17        'objective' : hp.choice('objective', ['binary:logistic',
18        ⇨ 'binary:hinge']) ,
19        'eval_metric' : hp.choice('eval_metric', ['aucpr', 'logloss',
20        ⇨ 'rmse']),
21        'seed': 42
22    }

```

Exemplo de código 4 – Espaço de busca para CNN

```

1  hyperparameters['cnn'] = {
2      'lr_power': hp.randint("lr_power", 3, 7),
3      'decay': hp.choice("decay", [.01, .001, .0001, .00001]),
4      'reduce_lr_factor': hp.choice("reduce_lr_factor", [0.8, 0.5, 0.2,
5      ↪ 0.1]),
6      'batch_size': hp.choice("batch_size", [2, 4, 8, 16, 32]),
7      'baseline_weight': hp.randint("baseline_weight", 2, 3),
8      'filters_multiplier': hp.choice("filters_multiplier", [0.5, 1, 2]),
9      'kernel_size_multiplier': hp.choice("kernel_size_multiplier", [0.5,
10     ↪ 1, 2])
11 }

```

Exemplo de código 5 – Espaço de busca para FCN

```

1  hyperparameters["fcn"] = {
2      'lr_power': hp.randint("lr_power", 3, 7),
3      'decay': hp.choice("decay", [.01, .001, .0001, .00001]),
4      'reduce_lr_factor': hp.choice("reduce_lr_factor", [0.8, 0.5, 0.2,
5      ↪ 0.1]),
6      'batch_size': hp.choice("batch_size", [2, 4, 8, 16, 32]),
7      'baseline_weight': hp.randint("baseline_weight", 2, 3),
8      'filters_multiplier': hp.choice("filters_multiplier", [0.5, 1, 2]),
9      'kernel_size_multiplier': hp.choice("kernel_size_multiplier", [0.5,
10     ↪ 1, 2])
11 }

```

Exemplo de código 6 – Espaço de busca para LSTM

```

1  hyperparameters["lstm"] = {
2      'lr_power': hp.randint("lr_power", 3, 7),
3      'decay': hp.choice("decay", [.01, .001, .0001, .00001]),
4      'reduce_lr_factor': hp.choice("reduce_lr_factor", [0.8, 0.5, 0.2,
5      ↪ 0.1]),
6      'batch_size': hp.choice("batch_size", [2, 4, 8, 16, 32]),
7      'baseline_weight': hp.randint("baseline_weight", 2, 3),
8      'filters_multiplier': hp.choice("filters_multiplier", [0.5, 1, 2]),
9      'kernel_size_multiplier': hp.choice("kernel_size_multiplier", [0.5,
10     ↪ 1, 2]),
11     'lstm_units': hp.choice("lstm_units", [1, 2, 3])
12 }

```

Exemplo de código 7 – Espaço de busca para ResNet

```

1  hyperparameters["resnet"] = {
2      'lr_power': hp.randint("lr_power", 3, 7),
3      'decay': hp.choice("decay", [.01, .001, .0001, .00001]),
4      'reduce_lr_factor': hp.choice("reduce_lr_factor", [0.8, 0.5, 0.2,
5          ↪ 0.1]),
6      'batch_size': hp.choice("batch_size", [2, 4, 8, 16, 32]),
7      'baseline_weight': hp.randint("baseline_weight", 2, 3),
8      'filters': hp.choice("filters", [16, 32, 64]),
9      'kernel_size_multiplier': hp.choice("kernel_size_multiplier", [1, 2,
10         ↪ 4]),
11     'depth': hp.choice("depth", [2, 3, 4])
12 }

```

3.4 Avaliação

Para calcular a performance dos modelos, cinco métricas serão utilizadas, *Acurácia* mede a porcentagem de acerto do modelo, seja para predições positivas ou negativas. *AUC* que elabora uma curva relacionada a taxa de verdadeiro positivos com a taxa de predições falso negativas. *Precisão*, que mede a taxa de acerto dentre as classificações positivas realizadas pelo modelo em relação aos valores realmente positivos. *Sensibilidade*, a qual demonstra quantos casos positivos foram corretamente classificados dentre todos os verdadeiros positivos possíveis e por ultimo F_1 que combina as duas primeiras métricas em uma média harmônica. Para a formulação destas métricas, vale-se entender o resultado da classificação do modelo, que produz uma matriz de confusão, discriminando os valores reais dos valores previstos do modelo em quatro classes, sendo verdadeiro positivo (TP), verdadeiro negativo (TN), falso positivo (FP) e falso negativo (FN). A partir destes valores, as métricas podem ser calculadas de acordo com as formulas entre 3.1 à 3.4.

$$\text{Acurácia} = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.1)$$

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (3.2)$$

$$\text{Sensibilidade} = \frac{TP}{TP + FN} \quad (3.3)$$

$$F_1 = \frac{2 \cdot \text{Precisão} \cdot \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \quad (3.4)$$

3.5 Considerações do capítulo

Este capítulo apresentou as etapas para a construção do modelo de reconhecimento de esforço cognitivo. Partindo do conjunto de dados composto por diferentes sinais psicofisiológicos coletados por diferentes dispositivos, passando por etapas de processamento de sinais, extração de características e seleção das mesmas, além do treinamo de algoritmos distintos de aprendizado de máquina. Ademais, uma outra estratégia, que visa responsabilizar arquiteturas de aprendizado profundo também com as etapas de pré-processamento de dados, foi considerada, na qual estes modelos serão treinados para identificar esforço cognitivo a partir dos dados brutos coletados, estratégia denominada de aprendizado de ponta a ponta. Finalmente, métricas de *Precisão*, *Sensibilidade* e F_1 serão utilizadas para comparar os diferentes modelos na sua capacidade de identificação de esforço cognitivo.

4 RESULTADOS E DISCUSSÃO

Este capítulo descreve a execução das atividades propostas, detalhando cada processo envolvido no desenvolvimento dos modelos para detecção de esforço cognitivo. Nesta etapa, foram empregadas tecnologias amplamente utilizadas no ramo de inteligência artificial. Todos os códigos foram escritos na linguagem *python* e serviços de computação em nuvem foram utilizados para treinamento de modelos. Ademais, as principais bibliotecas de código utilizadas foram *scikit-learn* por Pedregosa et al. (2011), *keras* por Chollet et al. (2015), *Hyperopt* por Bergstra, Yamins e Cox (2013), *scipy* por Virtanen et al. (2020), *matplotlib* por Hunter (2007), *numpy* por Harris et al. (2020), *pandas* por McKinney (2010) e The pandas development team (2020), *HeartPy* por Gent et al. (2019a) e Gent et al. (2019b) e *NeuroKit2* por Makowski et al. (2021). Todos os artefatos, modelos, métricas e códigos produzidos no decorrer desta etapa foram publicados em um repositório público no *GitHub*¹ e estão disponíveis para livre acesso.

4.1 Análise Exploratória de Dados

Primeiramente, após carregar o conjunto de dados em um ambiente local, diversos arquivos, os quais não eram diretamente relacionados à detecção de esforço cognitivo, foram removidos. Em seguida, os resultados dos sensores coletados foram reorganizados, agrupando os dados de cada sensor por voluntário, removendo a separação original, feita por dia do experimento e valor da medição, a fim de simplificar o uso dos dados nos modelos, mas mantendo a sequência temporal em que cada evento ocorreu, ou seja, uma medição inicial em repouso, seguida pela realização de atividade, monitorando a condição de esforço cognitivo. Para cada voluntário, dois dias com as mesmas etapas foram coletados.

Em sequência, para iniciar a análise dos dados, valores nulos entre os dados coletados foram removidos. Dentre os quatro sensores, apenas o sinal de EEG apresentou valores nulos, sendo em torno de 0.02% e 0.3% do total dos dados coletados. Ademais, para a maior parte dos voluntários, o tempo de medição entre os sensores foi próximo um do outro, com exceção, novamente, do sinal coletado pelo *Muse S headband*, ao qual em alguns casos, possuía uma diferença grande no tempo de coleta, por vezes metade dos outros valores. Esta diferença em tempo de coleta sugere alguma falha na etapa de coleta de dados, podendo ser falha de equipamento, uso ou coleta. Neste cenário, de acordo com Grzeszczyk et al. (2023), os dados de EEG foram checados antes de iniciar o procedimento de coleta, não obstante, não há referência às instruções do fabricante para assegurar a coleta com qualidade.

Outrossim, nota-se que há uma diferença considerável na qualidade dos valores coletados pelos dois dispositivos comerciais, *Samsung Galaxy Watch4* e *Muse S*, comparado aos valores obtidos pelo *Empatica E4*, o qual é o equipamento voltado para estudos clínicos. Neste há uma clara delineação do formato esperado pra cada um dos sinais, com pouca interferência

¹ Endereço de acesso ao repositório <<https://github.com/domateus/cogwear>>

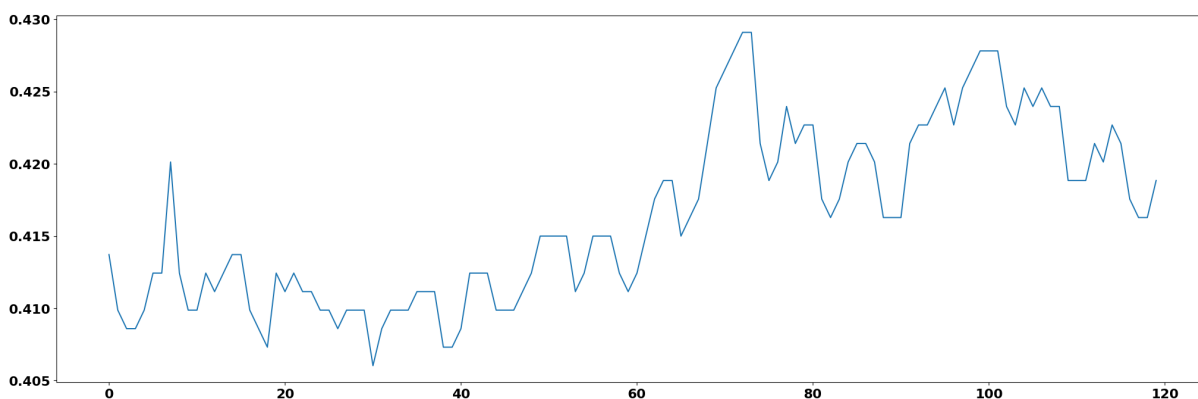
percebida, enquanto àqueles apresenta-se uma quantidade significativa de ruídos e artefatos, os quais comprometem a análise morfológica do sinal.

Finalmente, uma breve descrição dos dados é apresentada na Tabela 3, onde dados dos quatro sensores são divididos entre os voluntários do processo de coleta de dados. Nesta tabela, também são descritas algumas informações estatísticas sobre os sinais, com exceção do EEG, pois, diferentemente dos demais sinais, este é composto por 20 canais distintos, inviabilizando a exibição em formato tabular.

Em conjunto das métricas coletados sobre o conjunto de dados, a visualização dos sinais também auxilia no entendimento da qualidade do sinal obtido. A fim de analisar visualmente os dados coletados, primeiramente foram removidos quatro segundos iniciais e final para as medições, sendo o intuito de eliminar dados coletados ao vestir ou remover os equipamentos. Em seguida, cada sinal foi dividido em janelas de 30 segundos sem sobreposição, adicionando *padding* igual a 0 caso necessário para completar a janela. Sendo assim, são exibidos a segunda janela para cada um dos sensores entre as figuras 4 e 7, todas coletadas no mesmo voluntário, ou seja, representam o mesmo evento a partir de perspectivas diferentes.

Detalhando cada janela, na Figura 4 o sinal EDA do dispositivo *Empatica E4* apresenta uma série de picos esporádicos, característicos do componente fásico do sinal, o qual pode ser percebido em um intervalo pequeno de tempo. Ainda, no acompanhamento das janelas seguintes, é possível notar o movimento tônico do sinal, onde o valor base médio oscila paulatinamente.

Figura 4 – Sinal Empatica EDA



Fonte: Elaborado pelo autor.

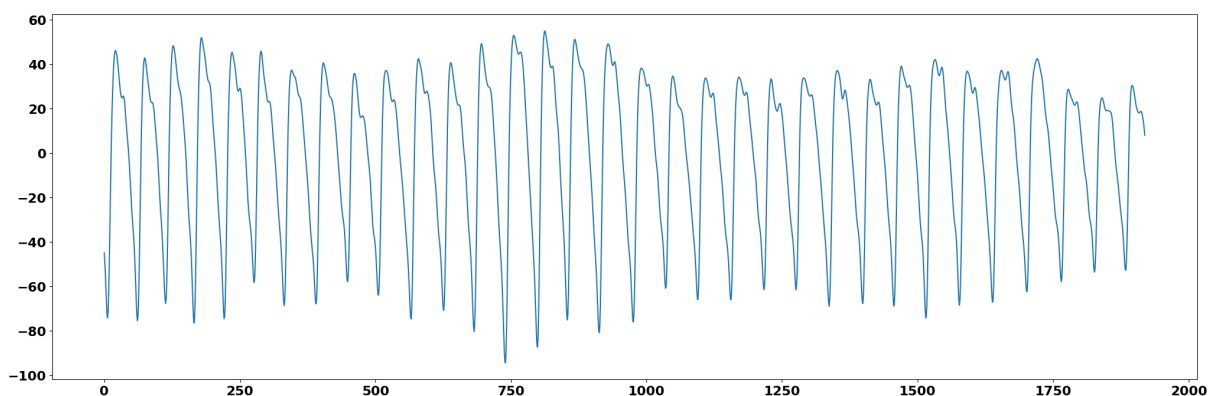
Quanto ao sinal de PPG, inicialmente a leitura feita pelo *Empatica E4*, demonstra as fases claras de pico sistólico e diastólico, além de uma periodicidade, a qual evidencia com facilidade os batimentos do coração. Estes valores possuem, consistentemente, ótima resolução e formato de onda, sendo o esperado, ao tratar-se de um dispositivos para estudos clínicos.

Tabela 3 – Resumo dos dados

| | | Tempo | Razão¹ | Amostras | Nulos | Mínimo | Máximo | Média | Desvio Padrão |
|----------------|--------------|--------------|--------------------------|-----------------|--------------|------------------|---------------|--------------|----------------------|
| Vol. 11 | Samsung PPG | 13'38" | 1.259 | 20474 | 0 | 1823074.0 | 2414563.0 | 2099593.878 | 127453.333 |
| | Empatica PPG | 13'35" | 1.26 | 52174 | 0 | -239.73 | 231.78 | 0.002 | 46.877 |
| | Empatica EDA | 13'35" | 1.26 | 3261 | 0 | 0.231 | 0.569 | 0.472 | 0.064 |
| | Muse EEG | 6'29" | 0.079 | 99730 | 72 | N/A ² | N/A | N/A | N/A |
| Vol. 12 | Samsung PPG | 16'11" | 1.69 | 24278 | 0 | 1315932.0 | 2520418.0 | 1854011.781 | 247407.574 |
| | Empatica PPG | 17'5" | 1.695 | 65631 | 0 | -873.33 | 537.78 | 0.115 | 73.405 |
| | Empatica EDA | 17'4" | 1.7 | 4099 | 0 | 0.256 | 0.847 | 0.514 | 0.149 |
| | Muse EEG | 15'20" | 1.553 | 235608 | 707 | N/A | N/A | N/A | N/A |
| Vol. 13 | Samsung PPG | 14'33" | 1.886 | 21831 | 0 | 1757389.0 | 2379034.0 | 1980817.224 | 130287.251 |
| | Empatica PPG | 15'33" | 1.483 | 59772 | 0 | -766.08 | 370.5 | 0.089 | 61.376 |
| | Empatica EDA | 15'33" | 1.48 | 3733 | 0 | 0.026 | 0.196 | 0.103 | 0.048 |
| | Muse EEG | 15'4" | 1.498 | 231675 | 333 | N/A | N/A | N/A | N/A |
| Vol. 14 | Samsung PPG | 15'29" | 1.588 | 23227 | 0 | 1811107.0 | 2454317.0 | 2066725.97 | 103696.358 |
| | Empatica PPG | 15'21" | 1.574 | 58994 | 0 | -339.89 | 653.52 | 0.011 | 38.819 |
| | Empatica EDA | 15'21" | 1.573 | 3684 | 0 | 0.018 | 6.381 | 2.652 | 2.595 |
| | Muse EEG | 7'52" | 1.637 | 120938 | 126 | N/A | N/A | N/A | N/A |
| Vol. 15 | Samsung PPG | 14'39" | 1.421 | 21979 | 0 | 1690713.0 | 2137882.0 | 1900057.031 | 104645.226 |
| | Empatica PPG | 14'45" | 1.449 | 56653 | 0 | -549.96 | 315.77 | 0.055 | 58.796 |
| | Empatica EDA | 14'44" | 1.447 | 3539 | 0 | 0.0 | 4.52 | 2.906 | 0.926 |
| | Muse EEG | 14'1" | 1.402 | 215437 | 229 | N/A | N/A | N/A | N/A |
| Vol. 16 | Samsung PPG | 14'4" | 1.35 | 21115 | 0 | 1213551.0 | 2437660.0 | 2069460.462 | 171143.459 |
| | Empatica PPG | 14'5" | 1.35 | 54093 | 0 | -620.35 | 568.64 | 0.031 | 44.369 |
| | Empatica EDA | 14'5" | 1.35 | 3380 | 0 | 0.242 | 1.031 | 0.56 | 0.25 |
| | Muse EEG | 13'43" | 1.286 | 210734 | 212 | N/A | N/A | N/A | N/A |
| Vol. 17 | Samsung PPG | 17'47" | 0.973 | 26697 | 0 | 1520790.0 | 2400669.0 | 1946413.892 | 214825.14 |
| | Empatica PPG | 14'39" | 1.439 | 56279 | 0 | -357.85 | 405.18 | 0.158 | 52.009 |
| | Empatica EDA | 14'38" | 1.437 | 3514 | 0 | 0.67 | 2.365 | 1.553 | 0.482 |
| | Muse EEG | 14'32" | 1.417 | 223426 | 311 | N/A | N/A | N/A | N/A |
| Vol. 18 | Samsung PPG | 17'5" | 1.84 | 25638 | 0 | 1450723.0 | 2319307.0 | 1867813.537 | 208300.513 |
| | Empatica PPG | 16'2" | 1.673 | 61576 | 0 | -459.75 | 455.78 | 0.094 | 56.701 |
| | Empatica EDA | 16'2" | 1.673 | 3849 | 0 | 0.395 | 2.392 | 1.445 | 0.541 |
| | Muse EEG | 15'48" | 1.629 | 242759 | 473 | N/A | N/A | N/A | N/A |
| Vol. 20 | Samsung PPG | 8'40" | 1.0 | 13014 | 0 | 1897342.0 | 2377533.0 | 2033008.325 | 83631.465 |
| | Empatica PPG | 8'40" | 1.0 | 33332 | 0 | -327.77 | 476.35 | 0.069 | 70.4 |
| | Empatica EDA | 8'41" | 1.0 | 2084 | 0 | 0.201 | 0.297 | 0.255 | 0.016 |
| | Muse EEG | 4'21" | 66998.0 ³ | 67016 | 18 | N/A | N/A | N/A | N/A |
| Vol. 21 | Samsung PPG | 7'58" | 2.354 | 11962 | 0 | 1383081.0 | 3034347.0 | 2388741.563 | 463529.49 |
| | Empatica PPG | 8'52" | 1.71 | 34084 | 0 | -418.53 | 541.37 | 0.036 | 43.941 |
| | Empatica EDA | 8'52" | 1.71 | 2130 | 0 | 0.079 | 0.197 | 0.136 | 0.032 |
| | Muse EEG | 8'40" | 1.815 | 133337 | 225 | N/A | N/A | N/A | N/A |
| Vol. 22 | Samsung PPG | 14'42" | 1.519 | 22061 | 0 | 1823911.0 | 2279930.0 | 1982624.719 | 65023.07 |
| | Empatica PPG | 15'4" | 1.472 | 57883 | 0 | -376.96 | 293.71 | -0.016 | 49.653 |
| | Empatica EDA | 15'4" | 1.471 | 3618 | 0 | 0.159 | 0.344 | 0.253 | 0.036 |
| | Muse EEG | 15'1" | 1.458 | 230681 | 357 | N/A | N/A | N/A | N/A |
| Vol. 23 | Samsung PPG | 15'55" | 1.657 | 23878 | 0 | 1515567.0 | 2209118.0 | 1911487.798 | 142497.965 |
| | Empatica PPG | 15'51" | 1.654 | 60897 | 0 | -211.96 | 186.0 | 0.013 | 50.419 |
| | Empatica EDA | 15'51" | 1.653 | 3807 | 0 | 0.115 | 0.352 | 0.203 | 0.073 |
| | Muse EEG | 15'40" | 1.619 | 240773 | 338 | N/A | N/A | N/A | N/A |
| Vol. 24 | Samsung PPG | 16'7" | 1.67 | 24199 | 0 | 1501292.0 | 2329379.0 | 1816165.537 | 152717.735 |
| | Empatica PPG | 16'16" | 1.625 | 62526 | 0 | -685.63 | 950.74 | 0.089 | 68.33 |
| | Empatica EDA | 16'17" | 1.629 | 3909 | 0 | 0.196 | 0.524 | 0.258 | 0.052 |
| | Muse EEG | 10'26" | 0.802 | 160274 | 379 | N/A | N/A | N/A | N/A |

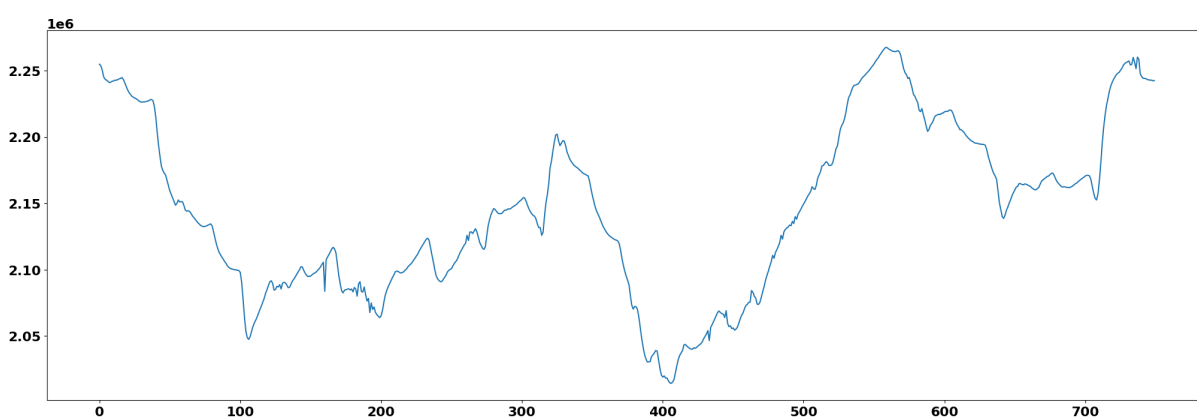
¹ Razão entre amostras de esforço cognitivo e medição base.² Há 20 sensores de EEG, portanto incapaz de representar todos aqui.³ Há apenas amostras de esforço cognitivo.

Fonte: Elaborado pelo autor.

Figura 5 – Sinal Empatica PPG

Fonte: Elaborado pelo autor.

Em comparação, por tratarem do mesmo indicador fisiológico, ao sinal da Figura 5, os valores coletados pelo dispositivo *Samsung Galaxy Watch4*, exibidos na 6, possuem uma deterioração significativa de qualidade, excluindo qualquer possibilidade de análise morfológica da onda visualmente. De forma constante, a falta de clareza neste sinal repete-se para todos os voluntários no decorrer de todo o experimento, ressaltando a diferença de qualidade entre um sensor de um dispositivo comercial para o de um clínico. Neste cenário em específico, a diferença de qualidade será um grande desafio, visto que o relógio da *Samsung* é o principal representante dos dispositivos de ampla inserção no mercado, não apenas pelo baixo custo, mas também por versatilidade de uso.

Figura 6 – Sinal Samsung PPG

Fonte: Elaborado pelo autor.

Finalmente, na Figura 7 são exibidos os quatro canais coletados para cada uma das cinco faixas de onda do EEG, coletado pelo dispositivo *Muse S*, o qual também é um equipamento voltado ao mercado consumidor, com mais baixo preço. Nesta imagem, é possível perceber

certa correlação entre os canais em alguns momentos. No entanto, diversas janelas coletadas apresentam baixa qualidade, com artefatos impedindo qualquer entendimento do sinal.

Figura 7 – Sinal Muse EEG



Fonte: Elaborado pelo autor.

4.2 Pré-processamento

Para cada um dos tipos diferentes de sinais psicofisiológicos, os dados passaram por um processo distinto de pré-processamento, sendo o principal foco dedicado ao PPG, visto que para este é possível comparar diretamente a qualidade de um dispositivo comercial de um clínico. Ademais, não apenas por questão do sensor PPG ser mais popular em dispositivos vestíveis comuns, por restrições de tempo relacionado ao prazo de entrega, o pré-processamento para EDA e EEG não foi feito de maneira igualmente detalhada.

No entanto, todos os valores passaram por um processo em comum de divisão em janelas de 30 segundos, como citado anteriormente. O motivo para agrupar os dados em uma janela de tempo são ambos de cunho prático quando teórico. Do ponto de vista prático, cada sensor faz uma medição múltiplas vezes por segundo, logo, caso um modelo fosse inferir os dados com tanta frequência, o consumo de recursos computacionais seria excessivo, além do fato de que não é relevante para uma pessoa monitorando seu esforço cognitivo ter diversos resultados dentro de um único segundo. Enquanto que do ponto de vista teórico, os modelos de aprendizado profundo utilizados não são capazes de gerar uma convolução em um único valor, enquanto que as características geradas aos algoritmos tradicionais de aprendizado de máquina, precisam de dados de uma quantidade de tempo maior para serem computadas, por exemplo, medir o HR necessita comparar o tempo entre batimentos consecutivos do coração, logo limitada a alguns segundos para obter um valor confiável.

Neste sentido, o tamanho em segundos da janela fornece um contexto sobre o estado psicofisiológico do indivíduo, portanto este valor pode influenciar a qualidade do modelo. Para Cai e Epp (2024), uma janela de 90 segundos apresentou os melhores resultados na predição de esforço cognitivo, em comparação com janelas menores e maiores. No entanto, o conjunto de dados utilizado neste projeto inviabiliza uma janela tao grande, pois há voluntários onde o tempo total de coleta é menor que nove minutos, contemplando dois intervalos de esforço cognitivo e dois intervalos no estado base. Ou seja, para dividir estes dados em janelas seria necessário descartar uma quantidade significativa de informações, algo inviável considerando o tamanho já limitado de amostras disponíveis.

Em contrapartida, Tervonen, Pettersson e Mäntyjärvi (2021) analisou o impacto de janelas bastante curtas na eficácia de modelos para detectar esforço cognitivo. Embora tenha sido detectado uma perda de qualidade para janelas muito pequenas, esta diferença não foi proporcional à quantidade de informação reduzida, ao comparar com janelas maiores. Ademais, também observou-se que para alguns indivíduos, os modelos tiveram melhor desempenho com janelas mais curtas, ressaltando a influência que diferenças individuais possuem sobre a capacidade de identificação de esforço cognitivo. Sendo assim, portanto, o tamanho de 30 segundos para a janela foi escolhido , pois, apesar de uma penalização de performance, este valor reduz a perda de informações do conjunto de dados enquanto permanece em um nível aceitável para a eficácia dos modelos de inteligência artificial.

4.2.1 EDA

O processamento específico para este sinal foi completamente realizado através da biblioteca *NeuroKit2*. Esta biblioteca recebe o sinal coletado sem nenhum processamento prévio, aplica um filtro condizente com a frequência de coleta e retorna informação sobre características do sinal no formato de uma lista com 11 colunas e mesmo comprimento do sinal bruto de entrada. Estas colunas são compostas pelo sinal bruto, sinal limpo, componente tônico, componente fásico, início da SCR, momento do pico da SCR, altura do pico de SCR, amplitude de SCR, tempo de crescimento de SCR, momento de recuperação de SCR e tempo de recuperação de SCR. Finalmente, para cada uma das colunas, valores inválidos foram substituídos por zero e então geradas medidas de valor mínimo, máximo, médio, mediana e desvio padrão para então cada valor ser escalado individualmente num intervalo entre zero e um.

4.2.2 EEG

Para este sinal, a biblioteca *NeuroKit2* também foi utilizada, no entanto suas funcionalidades para EEG são mais limitadas quando comparado ao EDA. Duas colunas foram geradas a partir dos valores brutos coletados, a primeira é a dissimilaridade global, que mede a diferença topográfica do sinal entre dois instantes consecutivos e a outra com o campo de potência global, a qual consiste basicamente do desvio padrão ao longo dos canais para cada instante de tempo. Sendo assim, para os 20 canais e as duas colunas geradas, foram calculadas valores mínimo, máximo, médio, mediana e desvio padrão, além de aplicar a escala entre zero e um da mesma forma ao EDA (Brandão, 2022).

4.2.3 PPG

Por último, o processo realizado para este sinal foi mais extenso devido à qualidade da informação coletada. Algoritmos e métodos já estabelecidos, como o fornecido pela *NeuroKit2*, não foram capazes de tratar do processo como um todo, de modo que tanto etapas adicionais foram realizadas, quanto alterações pontuais nos algoritmos existentes foram usadas em conjunto. Embora este processo tenha tido como foco a tratar o sinal coletado pelo *Samsung Galaxy Watch4*, ele também foi aplicado ao *Empatica E4*, visto que as especificidades necessárias para o primeiro sinal não representam um prejuízo aos valores do segundo.

Inicialmente, antes de aplicar a transformação do sinal em janelas, foi realizada winsorização do sinal, removendo os 1% extremos, a fim de eliminar leituras atípicas. Em sequência, um filtro passa-faixa de resposta de impulso infinita *infinite impulse response* (IIR) do tipo *Butterworth* de segunda ordem com limite inferior de 0.1Hz e superior de 9Hz, através do pacote *SciPy*, em acordo ao estabelecido por Lapitan et al. (2024) para minimizar as distorções no formato da onda para este tipo de filtro. Por último, tendo em vista a amplitude grande entre máximo e mínimo, os valores foram escalados entre zero e um para continuar às próximas etapas.

Na sequência, a biblioteca *HeartPy*, foi utilizada para a geração de colunas com as características do sinal. No entanto, devido à qualidade ruim do sinal, o método completo de processamento de PPG fornecido pela biblioteca não foi capaz de consistentemente extrair as informações necessárias, sendo necessário substituir o método de cálculo de picos no sinal por uma solução própria. Ainda, apenas algumas funcionalidades de todo o processo fornecido pela *HeartPy* puderam ser utilizados devido ao tamanho limitado do conjunto de dados, o qual restringe o cálculo de algumas métricas relacionadas a respiração, por exemplo.

Sendo assim, utilizando o sinal previamente filtrado e escalado, passou pela função de busca de picos disponibilizada pelo *SciPy*, considerando a altura mínima de um pico sendo zero e a distância mínima entre os picos permitindo uma taxa de 100 batimentos por minuto (BPM) como o máximo de frequência cardíaca, por fim, caso a posição do primeiro pico esteja nos 150 milissegundos iniciais, o mesmo é descartado, pois o sinal pode ter iniciado no meio do batimento. Desta forma, enquanto a detecção de picos foi mais eficiente do que o atingido pelo método disponível *HeartPy*, nem sempre o pico identificado é o sistólico, por vezes sendo o diastólico. Esta inconsistência causa uma leve perda na precisão dos valores, o que ainda representa uma limitação deste método.

Tendo a lista de picos do sinal, quatro funções disponibilizadas pelo *HeartPy* são utilizadas em ordem, sendo elas *calc_rr* responsável por calcular o tempo entre picos do sinal fornecido, *check_peaks* onde valores de pico que desviam de 30% do valor médio do intervalo entre picos são descartados, *check_fd_measures* onde uma análise no domínio de frequência é realizada conforme elaborado por Shaffer e Ginsberg (2017) e por último *clean_rr_intervals* que detecta e remove anomalias nos intervalos entre picos atualizando a lista de intervalos entre picos, conforme método proposto por Piskorski e Guzik (2005). Estas são algumas das funções utilizadas no processamento completo realizado pela biblioteca e precisam ser realizadas nesta sequência, reutilizando uma estrutura de dados em comum. Como resultado, foram utilizadas as colunas com intervalos RR, diferença em intervalos RR, diferença quadrada de intervalos RR, intervalos RR corrigidos, frequência, PSD e batimentos removidos. Detalhes mais aprofundados a respeito do funcionamento destas funções estão disponíveis no trabalho de Gent et al. (2019b).

Por último, também foram gerados valores de BPM pela diferença de tempo entre picos e o IBI pelo tempo total entre os picos. Deste modo, portanto, formando a composição final da janela de PPG, estão o total de batimentos removidos, total de picos e mediadas de valor mínimo, máximo, média, mediana e desvio padrão para todas as outras colunas geradas através do *HeartPy*, IBI, BPM, e o valor do sinal após filtro.

4.3 Otimização de hiperparâmetros

Após todos os dados estarem processados e disponíveis para uso, foi feita a busca por hiperparâmetros com maior eficiência para cada um dos pares entre sinal e algoritmo de inteligência artificial. Não obstante, o processo de otimização para todos os casos faz o uso da biblioteca *Hyperopt*. Neste processo, primeiro define-se um espaço de busca de hiperparâmetros, esses valores são dependentes conforme o modelo de inteligência artificial escolhido. Em sequência, uma fração de 20% dos dados de treinamento foram utilizadas para treinar e avaliar o modelo com os hiperparâmetros escolhidos pela biblioteca. Este processo foi executado 100 vezes para os pares entre sinal e algoritmo tradicional e 40 vezes para cada par de sinal e algoritmo de aprendizado profundo, sendo o alto consumo de recursos e tempo fatores limitantes para mais execuções neste último caso. Finalmente, um histórico destas execuções são salvas e o conjunto que tenha resultado no menor valor de perda do modelo será utilizado para o seu treinamento na próxima etapa.

4.4 Aprendizado tradicional

Iniciando o treinamento dos modelos, o primeiro grupo de algoritmos, considerados como aprendizado de máquina tradicional, foram os responsáveis pelo uso dos dados após o pré-processamento detalhado na seção 4.2. Nesta etapa, três algoritmos foram utilizados, Pedregosa et al. (2011) forneceu SVM e KNN através dos algoritmos *SVC* e *KNeighborsClassifier* respectivamente, pela biblioteca *scikit-learn* enquanto Chen e Guestrin (2016) disponibilizou a biblioteca *XGBoost* um algoritmo de GBDT, referenciado neste projeto por XGB.

Para cada um dos três modelos foi aplicado o mesmo fluxo de treinamento. Inicialmente, os dados são carregados e o pré-processamento específico do sinal em questão é aplicado, assim como os melhores hiperparâmetros para o modelo que foram estabelecidos na etapa anterior. Na sequência o modelo é treinado no formato de LOSOCV, ou seja, são realizadas dez rodadas de treinamento com dados de dez voluntários, os quais forma usados exclusivamente para a etapa de treinamento de modelos. Em cada uma destas dez rodadas de treinamento, o modelo seria treinado em dados de nove pessoas e testado nos dados da décima, sendo um voluntário de testes distinto de cada vez, no entanto, para o sinal de EEG há uma inconsistência nos dados, em que para um dos voluntários há apenas medições de esforço cognitivo, sem o comparativo em estado de repouso, o que impossibilita um teste do modelo.

Neste cenário, a fim de obter conformidade entre o formato de treinamento entre todos os treinamentos de aprendizado tradicional, para cada uma das dez rodadas de treinamento, os modelos foram treinados em dados de 8 participantes, sendo outros dois voluntários utilizados para teste do modelo, com um destes voluntários sendo sempre distinto para cada cenário de teste e o outro escolhido aleatoriamente entre os nove que seriam utilizados para treinamento. Finalmente, ao final de cada rodada, o modelo treinado foi salvo para futuramente ser comparados na etapa de testes.

Tabela 4 – Resumo dos modelos tradicionais

| Modelo | Qtd. Hiperparâmetros | Característica | Referência |
|---------------|-----------------------------|-----------------------|-------------------------|
| SVM | 5 | Linear | Pedregosa et al. (2011) |
| KNN | 5 | Vizinhos | Pedregosa et al. (2011) |
| XGB | 14 | Conjunto | Chen e Guestrin (2016) |

Fonte: Elaborado pelo autor.

4.5 Aprendizado de ponta a ponta

Diferentemente do processo tradicional de aprendizado, neste modelo de ponta a ponta, os algoritmos utilizados foram de aprendizado profundo e o único pré-processamento aplicado foi a separação dos dados em janelas de 30 segundos, com os valores originais lidos pelos sensores. Esta metodologia tem como intuito eliminar a necessidade de conhecimento específico sobre processamento de sinais e características específicas de cada tipo de indicador psicofisiológico, de modo a simplificar a execução do processo de aprendizado de máquina. Deste modo, o próprio modelo de aprendizado seria responsável por lidar com a limpeza de artefatos e reconhecer informações relevantes dos valores.

Em contrapartida, o custo computacional para realizar o treinamento é muito maior, sendo necessário um sistema de computação em nuvem, equipado com uma placa gráfica NVIDIA L4 com 21 GB de memória, resultando em um custo total de R\$ 583.44 ao longo de dois meses, considerando testes de implementação e treinamento final dos modelos.

Ademais, a estrutura dos dados para treinamento foi dividida em canais, janelas e valor, ou seja, para cada diferente canal de dados coletados pelo sensor, uma entrada diferente foi criada para o modelo, com todos estes valores sendo concatenados antes da camada final. Entre os sensores utilizados, apenas o EEG possuía mais que um canal, portanto este foi o único caso em que os modelos trabalharam com múltiplas entradas.

Para esta etapa, um total de quatro arquiteturas foram utilizadas, sendo todas elas voltadas para a classificação de dados temporais e também já utilizadas em trabalhos relacionados de detecção de esforço cognitivo. O primeiro modelo utilizado, proposto por Fawaz et al. (2019) é o Time-CNN, referido apenas por CNN no resto do trabalho, o qual é composto por duas camadas de convolução seguidas de uma camada de *pooling* local médio cada, sendo a quantidade de filtros múltiplos de seis e 12, respectivamente, com tamanho do filtro sendo um múltiplo de sete, enquanto para as camadas de *pooling*, ambas contam com tamanho três. O resultado é então combinado em uma camada densa, com função de ativação sigmoide.

O próximo modelo, também proposto por Fawaz et al. (2019) é uma rede FCN, que mantém constante o tamanho da série temporal devido ao fato da ausência de camadas de *pooling* locais. Neste modelo, faz-se presente três conjuntos de convolução seguida por normalização em lote e uma função de ativação ReLU, com a quantidade e tamanho dos filtros por conjunto,

respectivamente, sendo múltiplos de 128 e 8, 256 e 5 e por último 128 e 3. Finalmente, é aplicada uma camada de *Global Average Pooling* (GAP), para toda a série temporal, antes de passar para a camada densa com ativação sigmoide.

O último modelo utilizado proposto por Fawaz et al. (2019) foi o ResNet, foi originalmente composta por três blocos, mas esta quantidade foi tratada como hiperparâmetro neste trabalho, e fornecendo um atalho entre dois destes componentes consecutivos, que permite ligar a entrada e saída de um bloco, de modo a evitar o problema de dissipação de gradiente, o qual ocorre ao atualizar um peso pela derivada parcial do erro do modelo para o específico peso, podendo parar o treino do modelo quando muito profundo. Neste modelo, cada um dos blocos consecutivos são compostos por três camadas de convolução, normalização em lote e função de ativação ReLU, sendo a quantidade de filtros fixa em 64 e tamanhos múltiplos de oito, cinco e três, respectivamente, enquanto o atalho é composto por uma convolução com mesma quantidade de filtros mas com tamanho um, seguida de uma normalização em lote, sendo, finalmente, o resultado de um bloco a soma das três camadas de convolução ao resultado do atalho. O último bloco possui a mesma estrutura, mas a quantidade de filtros também passa a ser um hiperparâmetro. Finalmente, uma camada GAP é aplicada logo antes da camada densa com ativação sigmoide.

Finalmente, Kanjo, Younis e Ang (2019) descreve um modelo de CNN-LSTM que inspirou a rede referida apenas por LSTM neste trabalho. Algumas alterações foram feitas em comparação ao modelo original devido a restrições de poder computacional ao treinar no sinal de EEG, levando a redução da quantidade de filtros mas aumento do seu tamanho. O diferencial deste modelo de algoritmo é a introdução de blocos que controlam quais informações prévias devem ser efetivamente esquecidas e quais informações atuais devem ser lembradas, o que permite relacionar causas e efeitos nos dados iniciais de entrada, mesmo que distantes um do outro. Para a arquitetura usada, duas camadas de convoluções tendo quantidade e tamanho de filtros múltiplos de quatro e oito, respectivamente, foram seguidas, cada, de uma camada de *pooling* máximo, com tamanho 2. Na sequencia, diferentemente dos demais modelos, cada canal individual é concatenado, antes de passar por uma camada densa e então pelo bloco de LSTM que possui como hiperparâmetro a quantidade de células de memória a ser usada, para finalmente passar por uma camada densa com função de ativação sigmoide.

Após ter os dados e hiperparâmetros carregados, o treinamento ocorreu no modo LO-SOCV, também em dez rodadas de treinamento e os mesmos voluntários. No entanto, diferentemente do processo tradicional, em cada rodada os dados foram divididos em oito participantes para treino, um para validação e um para testes, com o voluntário de testes sendo único para cada uma das rodadas. Para todos os modelos, um limite de 300 épocas foi estipulado, podendo ser interrompido de maneira prévia caso não houvesse uma melhora significativa na performance do modelo por 30 épocas consecutivas. Finalmente, no final de cada rodada, apenas os pesos do melhor modelo foi salvo, para comparação posterior, durante a etapa de testes.

Tabela 5 – Resumo dos modelos

| Modelo | Camadas | Normalização | Pooling | Final | Referência |
|---------------|----------------|---------------------|----------------|--------------|----------------------------|
| CNN | 3 | N/A | Médio | Conv1D | Fawaz et al. (2019) |
| FCN | 5 | Lotes | N/A | GAP | Fawaz et al. (2019) |
| RESNET | 11 \pm 4 | Lotes | N/A | GAP | Fawaz et al. (2019) |
| LSTM | 5 | N/A | Max | Conv1D | Kanjo, Younis e Ang (2019) |

Fonte: Elaborado pelo autor.

4.6 Testes de modelos

Após todos os 28 tipos distintos de modelos terem sido treinados, cada um dos 10 melhores resultados salvos durante o processo de LOSOCV foi carregado para testar sua eficiência em detectar esforço cognitivo para dados novos, que foram reservados exclusivamente para testes, mas que passaram pelo mesmo fluxo de pré-processamento dos dados de treinamento. Este conjunto foi composto por três voluntários, com cada um sendo avaliado individualmente pelos modelos, com o objetivo de observar a capacidade de generalização para indivíduos distintos, os quais podem possuir padrões diferentes de indicadores psicofisiológicos quando em estado de esforço cognitivo. Após todos os testes, dentre os 10 melhores modelos para cada arquitetura, foi filtrado apenas o que apresentou melhor acurácia, para então realizar a comparação entre diferentes tipos de modelos.

4.7 Discussão

Neste trabalho, dois paradigmas distintos foram abordados para detecção de esforço cognitivo a partir de dados psicofisiológicos. Primeiramente um processo com engenharia de características, que envolveu processamento de sinais distintos, removendo ruídos e interferências nos valores coletados e derivação de valores indicativos de um estado físico do corpo humano produziu valores de entrada para três algoritmos tradicionais de aprendizado de máquina. Ainda, quatro modelos de aprendizado profundo também foram utilizados para identificar esforço cognitivo, mas a partir das informações originais, sem intervenção para limpeza de dados ou produção de características. A estrutura de classes desenvolvidas é exibida na Figura 8, com componentes relacionados sendo agrupados por cores.

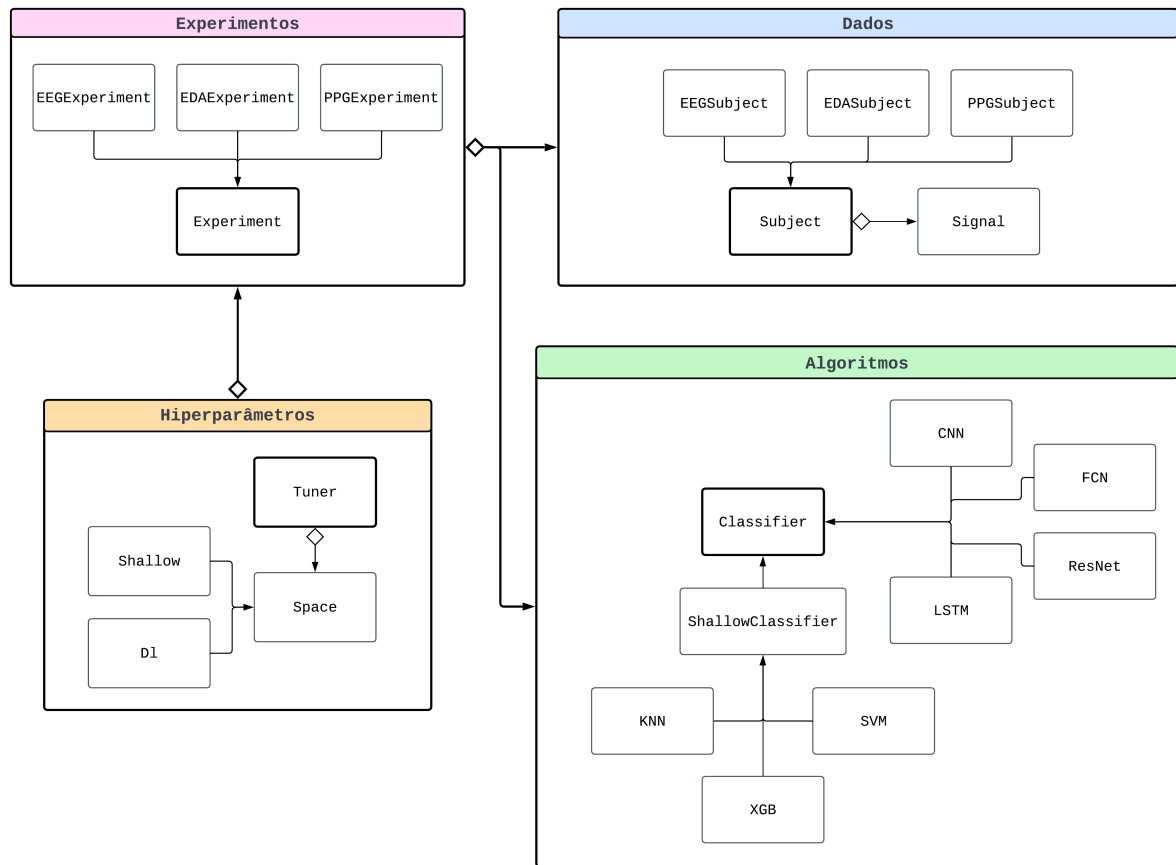
Todos os modelos foram treinados a partir dos dados de 10 voluntários e avaliados nos dados de outros três, com métricas para os melhores modelos agrupados por sensor disponíveis na Tabela 6. As métricas exibidas são uma média da performance de cada modelo entre todos os participantes de teste.

Pode-se observar que, em média, os modelos tradicionais tiveram melhor desempenho na classificação correta de valores, independentemente de qual era o estado cognitivo, enquanto os modelos de aprendizado profundo apresentaram um viés a favor de esforço cognitivo, por isso

Tabela 6 – Resumo dos resultados

| | Modelo | Acurácia | Precisão | Sensibilidade | F1 | AUC |
|---------------------|---------------|-----------------|-----------------|----------------------|--------------|--------------|
| Samsung PPG | CNN | 0.619 | 0.626 | 0.929 | 0.748 | 0.534 |
| | FCN | 0.682 | 0.671 | 0.966 | 0.789 | 0.608 |
| | LSTM | 0.608 | 0.608 | 1.0 | 0.756 | 0.5 |
| | RESNET | 0.714 | 0.686 | 1.0 | 0.812 | 0.639 |
| | KNN | 0.695 | 0.718 | 0.808 | 0.756 | 0.666 |
| | SVM | 0.653 | 0.643 | 0.95 | 0.767 | 0.579 |
| | XGB | 0.73 | 0.754 | 0.844 | 0.791 | 0.698 |
| Muse EEG | CNN | 0.595 | 0.589 | 0.667 | 0.609 | 0.603 |
| | FCN | 0.667 | 0.658 | 1.0 | 0.768 | 0.681 |
| | LSTM | 0.412 | 0.629 | 0.309 | 0.308 | 0.464 |
| | RESNET | 0.533 | 0.53 | 0.979 | 0.682 | 0.515 |
| | KNN | 0.602 | 0.601 | 0.754 | 0.662 | 0.603 |
| | SVM | 0.524 | 0.524 | 1.0 | 0.684 | 0.5 |
| | XGB | 0.571 | 0.558 | 0.922 | 0.688 | 0.56 |
| Empatica PPG | CNN | 0.55 | 0.64 | 0.6 | 0.618 | 0.536 |
| | FCN | 0.696 | 0.705 | 0.912 | 0.792 | 0.637 |
| | LSTM | 0.548 | 0.597 | 0.808 | 0.686 | 0.473 |
| | RESNET | 0.642 | 0.7 | 0.826 | 0.746 | 0.594 |
| | KNN | 0.636 | 0.686 | 0.746 | 0.714 | 0.602 |
| | SVM | 0.709 | 0.74 | 0.842 | 0.78 | 0.661 |
| | XGB | 0.614 | 0.614 | 1.0 | 0.761 | 0.5 |
| Empatica EDA | CNN | 0.602 | 0.609 | 0.983 | 0.752 | 0.492 |
| | FCN | 0.613 | 0.613 | 1.0 | 0.76 | 0.5 |
| | LSTM | 0.537 | 0.575 | 0.842 | 0.677 | 0.449 |
| | RESNET | 0.613 | 0.613 | 1.0 | 0.76 | 0.5 |
| | KNN | 0.688 | 0.733 | 0.799 | 0.758 | 0.658 |
| | SVM | 0.711 | 0.711 | 0.917 | 0.8 | 0.644 |
| | XGB | 0.743 | 0.75 | 0.885 | 0.812 | 0.696 |

Fonte: Elaborado pelo autor.

Figura 8 – Organização de classes desenvolvidas

Fonte: Elaborado pelo autor.

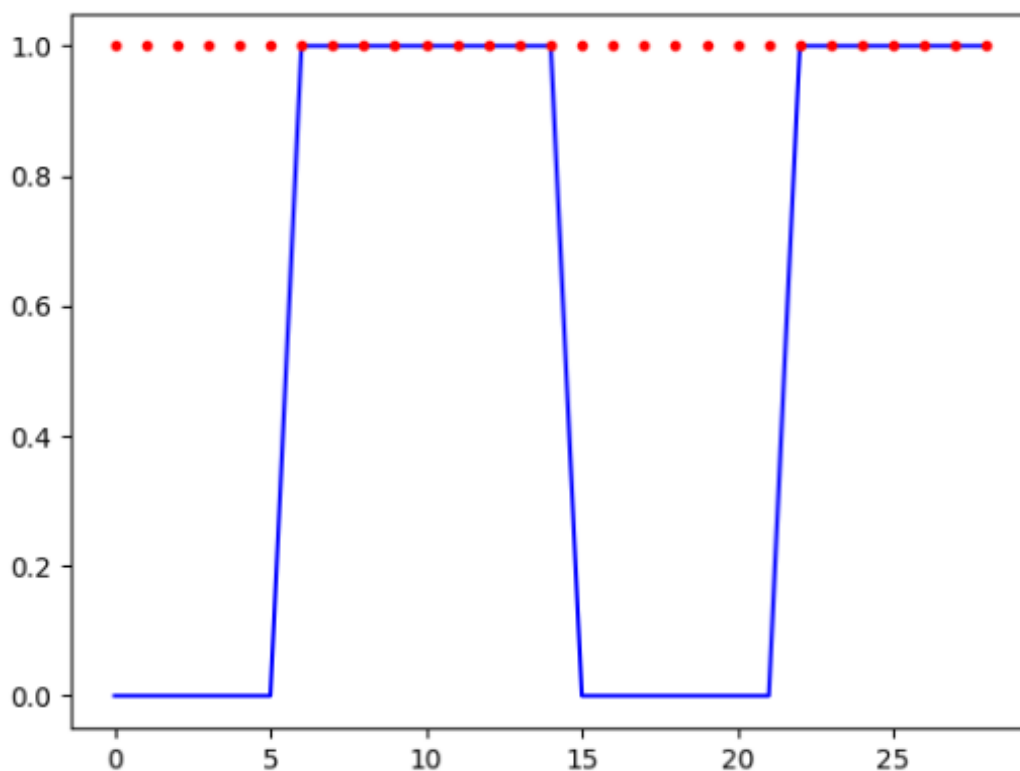
tendo, em média, um valor de F1 mais elevado. Isto ocorre pelo fato de existir mais informação disponível para esforço cognitivo, de modo que um modelo o qual sempre resulte em uma classificação positiva para uma janela, tenha um valor de precisão mediano e sensibilidade igual a um, resultando em uma boa métrica para F1.

Ademais, é importante ressaltar que os modelos tiveram desempenho muito diferente entre os participantes do cenário de testes. Este fator sugere que devem haver diferenças pessoais significativas para os indicadores psicofisiológicos. Mejia-Mejia et al. (2022) demonstra diferentes formatos de onda PPG quando coletado em diferentes idades, estas variações podem causar uma diferença significativa nos métodos para pré-processamento e a própria detecção por algum modelo de inteligência artificial, seja tradicional ou de aprendizado profundo. No entanto, o conjunto de dados selecionado não possui informações mais detalhadas sobre os participantes, como sua idade ou gênero, fator impeditivo para relacionar alguma diferença de performance com a característica de um participante.

Para exemplificar a magnitude de variações entre voluntários, na Figura 9 é exibido a classificação para o modelo FCN de melhor performance para o sinal EEG ao testar o primeiro

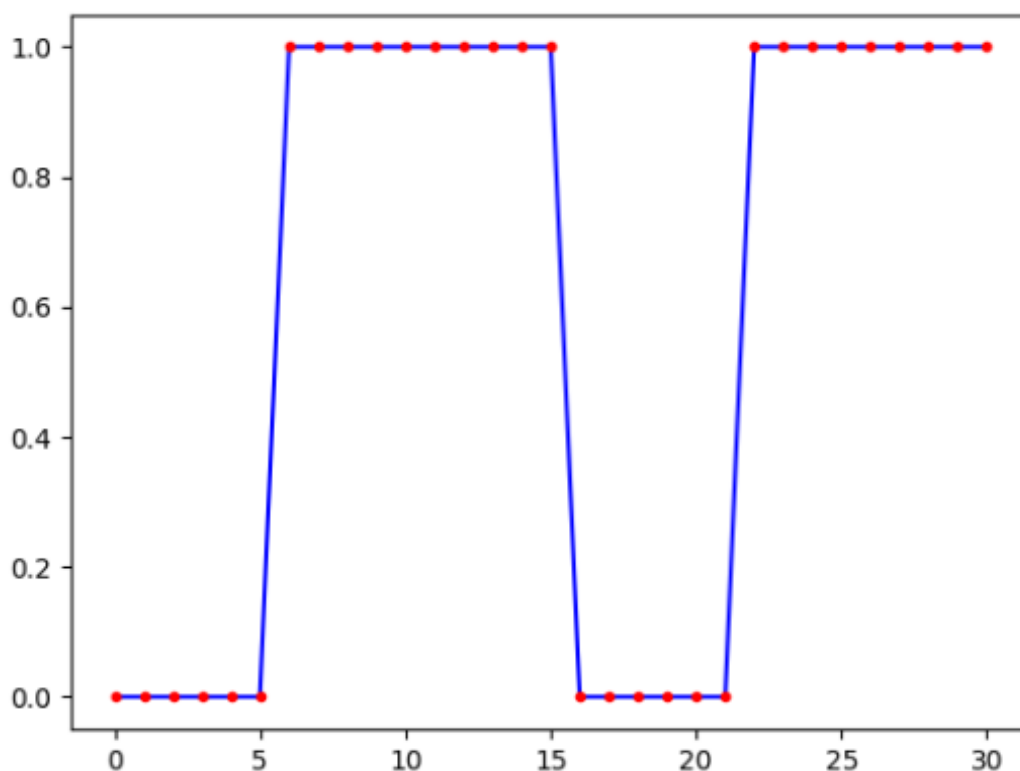
voluntário do conjunto de testes. Nesta imagem, a linha azul representa os valores reais do conjunto de dados, onde zero equivale ao estado normal e um a esforço cognitivo, enquanto os pontos vermelhos representam o estado inferido pelo modelo.

Figura 9 – Resultado FCN EEG para primeiro participante de teste



Fonte: Elaborado pelo autor.

Enquanto que na Figura 10, é exibido o teste do mesmo modelo mas para o segundo participante do grupo de testes. Embora que na primeira imagem o modelo tenha classificado todas as janelas como esforço cognitivo, para o segundo participante a classificação foi perfeita. Este exemplo repete-se para outros modelos e para outros sinais.

Figura 10 – Resultado FCN EEG para segundo participante de teste

Fonte: Elaborado pelo autor.

Outro ponto importante é a comparação da performance dos sinais coletados por dispositivos comerciais em relação ao dispositivo clínico. O melhor valor absoluto observado foi obtido pelo sinal EDA coletado pelo *Empatica E4*, enquanto que o valor de PPG coletado pelo mesmo dispositivo apresentou performance inferior aos dados vindos do *Samsung Galaxy Watch4*. Contudo, a seleção de características para PPG foi limitada e baseada pelos dados vindos do dispositivo da Samsung, devido a baixa qualidade do sinal, o que pode sugerir um viés nos dados obtidos no processo de engenharia de características. Mais um fator para corroborar esta tese do viés parte do fato de que a performance dos modelos em aprendizado de ponta a ponta, onde não houve intervenção nos dados coletados, foi melhor para o dispositivo de qualidade clínica, da Empatica.

4.8 Contribuições

Diversos modelos para detecção de esforço cognitivo foram desenvolvidos. Estes modelos podem auxiliar no monitoramento contínuo deste estado tanto para pessoas interessadas na métrica, quanto para pacientes que possam estar realizando algum tratamento para auxílio na atenção ou até utilizado em sistemas de notificação, podendo servir como mecanismo para identificar a necessidade de gerar um alerta para caso de ausência ou presença de esforço cognitivo.

Ademais, de acordo com os resultados obtidos, pode-se concluir que o uso de dados coletados por dispositivos vestíveis comerciais pode ser usado para esta tarefa, sem necessariamente depender de dispositivos projetados para estudos clínicos. Este fator acaba por democratizar o acesso a este mecanismo, tendo em vista que tanto estes dispositivos já estão altamente disponíveis no mercado, quanto pelo baixo custo do aparelho não representar uma barreira grande ao acesso para novos indivíduos. A fim de comparação, o preço atual dos três dispositivos utilizados neste trabalho são 199,99² USD para *Samsung Galaxy Watch4*, 399,99³ USD para *Muse S headband* e 1.690,00⁴ USD para *Empatica E4*.

Não obstante ao foco de classificar estado psicofisiológico de esforço cognitivo, as etapas realizadas neste processo podem servir de apoio para outros projeto que lidem com processamento de sinais psicofisiológicos, uso de sensores em dispositivos de entrada, uso de características para classificação em tempo real de estados emocionais e uso de modelos de aprendizado de ponta a ponta em séries temporais.

² Disponível em: <<https://www.samsung.com/us/watches/galaxy-watch4/buy/>>. Acesso em: 16 out. 2024

³ Disponível em: <<https://choosemuse.com/products/muse-s-gen-2>>. Acesso em: 16 out. 2024

⁴ Disponível em: <<https://store.empatica.com/products/e4-wristbands?variant=17719039950953>>. Acesso em: 16 out. 2024

5 CONSIDERAÇÕES FINAIS

Monitoramento de esforço cognitivo pode ser um aliado na prevenção ou detecção precoce de cenários de estresse crônico e depressão. No entanto, o processo de identificação de esforço cognitivo, até então depende de métodos subjetivos e com reprodução inviável, o que impossibilita o seu monitoramento em escala, de forma acessível e não intrusiva.

Desta forma, o desenvolvimento de um modelo de inteligência artificial, treinado com dados de dispositivos vestíveis populares, capaz de identificar janelas de tempo com esforço cognitivo contribui diretamente para o estabelecimento de uma forma de monitoramento contínuo e independente, até então, inexistente. Neste cenário, a capacidade de dispensar o uso de dispositivos clínicos, usados em estudos de contexto médico contribui para a democratização deste processo, tendo em vista que há uma ordem de magnitude na diferença de preço entre as duas classes de equipamentos.

Esta possibilidade de monitoramento de um estado psicológico pode contribuir para pesquisas a respeito de efetividade de medicamentos voltados a transtornos de atenção, acompanhamento de performance num contexto de trabalho moderno, como *knowledge workers*, e sua relação com transtornos mentais como depressão, ansiedade e síndrome de *burnout*.

Ademais, os métodos desenvolvidos no trabalho também beneficiam-se de avanços tecnológicos no setor de dispositivos vestíveis, uma vez que estes também passam por processos de melhoria de manufatura, qualidade de sensores utilizados e tipos distintos de sensores utilizados. Neste trabalho, os modelos de inteligência artificial limitaram-se ao uso de apenas um sensor, no entanto é cada vez mais comum que os aparelhos modernos estejam equipados tanto de um sensor PPG quanto EDA, por exemplo.

Não obstante, este trabalho possui limitações. Primeiramente, o conjunto de dados utilizados possui um tamanho bastante pequeno, com poucos participantes e um período curto de tempo coletado por participante, o qual chega a ser uma limitação para extração de características para alguns sinais psicofisiológicos. Além da falta de informações adicionais sobre os voluntários que compõem o conjunto, como faixa etária, que podem impactar no comportamento dos indicadores psicofisiológicos coletados. Outra limitação, além do uso de um único sensor já citada, deve-se ao fato de que dispositivos de qualidade comercial apresentam uma pior relação de sinal-ruído, o que exige métodos mais robustos de pré-processamento. Por último, o treinamento realizado foi com base em um cenário controlado, sem dados coletados durante o decorrer normal de um dia, o que pode trazer certa fragilidade aos modelos quando confrontados com dados obtidos durante a realização de atividades distintas. Todos estes pontos podem ser aprofundados em trabalhos futuros e incorporados como melhorias.

REFERÊNCIAS

- AHMAD, M. I.; KELLER, I.; ROBB, D. A.; LOHAN, K. S. A framework to estimate cognitive load using physiological data. **Personal and Ubiquitous Computing**, Springer, p. 1–15, 2023.
- ALLEN, J. **Photoplethysmography and its application in clinical physiological measurement**. 2007.
- AYRES, P.; LEE, J. Y.; PAAS, F.; MERRIËNBOER, J. J. van. **The Validity of Physiological Measures to Identify Differences in Intrinsic Cognitive Load**. 2021.
- AYRES, P.; PAAS, F. Cognitive load theory: New directions and challenges. **Applied Cognitive Psychology**, v. 26, 2012. ISSN 08884080.
- BERGSTRA, J.; YAMINS, D.; COX, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: PMLR. **International conference on machine learning**. [S.l.], 2013. p. 115–123.
- BERRAR, D. Cross-validation. In: RANGANATHAN, S.; GRIBSKOV, M.; NAKAI, K.; SCHÖNBACH, C.; CANNATARO, M. (Ed.). **Encyclopedia of Bioinformatics and Computational Biology**. Elsevier, 2019, (Reference Module in Life Sciences, v. 1). p. 542–545. ISBN 978-0-12-811432-2. Disponível em: <<https://oro.open.ac.uk/96776/>>.
- BORISOV, V.; KASNECI, E.; KASNECI, G. Robust cognitive load detection from wrist-band sensors. **Computers in Human Behavior Reports**, v. 4, 2021. ISSN 24519588.
- BOUCSEIN, W.; FOWLES, D. C.; GRIMNES, S.; BEN-SHAKHAR, G.; ROTH, W. T.; DAWSON, M. E.; FILION, D. L. Publication recommendations for electrodermal measurements. **Psychophysiology**, v. 49, 2012. ISSN 14698986.
- BRANDÃO, D. S. Investigação das funções cognitivas do sono e dos sonhos através de eletroencefalografia, relatos verbais e jogos eletrônicos. Universidade Federal do Rio Grande do Norte, 2022.
- BROUWER, A. M.; ZANDER, T. O.; ERP, J. B. van; KORTELING, J. E.; BRONKHORST, A. W. Using neurophysiological signals that reflect cognitive or affective state: Six recommendations to avoid common pitfalls. **Frontiers in Neuroscience**, v. 9, 2015. ISSN 1662453X.
- BUJA, A.; COOK, D.; HOFMANN, H.; LAWRENCE, M.; LEE, E.-K.; SWAYNE, D. F.; WICKHAM, H. Statistical inference for exploratory data analysis and model diagnostics. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, The Royal Society Publishing, v. 367, n. 1906, p. 4361–4383, 2009.
- CAI, M.; EPP, C. D. Exploring the optimal time window for predicting cognitive load using physiological sensor data. 6 2024. Disponível em: <<https://arxiv.org/abs/2406.13793v1>>.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.
- CHOLLET, F. et al. **Keras**. 2015. <<https://keras.io>>.

CINAZ, B. **Monitoring of cognitive load and cognitive performance using wearable sensing**. [S.l.: s.n.], 2013. ISBN 9781304292490.

COHEN, M. X. **Where Does EEG Come From and What Does It Mean?** 2017.

CONWAY, D.; DICK, I.; LI, Z.; WANG, Y.; CHEN, F. The effect of stress on cognitive load measurement. In: . [S.l.: s.n.], 2013. v. 8120 LNCS. ISSN 03029743.

DING, Y.; CAO, Y.; DUFFY, V. G.; WANG, Y.; ZHANG, X. Measurement and identification of mental workload during simulated computer tasks with multimodal methods and machine learning. **Ergonomics**, v. 63, 2020. ISSN 13665847.

DRUCKER, P. F. Knowledge-worker productivity the biggest challenge. **IEEE Engineering Management Review**, v. 34, 2006. ISSN 03608581.

DZIEŻYC, M.; GJORESKI, M.; KAZIENKO, P.; SAGANOWSKI, S.; GAMS, M. Can we ditch feature engineering? end-to-end deep learning for affect recognition from physiological sensor data. **Sensors (Switzerland)**, v. 20, 2020. ISSN 14248220.

FAWAZ, H. I.; FORESTIER, G.; WEBER, J.; IDOUMGHAR, L.; MULLER, P.-A. Deep learning for time series classification: a review. **Data Mining and Knowledge Discovery**, v. 33, n. 4, p. 917–963, 2019.

FLEMING, H.; ROBINSON, O. J.; ROISER, J. P. Measuring cognitive effort without difficulty. **Cognitive, Affective, & Behavioral Neuroscience**, Springer, v. 23, n. 2, p. 290–305, 2023.

GENT, P. van; FARAH, H.; NES, N. van; AREM, B. van. Analysing noisy driver physiology real-time using off-the-shelf sensors: Heart rate analysis software from the taking the fast lane project. **Journal of Open Research Software**, Ubiquity Press, v. 7, 2019. ISSN 20499647.

_____. Heartpy: A novel heart rate algorithm for the analysis of noisy signals. **Transportation Research Part F: Traffic Psychology and Behaviour**, Pergamon, v. 66, p. 368–378, 10 2019. ISSN 1369-8478.

GJORESKI, M.; MAHESH, B.; KOLENIK, T.; UWE-GARBAS, J.; GJORESKI, H.; PEJOVIC, V.; LUSTREK, M.; GAMS, M. Cognitive load monitoring with wearables-lessons learned from a machine learning challenge. **IEEE Access**, v. 9, 2021. ISSN 21693536.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

GRZESZCZYK, M. K.; BLANCO, R.; ADAMCZYK, P.; KUS, M.; MAREK, S.; PRĘCIKOWSKI, R.; LISOWSKA, A. Cogwear: Can we detect cognitive effort with consumer-grade wearables? 2023.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COUNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>.

HEARST, M. A.; DUMAIS, S. T.; OSUNA, E.; PLATT, J.; SCHOLKOPF, B. Support vector machines. **IEEE Intelligent Systems and their applications**, IEEE, v. 13, n. 4, p. 18–28, 1998.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

ISLAM, M. R.; MONI, M. A.; ISLAM, M. M.; RASHED-AL-MAHFUZ, M.; ISLAM, M. S.; HASAN, M. K.; HOSSAIN, M. S.; AHMAD, M.; UDDIN, S.; AZAD, A.; ALYAMI, S. A.; AHAD, M. A. R.; LIO, P. Emotion recognition from eeg signal focusing on deep learning and shallow learning techniques. **IEEE Access**, v. 9, 2021. ISSN 21693536.

JOSEPH, G.; JOSEPH, A.; TITUS, G.; THOMAS, R. M.; JOSE, D. Photoplethysmogram (ppg) signal analysis and wavelet de-noising. In: **2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD)**. [S.l.: s.n.], 2014. p. 1–5.

KANJO, E.; YOUNIS, E. M.; ANG, C. S. Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. **Information Fusion**, v. 49, 2019. ISSN 15662535.

KRAMER, O.; KRAMER, O. K-nearest neighbors. **Dimensionality reduction with unsupervised nearest neighbors**, Springer, p. 13–23, 2013.

LANGLEY, P.; SIMON, H. A. Applications of machine learning and rule induction. **Communications of the ACM**, ACM-PUB27 New York, NY, USA, v. 38, p. 54–64, 11 1995. ISSN 15577317. Disponível em: <<https://dl.acm.org/doi/10.1145/219717.219768>>.

LAPITAN, D. G.; ROGATKIN, D. A.; MOLCHANOVA, E. A.; TARASOV, A. P. Estimation of phase distortions of the photoplethysmographic signal in digital iir filtering. **Scientific Reports**, Nature Publishing Group UK London, v. 14, n. 1, p. 6546, 2024.

LEE, J.; MATSUMURA, K.; YAMAKOSHI, K. I.; ROLFE, P.; TANAKA, S.; YAMAKOSHI, T. Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion. In: . [S.l.: s.n.], 2013. ISSN 1557170X.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015.

MAKOWSKI, D.; PHAM, T.; LAU, Z. J.; BRAMMER, J. C.; LESPINASSE, F.; PHAM, H.; SCHÖLZEL, C.; CHEN, S. H. A. Neurokit2: A python toolbox for neurophysiological signal processing. **Behavior Research Methods**, Springer, v. 53, p. 1689–1696, 8 2021. ISSN 15543528. Disponível em: <<https://link.springer.com/article/10.3758/s13428-020-01516-y>>.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfan van der; MILLMAN Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 56 – 61.

MEJIA-MEJIA, E.; ALLEN, J.; BUDIDHA, K.; EL-HAJJ, C.; KYRIACOU, P. A.; CHARLTON, P. H. Photoplethysmography signal processing and synthesis. In: **Photoplethysmography**. [S.l.: Elsevier, 2022. p. 69–146.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. **Frontiers in neurorobotics**, Frontiers Media SA, v. 7, p. 21, 2013.

PAAS, F.; TUOVINEN, J. E.; TABBERS, H.; GERVEN, P. W. V. Cognitive load measurement as a means to advance cognitive load theory. **Educational Psychologist**, v. 38, 2003. ISSN 00461520.

PAAS, F. G.; MERRIËNBOER, J. J. V. Instructional control of cognitive load in the training of complex cognitive tasks. **Educational Psychology Review**, v. 6, 1994. ISSN 1040726X.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PISKORSKI, J.; GUZIK, P. Filtering poincare plots. **Computational methods in science and technology**, ICHB PAS Poznan Supercomputing and Networking Center, v. 11, n. 1, p. 39–48, 2005.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Pearson, 2016.

SAGANOWSKI, S. **Bringing Emotion Recognition out of the Lab into Real Life: Recent Advances in Sensors and Machine Learning**. 2022.

SAGANOWSKI, S.; PERZ, B.; POLAK, A. G.; KAZIENKO, P. **Emotion Recognition for Everyday Life Using Physiological Signals From Wearables: A Systematic Literature Review**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2023. 1876-1897 p.

SCHEFFEL, C.; GÄRTNER, A. **The role of cognitive effort in emotion regulation**. [S.l.]: OSF, 2024.

SCHMIDT, P.; DÜRICHEN, R.; REISS, A.; LAERHOVEN, K. V.; PLÖTZ, T. Multi-target affect detection in the wild: An exploratory study. In: . [S.l.: s.n.], 2019. ISSN 15504816.

SHAFFER, F.; GINSBERG, J. P. An overview of heart rate variability metrics and norms. **Frontiers in public health**, Frontiers Media SA, v. 5, p. 258, 2017.

SHANNON, C. E. Communication in the presence of noise. **Proceedings of the IRE**, v. 37, 1949. ISSN 00968390.

SHU, L.; YU, Y.; CHEN, W.; HUA, H.; LI, Q.; JIN, J.; XU, X. Wearable emotion recognition using heart rate data from a smart bracelet. **Sensors (Switzerland)**, v. 20, 2020. ISSN 14248220.

SIEGEL, M.; DONNER, T. H.; ENGEL, A. K. **Spectral fingerprints of large-scale neuronal interactions**. 2012.

SWELLER, J.; AYRES, P.; KALYUGA, S. **Cognitive Load Theory**. [S.l.]: Springer New York, 2011. v. 55. Academic Press. ISBN 978-1-4419-8125-7.

TERVONEN, J.; PETTERSSON, K.; MÄNTYJÄRVI, J. Ultra-short window length and feature importance analysis for cognitive load detection from wearable sensors. **Electronics** **2021**, Vol. 10, Page 613, Multidisciplinary Digital Publishing Institute, v. 10, p. 613, 3 2021. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/10/5/613/htmhttps://www.mdpi.com/2079-9292/10/5/613>>.

The pandas development team. **pandas-dev/pandas: Pandas**. Zenodo, 2020. Disponível em: <<https://doi.org/10.5281/zenodo.3509134>>.

TUKEY, J. W. et al. **Exploratory data analysis**. [S.l.]: Springer, 1977. v. 2.

VILLE, B. D. Decision trees. **Wiley Interdisciplinary Reviews: Computational Statistics**, Wiley Online Library, v. 5, n. 6, p. 448–455, 2013.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J.; van der Walt, S. J.; BRETT, M.; WILSON, J.; MILLMAN, K. J.; MAYOROV, N.; NELSON, A. R. J.; JONES, E.; KERN, R.; LARSON, E.; CAREY, C. J.; POLAT, İ.; FENG, Y.; MOORE, E. W.; VanderPlas, J.; LAXALDE, D.; PERKTOLD, J.; CIMRMAN, R.; HENRIKSEN, I.; QUINTERO, E. A.; HARRIS, C. R.; ARCHIBALD, A. M.; RIBEIRO, A. H.; PEDREGOSA, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. **Nature Methods**, v. 17, p. 261–272, 2020.

WESTBROOK, A.; BOSCH, R. van den; MäätTä, J. I.; HOFMANS, L.; PAPADOPETRAKI, D.; COOLS, R.; FRANK, M. J. Dopamine promotes cognitive effort by biasing the benefits versus costs of cognitive work. **Science**, v. 367, 2020. ISSN 10959203.

WU, J.; CHEN, X. Y.; ZHANG, H.; XIONG, L. D.; LEI, H.; DENG, S. H. Hyperparameter optimization for machine learning models based on bayesian optimization. **Journal of Electronic Science and Technology**, v. 17, 2019. ISSN 2666223X.

IDENTIFICAÇÃO DE ESFORÇO COGNITIVO COM AUXÍLIO DE DISPOSITIVOS VESTÍVEIS DE QUALIDADE COMERCIAL E INTELIGÊNCIA ARTIFICIAL

Mateus Antonio Franceschina

mateus.a@unochapeco.edu.br

Universidade Comunitária da Região de Chapecó (Unochapecó)

Este trabalho dedica-se para a elaboração de um sistema de detecção de esforço cognitivo através de arquiteturas de inteligência artificial e dados psicofisiológicos coletados por dispositivos vestíveis comuns, amplamente disponíveis no mercado. Inicialmente, o conceito de esforço cognitivo refere-se a aplicação mental de algum indivíduo na realização de determinada tarefa, a qual exige certo grau de conhecimento específico aplicado. Dentre as diversas formas em que se pode identificar esforço cognitivo, o uso de indicadores psicofisiológicos destaca-se por tornar este processo independente da participação direta do sujeito avaliado ou de algum observador avaliando o desempenho na execução da tarefa em questão. Estes indicadores carregam informações sobre os sistemas internos do corpo humano, como pelo volume de sangue em determinada região em um sensor PPG, potencial de condutibilidade elétrica na pele por EDA ou a atividade de células no cérebro através de EEG. A partir destes sensores, uma série de processos foram aplicados para limpar os dados de possíveis interferências e artefatos para então extrair características descritivas em determinado período de tempo, com o resultado deste processo sendo direcionado para os modelos tradicionais de aprendizado de máquina SVM, KNN e GBDT. Ademais, outros modelos de aprendizado profundo, sendo eles CNN, FCN, LSTM e ResNet, foram utilizados em um fluxo sem intervenção humana nos sinais coletados, denominado aprendizado de ponta

a ponta, no qual os modelos foram responsáveis por, internamente, remover ruídos dos sinais e extrair características relevantes de cada sinal, reduzindo, portanto, o viés de características geradas manualmente além da necessidade de conhecimento específico de processamento de sinais e cada indicador psicofisiológico em particular. Finalmente, todos os modelos foram

testados para detecção de esforço cognitivo em dados de três voluntários a fim de comparar a capacidade de generalização para cada modelo. Concluindo, destaca-se a capacidade de modelos treinados em dados de dispositivos comerciais em identificar esforço cognitivo, especialmente através de engenharia de características, mas também através do aprendizado de ponta a ponta, o que contribui para o tanto para o monitoramento ativo deste estado mental, quanto para outras pesquisas de detecção de emoções através de sinais semelhantes.